2\textsuperscript{nd} International EuroConference on
Computer and IT Applications in the Maritime Industries

COMPIT’03

Hamburg, 14-17 May 2003

Volker Bertram (Ed.)

An event supported by the European Commission,
Human Potential Programme, Contract HPCF-CT-2002-00291

We are grateful for the support of Blohm\&Voss GmbH, DFG and TUHH Technologie GmbH

This work relates to Department of the Navy Grant N00014-03-1-1066 issued by Office of Naval Research International Field Office. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.
# Index

Olaf Lindenau, Volker Bertram  
*The Making of a VRML Model for an SES with Streamlines and Pressure Distribution*  
5

Yvo Saanen  
*Realising Robotised Terminals: A Simulation Supported Approach*  
15

Ivan Siksne-Pedersen  
*Material Management in Shipbuilding*  
29

Zbigniew Pietrzykowski, Janusz Uriasz  
*Methods of Artificial Intelligence in Navigational Safety Assessment of Ship Encounters*  
40

Yury Nechaev, Alexander Degtyarev, Ilia Kirukhin  
*Artificial Neural Network and Genetic Algorithms in Identification Problems and Ship’s Behaviour Forecast in Real Time Intelligence Systems*  
48

Edwin R. Galea, Steve Gwynne, Peter J. Lawrence, Darren Blackshields, John Ewer, Zhaozhi Wang, Nathan Hurst, Nigel Mawhinney  
*The Application of Fire and Evacuation Simulation in Ship Design*  
55

Kenneth Wauchope, Stephanie Everett, David Tate, Tucker Maney  
*Speech-Interactive Virtual Environments for Ship Familiarization*  
70

Ásmund Hellesøy  
*Escape Route Optimization through Evacuation Analysis of a Recent Ro-Pax Design*  
84

Philippe Rigo, Catalin Toderan  
*Least Cost Optimization of a Medium Capacity Gas Carrier*  
94

Jean-Yves Pradillon  
*ODIGO: A Crowd Movement Simulation Tool for Passenger Vessels*  
108

Marcus Bentin, Kunihiro Hamada, Mitsuru Kitamura  
*Multi Objective Optimization for the Master Plan of Cooperative Assembling with Several Shipyards*  
118

David Jaramillo, Tim Meyer-König  
*An XML-based Data Interface for Simulation of Passenger Evacuation*  
133

Herbert J. Koelman  
*Exploring the H-rep Ship Hull Modelling Concept*  
140

Regis Duvgneau, Michel Visonneau  
*Shape Optimization Strategies for Complex Applications in Computational Fluid Dynamics*  
154

Alessandro Giassi, Jean-Jacques Maisonneuve, Fouad Bennis  
*Multidisciplinary Design Optimisation and Robust Design Approaches Applied to the Concurrent Design*  
162

Roman Smierzchalski, Andrzej Lebkowski  
*Search Safe Ship Trajectories in Collision Situation at Sea by Evolutionary Computation*  
175
Lúcia Moreira, Carlos Guedes Soares  
*Training Recurrent Neural Networks with Noisy Data for Manoeuvring Simulation*  
183

Geert J. Tepper, Theodoor de Jonge  
*Modern Procedures in the Structural Design and Production of Ships*  
196

Dirk Steinhauer  
*The Virtual Shipyard – Simulation in Production and Logistics at Flensburger*  
203

Samuel E. Kemp, Paul A. Roach, Andrew J. Ware, Ian D. Wilson  
*Artificial Intelligence for Automatic Container Stowage Planning Optimisation*  
210

David J. Singer, Michael G. Parsons  
*Evaluation of the Effectiveness of a Fuzzy Logic Software Agent to Aid Design Team Negotiation and Communication*  
220

Desmond Simpson, Ludwig Furstenberg, Ektor Kalathas, Volker Bertram  
*Virtual Prototyping for Developing South African Ports*  
235

Bernd Klehn  
*3D-Modelling in Shipbuilding and the Missing Bricks*  
247

Stein Ove Erikstad, Arnulf Hagen  
*Web-based Tendering Collaboration in Project-Centric Industries*  
255

Antonio Rodriguez, Carlos González, Iñigo Gurrea, Lluis Solano  
*Kernel Architecture for the Development of CAD/CAM Applications in Shipbuilding Environments*  
265

Christoph Baier  
*Practical Application of Virtual Reality Technology in Engine Room Design*  
278

Trevor J.T. Whittaker, Björn Elsäßer, Christopher Cain  
*The Development of a Training Software to Reduce the Risk of Wake from Fast Ships*  
284

Martin-Christoph Wanner, Matthias Neumann, Björn Weidemann  
*Internet-Based Cooperation between Supplier and Shipyards in the Maritime Cooperation Network*  
297

Thomas Dirix, Hugo T. Grimmelius, Do (J.T.) Ligtelijn, Joost C. Moulijn  
*Generic Model Building for Concept Exploration*  
308

Thomas Hellström  
*Three Levels of Fuel Optimization at Sea*  
321

Volker Bertram  
*Cyber-Ships – Science Fiction and Reality*  
336

Kees van Es, Martin Th. van Hees  
*Application of Knowledge Management in Conceptual Naval Ship Design*  
350

Yuichi Sasaki  
*Application of Factory Simulation to the Shipyard*  
362

Larry Henesey, Fredrik Wernstedt, Paul Davidson  
*Market-Driven Control in Container Terminal Management*  
377
Virgil Mutu, Ovidiu Ionas, Ionel Gavrilescu
SURF – An In-House Development for a Ship Hull Design Software

Tommi Toivonen
NAPA System Supporting Ship Design

Robert Bronsart, Ulf Cantow, Ventzeslav Petkov
Linking Knowledge to Product Data

Peter Weiss, Frédéric Schom, Yves Chardard
Intervention of Robots in Shipbuilding and Maintenance

José Miguel Varela, Tiago Santos, Carlos Guedes Soares
Simulation of Fluid Dissemination in a Virtual Reality Environment Onboard the Ship

Claus Abt, Stefan Harries, Justus Heimann, Henning Winter
From Redesign to Optimal Hull Lines by means of Parametric Modelling

Bekir S. Türkmen, Osman Turan
Multi-Agent Systems In Ship Design

Kourosh Koushan
Artificial Neural Networks For Prediction of Ship Resistance and Wetted Surface Area

Reinhard Staebler, Matthias Grau
Integration of CAD and CAM Systems in Shipbuilding

Jenny Coenen, Ubald Nienhuis, Reinier Rijke, Jan van der Wagt
Economic Feasibility Prediction for Welding Robots in Shipbuilding

Hugo Grimmelius
Simulation Models in Marine Engineering: From Training to Concept Exploration

Manuel Armada, Pablo Gonzalez de Santos, Teodor Akinfiev, Roemi Fernández
Underwater Climbing Robot for Ship Cleaning

Stefan Krüger
The Role of IT in Shipbuilding

Jong Gye Shin, Kwang Kook Lee, Jong Hun Woo, Jang Hyun Lee, Se Hwan Kim
A Modeling and Simulation of Production Processes at a Virtual Shipyard

Wolfgang Vogel
IT Support for the Integrated Engineering Process of Complex Vessels

Index of authors

Call for Papers COMPIT’04 at last page

Email the authors to obtain an electronic copy often containing figures in colour.
The Making of a VRML Model for an SES with Streamlines and Pressure Distribution

Olaf Lindenau, TU Hamburg-Harburg, lindenau@tu-harburg.de
Volker Bertram, ENSIETA, bertram@waves.insean.it

Abstract

This paper presents a way to built a virtual reality model in the VRML standard from information of a computation with the commercial viscous flow code COMET. The first step is to extract all relevant geometry data from the COMET postprocessor and convert them into the VRML standard for first inspection. As this model needed too much memory and CPU time for rendering, the model needed to be optimized. A program in FORTRAN 77 was written for this purpose. Streamlines were computed with the IBM tool DX. The resulting streamlines were again simplified without any visible losses. For the pressure distribution, the pressure taken from COMET was converted into colors. A program in Fortran 77 for optimization under consideration of geometry and color was created. The VRML model is constructed using shell scripts under a LINUX environment. User interaction in the model is realized with the help of some scripts written in JavaScript.

1. Introduction

CFD (Computational Fluid Dynamics) methods support, to an ever increasing extent, model tests in towing tanks. The predominant role of post-processing for CFD is visualization of the complex flow structures and ship geometries under investigation. The visualization serves a twofold purpose:

1. To aid the CFD expert in understanding the flow and thus derive conclusions on either how to improve his computational model (quality control) or on how to aid the customer with his design.
2. To communicate his findings to the customer, e.g. by pointing out where, how and why to modify a design in order to improve its hydrodynamic characteristics.

State-of-the-art in CFD post-processing are static views. For unsteady problems or complex geometries series of 'snapshots' are used. Animations (avi or mpg files) appear only rarely as they require large storage capacity even for a relatively small resolution and short simulation time. A natural extension in post-processing will be 'Virtual Reality' animations allowing zoom in space and time.

We use the term 'Virtual Reality' here for the "poor man's" version of Virtual Reality, i.e. a usually mouse-controlled navigation through a three-dimensional environment on a graphics monitor. This is far from what fully immersive Virtual Reality envisions, Beier (2000), but can be realized on computer hardware widely available in industry now and is sufficient for our applications.

Following the reasoning of Barcellona and Bertram (2000), we identified VRML (Virtual Reality Modeling Language), www.web3d.org/vrml/vrml.htm, Hartman and Wernicke (1996), as best tool to realize our task as it offered the following features:

- Standardized: VRML97 is ISO/IEC standard, www.vrml.org/Specifications
- Portable: VRML model can be viewed on Windows and Linux platforms with public domain software. The models are small enough to allow easy communication via the internet. However, Linux support is still limited making it as a VRML platform rather cumbersome.
- Cheap: VRML models can be created using any editor and run on plain PCs, avoiding thus costly investment in hardware, software or training.
- Easy: The modeling language is relatively easy to learn, allowing modification and maintenance of models by "plain" engineers.
As a prototype demonstrator, we decided to use the data of the commercial CFD code COMET to build a VRML model for the visualization of the aerodynamic flow around a surface effect ship (SES). The main focus was to interface the programs such that an efficient VRML model reflecting the needs for practical purposes could be created in short time. A mere formal export of data is already often supported by many codes, but results in practically useless Virtual Reality models which are far too cumbersome.

2. **Creation of the VRML model**

2.1. **Getting started: Hull geometry model**

Recommended introductory papers on VRML applications for naval architects are Beier (2000), Barcelona and Bertram (2000). A good way of getting started with building VRML models is to learn from related examples, e.g. [http://www-vrl.umich.edu, http://www.ifss.tu-hamburg.de/IFS/AB/AB-3-13/ab313.html](http://www-vrl.umich.edu, http://www.ifss.tu-hamburg.de/IFS/AB/AB-3-13/ab313.html).

CFD codes (in our case Comet) describe the hull geometry by nodes connected to form hull surface patches. If the geometry is defined in several blocks (block-structured grid), it is recommended to export the geometry for each block separately. This makes data handling easier and reduces the CPU time for the “optimization” (= reduction of the level of detail required). We describe here the specific commands using the Comet pre- and postprocessor Cometpp. For other codes, similar steps are straight-forward:

1. Write in Cometpp the vertices on the surface of the geometry into a file. Here an example for the region 1 defining the surface:

   ```
   rset news type 1
   vset news rset
   vset inve
   vdel vset
   vset all
   vwrite (default is CaseName.Vert)
   ```

   Leave Cometpp with the command **quit nosave** to avoid loss of data. The resulting file has the format: `VertexIndex X Y Z`

2. Create the topology (connections between vertices) in Cometpp:

   ```
   rset news type 1
   rset inve
   rdel rset
   rset all
   rwrite (default is CaseName.Regi)
   ```

   Again leave Cometpp with the command **quit nosave** to avoid loss of data. The resulting file has the format: `RegionIndex V1 V2 V3 V4 RegionType`

3. A self-written Fortran program `comet2vrm.lfor` converts the Comet geometry data into VRML standard. This program reads the two files `CaseName.Vert` and `CaseName.Regi` and has as output two files `CaseName.point` and `CaseName.face`. The `CaseName.point` file has the format `x y z`. In the file `CaseName.face` the connections between the points are given according to the VRML standard (p1 p2 p3 p4 -1).

4. Finally a first VRML geometry model is put together concatenating the individual blocks. This can be realized using a script basing on the command `cat` in the Linux environment or an editor resulting in a file defining a geometry as “IndexedFaceSet” is written. Navigation through the model
will usually be impossible on a normal PC, as the geometry needs too much memory and too much CPU time for rendering. But the static views are sufficient for a first inspection.

2.2. Reducing polygon counts

The FORTRAN 77 program `geocomp.for` reduces the polygon count by getting rid of superfluous points and merging faces together which lie approximately in one plane. The polygon count determines the file size and rendering time. The rendering time can be reduced even more by defining all face counter clockwise (ccw), when seen from the outside. Then only the outside of an object is displayed and the inside is left out and therefore needs no rendering. The last step in the downsizing process is to round off the coordinates given as floating point numbers to a level which is appropriate for the size of the object.

To start the downsizing process all relevant parameters should be provided in an input file which is directed to the program (`geocomp < geocomp.in`). The following input parameters are needed (here as example for the antenna of the SES):

```
ant.point                        INPUT POINT-FILENAME
ant.face                         INPUT COORDINDEX-FILENAME
-0.001                           EPSILON FOR POINT MERGING
0.000001                         EPSILON FOR POINTS
0.0001                           EPSILON FOR FACES
ant.inside                       INSIDE-POINTS-FILENAME
ant_new.point                   OUTPUT POINT-FILENAME
ant_new.face                    OUTPUT COORDINDEX-FILENAME_1
ant_new.ils                     OUTPUT COORDINDEX-FILENAME_2
```

Input and output files for the points and the faces have basically the same format. The VRML format for points are three floating point numbers per line of the file giving coordinates for NPOINT+1 points starting with coordinate for point 0 and ending with the coordinate for point NPOINT. The index of the points is implicitly given through the order of the points in the file. The VRML format for the faces in an “IndexedFaceSet” is a list of integer numbers referring to the index of the points, ending for every face with “-1” (also see the example of a VRML model). Additionally, the faces are written in the format used to display an “IndexedLineSet”, showing a wire model of the shape after downsizing by repeating the index of the first point in a face at the end of a line before the “-1”.

After reading all input parameters listed above, the program checks whether points in the point list lie too close to each other. The distance ε for merging the points is read from the input file. If a negative distance is given this step is skipped (This routine is not very CPU time consuming, but for matching CFD grids it should be superfluous). The next optional routine checks whether all faces are defined ccw (counter-clockwise). This check is based on user-defined points inside the body. (In our example, the inside of the SES was defined using 42 points and for the antenna 5 points.) Most CFD grids define only one half side of the ship and then mirrored to the other side. Then half of the face definitions have to be changed in orientation.

Now the normals for the faces are calculated. This is straight-forward for triangular faces. Quadrilateral faces are split into two triangles if the normals for these two triangles differ by more than a given threshold. This is important for rather coarse initial grids.

To reduce the polygon count, neighboring faces lying in the same plane (i.e. with almost collinear normals) are merged. The normal of the first face is kept as normal of the new merged face. After the merging, all inner borders have to be removed. Likewise, collinear lines are merged by deleting the middle points lying on a line. The merging process for all faces stops when no further faces for merging can be identified. A final check examines each face for any superfluous points removing now the
restriction that there might be a neighbor. The overall process takes some time depending on the number of faces to be checked. Finally the resulting points and faces are written to output files in the format described above.

The Fortran 77 source code is documented with many comments and will be emailed upon request like the VRML models.

For the SES, the procedure reduced the original model from 43000 polygons (2810 Kbyte) to 900 polygons (130 Kbyte), Fig.1. The reduced model exploits the smoothing capabilities of VRML to create again a realistic display.

![Fig.1: Original model exported from CFD grid (left) and model with automatically reduced polygon count (right)](image)

**2.3. Streamline model**

The calculation of streamlines is rather clumsy in Comet. We employed instead the DX of IBM, a public-domain tool package within the SUSE Linux distribution. The steps are similar as for the hull geometry:

1. Write in Cometpp the vertices on the surface of the geometry into a file. Here an example for the region 1 and 2 defining the surface:

   ```
   rset news type 1
   rset add type 2
   rset inve
   rdel rset
   rset all
   rwrite   (default is CaseName.Regi)
   ```

   Leave Cometpp with the command `quit nosave` to avoid loss of data.

2. Write from Cometpp a file with data for the commercial plot program TecPlot with the command:

   ```
   tecplot
   ```

   The file `CaseName.tecp` has the data for all cells and on the boundary.

3. Using the program `convert2DX` (written by Dieke Haermann, HSVA) the data are transferred from TecPlot format to DX format:

   ```
   convert2dx CaseName.tecp CaseName.Regi Number_boundary_elements
   ```
The result are the files `bnd.dat` (with the calculated data from Comet) and `tmp.dx` (with a header for DX describing the type and location of data in `bnd.dat`). For details on the source code contact hafermann@hsva.de

4. Calculation of streamlines in DX specifying different predefined viewpoints, starting points of streamlines and formats of output. The streamlines are displayed on the screen and can be output into a file as coordinates. These files are used to build a VRML model of the streamlines.

5. The defining coordinates of the streamlines are reduced with the program `streamcomp.for`. The merging is rough and primitive, but reduces the size to approximately 1/20 of the original size.

6. The streamlines are combined in a VRML model using a script based on the command `cat` in Linux writing a file defining the streamline geometry as “Extrusion” in VRML. Alternatively this step could be realized using an editor under Windows.

For the VRML model, 32 streamlines were included, using a matrix of 4 columns and 8 rows.

2.4. Interactivity

Two methods of interaction were implemented in the model – the first for blending in and out of streamlines and a second for highlighting a streamline which is under the mouse cursor. As these are already more complex forms of interaction, scripts in JavaScript or Java have to be used. “JavaScript is a scripting language, developed by Netscape Communications, that uses a somewhat Java like syntax. Despite some similarities, however, Java and JavaScript are two different languages. JavaScript is easier to use, especially for small tasks; Java is more powerful and more general.”, Hartman and Wernecke (1996). Because only some logical operations must be performed in the script, JavaScript was chosen here. The script for blending in and out is a modified example and the script for the highlighting was directly taken from Hartman and Wernecke (1996). As the scripts are stored in an extra file, they can easily be taken for another VRML model.

Fig. 2: VRML model with streamline matrix

To activate streamlines it is best to switch to the viewpoint “Front”. A matrix of bullets is shown, Fig.2. The blue bullets are toggles at the position where the 32 streamlines start. Click with the cursor on one of the switches and the streamline will appear/disappear. The red switches to the left switch rows of streamlines on and off. The green switches at the bottom of the matrix switch columns of streamlines on and off. Pass with the cursor over one of the streamlines to turn it red, Fig.3. One can then follow it in a bundle of streamlines. Once streamlines are selected, one can use the usual handles of the VRML browser to zoom, rotate and pan the SES together with the streamlines.
As in all VRML models, pre-specified viewpoints can be selected interactively from the browser console. These viewpoints are given in an extra file. They can be used easily in another VRML model. The viewpoints were programmed using the shareware tool Peek v1.0, http://www.vapourtech.com.

Fig.3: Streamlines with one selected streamline in red

Fig.3: Pressure wallpaper mapping, Barcellona and Bertram (2000)  
Fig.4: Pressure interpolation in VRML

2.5. Pressure distribution

There are various ways to realize a surface color distribution (e.g. representing pressures) in VRML:

1. Map a 2D image (e.g. a jpg file) onto the object’s surface stretching the image over the whole geometry – like putting wallpaper on a wall. This simple approach works nicely for simple geometries, Fig.4, Barcellona and Bertram (2000). For real ship geometries, the mapping is at least very complex or even impossible.

2. Map wallpapers for every face of the geometry. Then the problem is creating wallpapers for every face and connecting all images with the correct faces. This approach requires dedicated software and creates also rather large models.

3. Each face can be associated with one color from the color list of VRML. This approach is again simple and results in acceptable representations for models with fine grids, Bertram et al. (2002). For models with reduced polygon counts, this approach is not appropriate.

4. A better approach is to give every point of the geometry a color taking advantage of the capability of the VRML browser to interpolate colors. We developed a prototype program colocomp.fort, based on the program geocomp.fort, to reduce the polygon count in VRML geometries taking into account the colors in the points. Although the program works, the effectiveness shows potential for improvement. Only a reduction of a factor 3 was achieved so far. Only faces where all colors are the same are merged. In addition, the original pressure data set oscillated slightly along the surface
making straight-forward reduction difficult. Intelligent smoothing, avoiding significant loss of physical information, will require further research.

We employed the 4. approach which resulted in a realistic and detailed pressure distribution, Fig.5, but the algorithm needs to be required to obtain models with sufficiently fast response on standard PCs. We hope that commercial software will be available in the future to support this task.

3. Conclusion

Virtual Reality modeling with VRML offers simple and cheap engineering support solutions for CFD post-processing. While our applications admittedly fall short of ambitious “real” Virtual Reality applications (see e.g. RWTH Aachen, http://www.rwth-aachen.de/vrca/Projekte/stroe.html), the combination of cost, quality, and time aspects lets VRML appear at present still as our preferred choice.

VRML is relatively easy to learn and VRML geometry data is easy to export from CFD grid descriptions. Commercial CFD codes like FLUENT and CFX allow already direct geometry export to VRML format. However, CFD models and virtual reality models have different requirements and intelligent interfaces require some effort and tailored software. The benefits are sufficient to justify such development.

To illustrate the need of post-processing it is helpful to take a look at some file sizes: Comet needed 533 MB disk space for the data of the CFD calculation and 1.1 GB (!) for the geometry of a grid with 3 million cells. The file for TecPlot had a size of 486 MB and the file for DX only 188 MB having as data x, y, z, p, u, v, w (geometry, pressure, velocity). Compared to this, the final VRML model containing the geometry and the streamlines with 324 KB is really small.

Following our examples, creating a similar interfacing chain may require an estimated 2 weeks. Once the chain is established, the actual creation of new models for given geometry and CFD data takes at most half a day.

References


Appendix I: VRML code for example

The example illustrates defining a geometry as “IndexedFaceSet” (complex shapes like ships) and as “Extrusion” (long shapes like streamlines) in VRML. Also included are two scripts for activating and highlighting a streamline. The VRML model for the SES is based on this example. The VRML model is split into different files for better handling and possible reuse. In addition a list of predefined viewpoints is provided in one file. This example can easily be modified for a different ship.

```vrml
#VRML V2.0 utf8

# This file is an example for the definition of
# 1.) a geometry as IndexedFaceSet
# 2.) a geometry as Extrusion
# 3.) a script for switching a geometry on and off
# 4.) a script for highlighting a geometry by passing
#     over it with the mouse device

# Definition of a shape as IndexedFaceSet
Shape {   # cube
appearance Appearance {
    material Material { diffuseColor 1.0 1.0 0.0 }
}  
geometry IndexedFaceSet {
    coord Coordinate { point [
        0.0 0.0 0.0
        2.0 0.0 0.0
        2.0 0.0 2.0
        0.0 0.0 2.0
        0.0 2.0 0.0
        2.0 2.0 0.0
        2.0 2.0 2.0
        0.0 2.0 2.0
    ]
}
    coordIndex [
        0 1 2 3 -1
        1 5 6 2 -1
        5 4 7 6 -1
        4 0 3 7 -1
        4 5 1 0 -1
        3 2 6 7 -1
    ]
}
# Definition of the geometry of a streamline as extrusion
PROTO Stream1 [
    exposedField SFColor color 0.8 0.8 0.8
] { 
    Shape {
        appearance Appearance {
            material Material { diffuseColor IS color }
        }
        geometry Extrusion {
            spine [
                -2.0 0.5 1.0
                -1.0 0.5 1.0
                -1.0 0.5 2.5
                3.0 0.5 2.5
                3.0 0.5 0.0
                4.0 0.5 0.0
            ]
            crossSection []
            scale [0.1 0.1]
            creaseAngle 2.0
        }
    }
}
```
# This part defines the used scripts
#------------------------------------------------------
# Activater = PROTO for switching an object on and off
PROTO Activater {
  eventIn SFBool isActive
  eventOut SFInt32 whichChoice
  field SFBool isShown FALSE

  {Script {
    eventIn SFBool isActive IS isActive
    eventOut SFInt32 whichChoice IS whichChoice
    field SFBool isShown IS isShown
    url [
      "javascript:
        function isActive(eventValue) {
          if (isShown == false) {
            if (eventValue == true) {
              whichChoice = 0
              isShown=true }
            else
                if (eventValue == true) {
                  whichChoice = -1
              isShown=false }
        }
      "
    ]
  } }
#------------------------------------------------------
# Highlighter = PROTO for highlighting an object
PROTO Highlighter {
  eventIn SFBool isActive
  eventOut SFColor color
  field SFColor activeColor   1.0 0.0 0.0
  field SFColor inactiveColor 0.8 0.8 0.8

  {Script {
    eventIn SFBool isActive IS isActive
    eventOut SFColor color IS color
    field SFColor activeColor IS activeColor
    field SFColor inactiveColor IS inactiveColor
    url [
      "javascript:
        function isActive(eventValue) {
          if (eventValue == true )
            color = activeColor;
          else
            color = inactiveColor;
        }
      "
    ]
  } }
#------------------------------------------------------
# Definition of a switch to blend a streamline in and out
#------------------------------------------------------
Transform {
  translation 1.0 1.0 4
  children [
    Shape {
      geometry Sphere { radius 0.1 }
      appearance Appearance {
        material Material { diffuseColor 0.0 0.0 1.0 } }
    } DEF Switch1_TS TouchSensor { } ]
}
#------------------------------------------------------
# Definition of a streamline and a switch in one group,
# so the streamline is highlighted when the mouse device passes over the streamline.
#------------------------------------------------------
Group {
  children [
    DEF Stream1_S Switch { choice [ DEF Stream1_color Stream1 { } ] }
    DEF Stream1_TS TouchSensor { } ]
}
#------------------------------------------------------
The interaction of the mouse device with the VRML model is realized by routing the events from the switches to the scripts and then to the geometry.

```vrl
DEF Stream1_High Highlighter { }
ROUTE Stream1_TS.isOver TO Stream1_High.isActive
ROUTE Stream1_High.color TO Stream1_color.color

DEF Switch1_Act Activater { }
ROUTE Switch1_TS.isActive TO Switch1_Act.isActive
ROUTE Switch1_Act.whichChoice TO Stream1_S.whichChoice
```

Appendix II: The files for the VRML model

The directory for the VRML model of the optimized SES with 32 streamlines contains files of a total size of 313 KB.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0degstream.wrl</td>
<td>190 KB</td>
<td>This file contains the 32 streamlines. The geometry is defined as “Extrusion” and the single streamlines are declared as “PROTO”. This definition enables a change in color through the main program.</td>
</tr>
<tr>
<td>ant.wrl</td>
<td>26 KB</td>
<td>This file defines the antenna as “IndexedFaceSet” and declares it as “PROTO”. 589 points and 320 faces are used.</td>
</tr>
<tr>
<td>hull.wrl</td>
<td>61 KB</td>
<td>This file defines the hull as “IndexedFaceSet” and declares it as “PROTO”. 1338 points and 586 faces are used.</td>
</tr>
<tr>
<td>Main_ses.wrl</td>
<td>45 KB</td>
<td>Load this file to start the VRML world of the SES. This file links all parts of the VRML world together.</td>
</tr>
<tr>
<td>MyProtos.wrl</td>
<td>2 KB</td>
<td>This file contains the two scripts for the blending in and out of an object and for the highlighting of an object. They are defined as “PROTO” and are used several times in the main program.</td>
</tr>
<tr>
<td>ses_viewp.wrl</td>
<td>2 KB</td>
<td>This file defines a list of predefined viewpoints for quick and easy navigation through the VRML model.</td>
</tr>
</tbody>
</table>
Realising Robotised Terminals: A Simulation Supported Approach

Yvo Saanen, TBA Nederland/Delft University of Technology, yvo@tba.nl

In order to meet the demands from today’s customers, container terminals have to increase their productivity with a significant rate. Besides, they are confronted with ever-increasing wages. Therefore in all new container terminal development projects robotisation of operations is considered as a serious option to cope with both demands: higher productivity and lower cost.

Simply increasing equipment performance or putting in more equipment will not accomplish higher productivity at a lower cost. It requires rethinking of the entire terminal concept. When designing a robotised terminal an integrated design approach is required, combining hardware (equipment) and software design. Furthermore it should integrate the different design tasks such as functional design, technical design and realisation.

A key element in a robotised terminal is the process control system, which controls and co-ordinates all functions in the terminal. Especially the design of this real-time planning and control system requires serious attention.

The approach needs supporting tools that address the specific characteristics of processes at a container terminal – high impact of external influences, high degree of uncertainty, many disturbances – and are able to provide this support throughout all design activities, including realisation. A design-supporting tool that has proven to be applicable for that is an environment of various simulation models.

Simulation is used to support the entire design process, especially focussing on the interaction between hardware and software. Within the simulation environment prototypes of the real production software are developed and production software is tested using the simulation environment.

The design approach and support environment have been applied in a number of real-life cases and have shown to improve insight for all people involved including technical and managerial staff. It also provided early support for equipment design and software testing.

1 New era in development of process control software for robotised container terminals

1.1 Introduction

The design of process control software for large, complex logistical systems is a process with many hurdles. Still in many projects requirements are not met, functionality is not as the customer expected and software is never free of bugs. It is even so that it does not cause any raised eyebrows or worse when an exceeding of a software implementation project is announced: it is an accepted fact.

In this paper we will describe a set of guidelines that support the design of a process control system (PCS) – a key part of an automated system – for a robotised maritime container terminal. Applying these guidelines decreases the chance of a mismatch between functional and technical specification of the actual implementation. Furthermore it provides the possibility to test software in a much earlier stage. It also enables us to test software on its functionality and performance as if it had already been implemented. Finally, it creates insight for people involved in the new system in an early stage.

The outline of the paper is as follows. In this section we elaborate on the specific characteristics of the container terminal industry leading to the automation of terminal. Furthermore, we describe the problems that arise when designing and realising an automated terminal. In section 2, we propose a set of design guidelines that meet the specific problems during a design-engineering trajectory of an automated terminal. In section 3, we focus on the design and implementation of the heart of an automated terminal: the PCS. In section 4, we then zoom out to tools that can be applied throughout
the design process, mentioning a number of examples of typical questions that rise. In the final section (4) we pay attention to the current accomplishments and the next steps to take.

1.2 An increasing need for automation

It is already 16 years ago that Europe Combined Terminals (ECT) in Rotterdam started its robotisation project of the Delta terminal on the Maasvlakte. In 1993 the final commissioning of the system was a fact, the first terminal with automated yard and automated transportation system was in operation. Since then a number of terminal in the world have applied robotised yards as well – for instance Thames Port in London and the Pasir Panjang Terminal in Singapore. The most recent case is the newly realised container terminal Altenwerder in Hamburg, a state-of-the-art terminal with an automated yard and an automated transportation system.

There are a number of reasons for the increasing need to automate processes at maritime container terminals. First of all, due to labour scarcity, labour costs are rising, putting pressure on the profit margins. Secondly, the scale of deep-sea shipping has increased enormously since the sixties (see Figure 1), probably reaching the 10,000 TEU limit in the next few years. This does not remain without consequences for the maritime terminal operators, since they have to increase their productivity to handle the larger vessels within the required time-windows, shipping lines have set.

Large-scale robotisation implies large-scale automation, because at the control level tasks have to be carried out that were done by people; drivers and personnel executing the transhipment processes. Partly these tasks have been replaced by control room tasks; partly computer systems have taken over the work. The type of systems needed to operate and control automated equipment has to be categorised as very complex, even in comparison with large administrative systems. There are several reasons for this classification.

- First, the control tasks have to be executed in real-time, which means that fast communication, possible multi-tasking, and reliable response times are obligatory.
- Secondly most communication with the equipment is done via radio networks, and therefore the communication may not be too extensive, whilst the systems dimensions – number of
equipment, number of interaction moments – are very large and communication load can vary a lot.

- Thirdly it requires extensive planning to meet the requirements of the shipping lines regarding the sequence of containers and the door direction of containers, especially – and that is the fourth reason – taking into account the often bad information that has bee provided, about container weight, final destination, et cetera.

- The last reason, but certainly not that of least importance, is that the processes that are still manually operated – operation of the ship-to-shore crane and the handling of hauliers, or landside trucks – contain large unpredictable variations in terms of cycle times and arrival patterns. For instance a common cycle time distribution of a quay crane shows a variation between 40 seconds and 450 seconds, without any possibility to foresee long cycle times. The process control system has to be able to deal with these variations in such a way that the performance does not suffer from it.

At the moment there are no standardised packages to control a robotised container terminal. Of course, on the global market there are two major players – and a number of smaller other providers, such as TSB –, providing terminal planning and control software – Navis with Sparcs/Express and Cosmos with Space/Traffic – but they lack the control of fleets of robotised equipment and the special precautions needed for this type of equipment. Up to now, the terminals that automated part of their processes, developed the specific software needed for controlling the fleet of automated equipment themselves or had dedicated adjustments made to existing packages.

Figure 2: Snapshot from a simulation model of the DDW terminal of ECT; 1.2 km quay, maximum of 8 quay cranes, 32 RMGs 1 over 3, 6 wide, transportation by means of AGVs.

These development processes were not always as successful as hoped for, with pioneer ECT as the exemplary example (see Figure 2). A couple of reasons can be mentioned to explain the reason for the relative lack of success. At first, the technology used twelve years ago, especially concerning computing power, was many times lower than today’s. Secondly there were no best cases present, so everything was new. But an important reason was also that the implemented system was not according to the functional specification. Even now, more than ten years later, some originally specified functionalities are not put into practice.

Other reasons are less obvious, but still important to take into account:

- The operation at a terminal with perpendicular RMGs and AGVs (or possibly: shuttle carriers, see Saanen, 2002; see also Figure 2) is more dependent on its PCS than an RTG-TT or a straddle carrier terminal, because the capacity on a specific section of the yard is limited. This requires a look-ahead function in the yard planning to distribute containers for a certain vessel
over the yard. It also requires real-time re-planning of the stowage plan in order to spread the workload of the RMGs.

- The interaction between human operators and the automated system is quite complicated in the sense that the operators should be able to oversee the consequences of changes in the parameters of the PCS. This requires adequate feedback of the automated system. Second, in the event of a disturbance, the PCS should be so flexible that the operator can act according to the situation.

1.3 Automated terminals require a different design approach

Which special precautions have to be taken when designing and realising an automated terminal? There are several important differences – from an operational point of view, of course there are other differences, cost structure for instance, but these will not be discussed here – between manually operated terminals and automated terminals.

- First, there is an important difference regarding the intelligence of drivers. Manual terminal truck drivers or straddle carrier drivers optimise their processes themselves, solving local problems like traffic jams, shortest routes, crane positions, et cetera, whilst automatic equipment, such as automatic guided vehicles (AGVs) have to be controlled without that bit of local intelligence that extends the strict rules of the AGV router for example. So the flexibility at the lowest level is – up to now – less in an automated environment.

- The second difference concerns the flexibility in the assignment of the pool of equipment. Whereas it is common to dedicate a number of straddle carriers or terminal trucks to a quay crane, AGVs or automatic lifting vehicles (ALVs) are much more freely assigned, because their internal route map will tell the exact position of quay crane so and so, but the truck drivers have difficulties in keeping track of the ongoing changes. Often the flexibility – or better said, the lack of flexibility – built in the control system does not justice the qualities of the equipment present, such as AGVs and ALVs (Wysk, 2001).

- A third difference concerns the operation of the yard: housekeeping operations during quiet hours are relatively cheap in an automated setting, but require manning if not. So the yard can be prepared much better with automated equipment with relatively low cost increase – only electricity and maintenance cost increase. But it requires planning, both middle-long term planning – 8 hours to 7 days as horizon – as real-time planning. A fourth difference is the need for co-ordination between operations. Especially when the different operations, i.e. horizontal transport, yard operation and quay operation are linked to each other, as is the case at ECT, the need for co-ordination – scheduling equipment rendezvous – is high.

Summarising we can say that at the planning and scheduling level we need mechanisms that allow for preplanning with a middle-long term horizon, say 24-48 hours, and real-time planning for coping with variations. We also need robust but flexible low-level control of the fleet of automated equipment. Both things, for different reasons, are less needed in the control system of a manually operated terminal.

What kind of requirements do these changes bring up for the design process? And secondly, how can we cope with these requirements during the design process? Third, how can be guarantee or at least ensure that the functional specification is met during the implementation?

2 Design guidelines to improve the design-engineering of an automated terminal

2.1 Using simulation throughout design-engineering process

Having experienced the differences between the operation at a manually operated terminal - its flexibility, little degree of control at the execution level, the many just-in-time decisions, but also the inefficiency -, and an automated terminal – long-term planning, inflexible short-term interventions,
sensibility to break-downs, and hampered interaction between operator and system, we have
developed a number of design guidelines, that could improve the design-engineering process, and as a
result the operations.

The design-engineering process of an (automated) container terminal could typically be divided in
four types of activities – in system design literature (e.g. Roozenburg and Eckels, Pahl and Beitz) the
same activities are divided over different phases, but can all be covered by the following activities
(Saanen, Verbraeck and Rijnenbrij, 2000):

1. Conceptual (or functional) design.
2. Technical design.
3. Implementation and realisation.
4. Commissioning and operations.

These four activities are not necessarily executed sequentially, in practice there probably will be
significant overlap and iterative feedback loops. Nevertheless, the type of problems that have to be
solved differs between the phases. The differences can be categorised into three categories:

– Contents of problems to be solved (aggregation level, type of questions).
– Information available (problem space, solution space).
– Type of people involved (from managers to technical and operational people).

The main guideline that we propose is to support the entire design-engineering process by means of
discrete event simulation (DES), i.e. all four activities. In order to make the different applications
throughout this process of DES clear, we have to define first what we mean with simulation.
Simulation is the process of applying dynamic models in the representation of reality. Those models
have a certain degree of reduction (leaving non-relevant details away), depending on their purpose.
Besides simulation, we discern emulation, which can be defined as running production software in
virtual environment, i.e. without real equipment but with simulated equipment instead (Rengelink and
Saanen, 2002). Emulation is applied often when testing software: the software is run without the real
equipment. So the main difference we make between simulation and emulation is that the
representation of the information system in simulation is of an aggregate level, whilst in emulation the
representation is identical. However, in both applications, an aggregate model represents the real
system (equipment).

Traditionally DES is used as support tool during the conceptual and functional design phase, typically
answering what-if questions (Zeigler, 1976). Examples are: “what performance increase is
accomplished if instead of 3, 4 straddle carriers are used on one quay crane?” or “how much does the
berth utilisation increase when the average call size is increased with 20%?”.

The second type of simulation concerns detailed simulations of certain system components, for
example simulations of the dynamic behaviour of a crane or a vehicle movement. Using these models,
for instance the effects of a higher hoist speed or traverse speed can be determined. We found that this
kind of simulations, with an aggregate model of the whole system in which the component – modelled
by means of either a mathematical representation of the dynamic behaviour or an exact representation
of the behaviour – fulfils a function, is not commonly applied. This type of application can be used
during functional design, technical design and during the implementation, especially of the PCS.

The third type of application that is addressed by simulation is the testing of software under
conditioned circumstances. In this case production software is run without the real environment it
normally runs in, i.e. the user or the equipment are simulated. The software itself is emulated, i.e. it is
a copy of the real software that is executed. It might also be the case that only a part of the real
software is running, the other parts are still running in an aggregate form (Saanen and Franzke, 2000).
This type of application is important during the development of the software of the PCS and during
the commissioning of the terminal.
A fourth type of application (see e.g. McConnell and Medeiros, 1992 Rogers and Gordon, 1992, Ruiz-Torres and Nakatani, 1998) is the use of simulation as support tool during execution of the normal operation. In most of these applications the simulation model is used as real-time decision support, for instance to determine to support the decision which course of action to take, or which planning to use; especially in case of irregular situations that did not occur yet, this kind of decision support is very useful. In these cases the simulation is fed with real data and then runs the operation in fast mode so that the future effect of the different possible decisions can be seen. Possible applications are berth planning and stack planning.

Although all four applications can be found in literature, with most examples of traditional simulation of course, hardly any evidence can be found concerning the application of simulation throughout the entire design process. However the idea of using simulation techniques throughout the entire design process is supported by Fishwick. He states (1992) that “the ultimate simulation is the actual creation of the software which is then executed; lumped statistical behaviours can easily be obtained when one has the lowest level performance traces. The ultimate software development tools is one that permits modelling the software at a variety of abstraction levels – the lowest level providing detailed behaviour of the sort normally associated with program input and output traces. Given this view, simulation is the process of creating abstract versions of programs; the final most detailed simulation is simply the executable program.”

2.2 Benefits of using simulation throughout the design-engineering process

This idea of using simulation tools throughout the design and development process of automated container terminals aims at a number of goals:

1. The first objective is to create a support environment that is a consistent factor throughout the entire process. Its visualisation and computing possibilities create a basis for understanding processes and quantifying different solutions, instead of educated guessing. The consistency objective is especially important, because during a (long) process as the design of a container terminal is, there are several sources of inconsistency. First of all the people involved, change during the project. Secondly the content changes, both regarding the level of detail, and the type of issues to be addressed, solved or discussed. Thirdly the external conditions, influencing the process, change, which could cause the process to go into another direction.

2. The second objective is to create a support environment, which builds up during the process with ideas, concepts, alternatives, either chosen or discarded, but at least evaluated. In that way the simulation environment also serves as a process history, which can easily be recapitulated whenever the decisions change. An additional advantage of building up an environment, instead of using one-shot models in the different project phases, is that assumptions earlier made, can be validated in a later phase. Whenever some assumptions prove to be less valid, one can go back and reconsider the choices made then. Finally it creates a basis to build upon, because certain relationships have been sought out sometime, which can be used in a later phase.

3. The third objective is to connect the different phases of the design process more to each other. Because of the reasons mentioned under 1) the different phases in the design project often contain a number of gaps, which cause a final result that is worse than originally expected and not meeting the functional specification. We experienced this in several projects, especially when the functional specification has to be translated into a technical specification and when the technical specification has to be transformed into a real system containing hard- and – maybe even more important considering an automated container terminal – software. When a single simulation environment is used throughout the process, i.e. containing the translation of functional requirements into rough concepts, then translated into prototypes at a technical specification level, which then can be used during the implementation and testing phases, there is a ongoing connection in between thinly connected processes.

4. The fourth objective is to create a support environment, which is capable to determine the complex relations within a robotised terminal in order to optimise the logistic control system. It
is not only case to determine the number of equipment, but also the equipment specification. Furthermore it is interesting to find a right balance between hardware and software solutions. A combination for instance of rail mounted gantry cranes to operate the yard and automatic guided vehicle to perform the horizontal transportation can be a very efficient one, if the equipment control is adjusted so that waiting times are minimal and cycle times are tuned to the quay crane demands. By prototyping the control algorithms in a simulation model all the processes are dynamically linked to one another, so that one can observe the dynamics and the robustness of the interfaces.

5. Last objective is to improve the software development process by creating continuous insight in the interaction between components and the performance of the whole. Especially when the control systems get more complex, parallel development and a distributed, decentralised architecture of software is mostly needed and increasingly applied. However, the performance as a whole is hard to assess during the development, since the entire system is not ready yet. We have experienced that the use of already tuned and validated simulation models as support tool during the software development and testing process reduces development time and deviations from original concepts.

Figure 3: Two examples of claiming algorithms; upper picture shows optimised claims, lower picture shows standard claims that hinder other vehicles (Saanen and De Waal, 2001).

2.3 Guidelines to enable the use of simulation throughout the design-engineering process

In order to realise the general approach described above, we recommend applying a number of other guidelines. These guidelines are supportive to the guideline to apply simulation throughout the design-engineering process for, as discussed, various purposes.

2.3.1 Holistic, but layered view on the terminal processes

We propose to analyse the container terminal from a holistic perspective, taking all processes between the terminal boundaries into consideration. Of course, some processes require more attention than others, but due to many changes, some processes, which at first seem unimportant, may very well influence the system as a whole. In order to keep the design process manageable, we also apply
different hierarchical abstractions levels in our analyses and models. Depending on the design activity, we focus on a specific terminal process or component.

2.3.2 Object-oriented view on the real world

We propose to use the object-oriented modelling paradigm, which means that the entities that execute actions are leading. The object-oriented modelling paradigm has a number of advantages (Booch, 1986), (Rumbaugh et al., 1991), (Schlaer and Mellor, 1998), which make this way of viewing the world suitable for a terminal design process. When the object-oriented way of modelling is compared with the flow-oriented way of modelling, the advantage appears in the fact that there are many different processes (flows) throughout the terminal dependent on internal and external conditions, not known at the time of arrival. However, the actions that can be performed by the entities (equipment, terminal personnel, customers) are known and defined. These two aspects make it easier to conceptualise a terminal in an object-oriented way than in a flow-oriented way. Moreover, in the case of robotised container terminals, the use of an object-oriented view of reality eases the conceptualisation of the control software, because most control software is object-oriented and, therefore, the conceptual description is much closer to the implementation in software.

2.3.3 Explicitly taking uncertainty and process variability into account

A dominant property of a container terminal is the lack of deterministic elements, as already has been argued. The influence of external processes is high, the information present is of a poor quality or missing, and the variation in behaviour of terminal processes is relatively high due to unreliable manual operations or equipment failure. In order to create a design that also works in practice the design has to address the dynamic system behaviour of the real system. Therefore, our guideline is to take the variation explicitly into account when modelling and analysing the system. We prefer this approach above an approach in which the variability is averaged and the outcome is increased with a certain safety margin to cover peaks. Explicitly modelling the variation of process behaviour requires a better knowledge about the range of outcomes of each process, because not only the (estimated) average is required, but also the minimum, maximum and relative frequency of all outcomes are required. The choice to model the variation in an explicit way has consequences for the solutions that can be applied, especially in the area of optimisation and control algorithms. Usually optimisation algorithms (such as the Hungarian algorithm) treat information as certain. Therefore, in order to be able to use these optimisation algorithms, continuous re-planning based on the actual available information is required. Only then, the information used as input for the optimisation can be considered as relatively certain.

2.3.4 Operational processes as leitmotiv for the design

A well-organised operational process is the key to success at a container terminal. Therefore, the system design should be determined by the operational processes and not vice versa. The knowledge about the operational processes is the key to success, because the container business is full of exceptions. The operational processes cannot be changed easily, because the container terminal under consideration is only a single link in a worldwide network of terminals and changes at a single location could require changes all over the world. Therefore, we define the guideline to take the operational process as leading does not mean that it cannot change, but the boundaries are set. Examples are, for instance, the existence of a large variety of twistlocks used to connect containers, the existence of off-standard containers or the existence of (local) time-windows for freight transportation, which increase the peak loads landside.

Design decisions, however, that do not influence the operations at other terminals or on board vessels, trains or trucks, should be made independent of the current operation, especially when it concerns automation. Because most terminals are manually operated, the design of a robotised terminal must consider these operational processes. Many manually operated terminals operate in a similar manner, with similar yard strategies and equipment control strategies that are mostly aimed at limiting the number of additional moves of equipment in order to save on labour costs. However, this key design
parameter is not valid in a robotised terminal, where particularly equipment control is automated, diminishing the cost of an additional move (Dobner et al., 2001).

2.3.5 Integrate the design of manual operations and automated operations

Although many processes are executed automatically at a robotised terminal, manual operations will remain necessary, because of the many physical differences between the equipment in operation (vessels, twistlocks, railcars and trucks). These differences make it difficult to develop standardised solutions that can be robotised.

Therefore, the interaction between man and machine at execution and control level is a key issue in a terminal’s design. The operators should understand the way in which the system works and be able to support it fully. Any interaction between manual operations and automated operations that is not strictly necessary should be avoided. Possibly, incentives might have to be implemented to accomplish this goal. If operators in an automated operation can influence operation, their interference actions should have a clearly defined functionality and the feedback of the action should be immediate and understandable for operational people.

Therefore, we define as guideline to design the manual operations and automated operations in an integrated way, consider the interactions between the manual operations and automated operations, and train the manual operators to deal with the automated processes.

2.3.6 Base the decisions within the design process on performance measurements

In order to understand the behaviour of the process that is carried out on a container terminal, adequate measurement criteria have to be developed, because only then relationships between events or actions and the output of the system can be laid. In addition, operational data have to be collected in order to determine whether the criteria have been met. The designers can analyse the processes and define the bottlenecks with actual operational information. Therefore, the performance indication instruments should not be limited to the indicators that measure the performance towards the customer. Earlier cases (e.g., ECT-DSL but also Dobner et al., 2002) have shown that it is better to have too many indicators than to have too few, mainly to create more insight when the system behaves in a different manner than expected.

Moreover, improvements should always be instituted when there is a lack of performance in accordance with the measurement criteria. When these criteria do not converge with both the terminal goals and customer goals, the criteria have not been well defined. Subsequently, the priority of improvements should be determined based on the potential performance increase of the improvement.

2.3.7 Make comprehensive and measurable objectives of design process

In order to be able to evaluate the design process afterwards (and to utilise simulation to a maximum extent), the objectives have to be clear in advance (Saanen, 1996). The performance indicators should be in line with these objectives in order to be able to determine whether the objectives are accomplished when analysing the performance indicators. Furthermore, these indicators should be measured on a continuous basis during the operation; first in order to evaluate the design when commissioned, afterwards to evaluate the potential of improvements and changes.

2.3.8 The design process should be an ongoing process in order to keep the terminal up-to-date with continuous changes

The environment of a container terminal is ever-changing, for instance the handled volume increases, the size of vessels changes, the modal split changes and the labour cost changes; in most design-engineering processes, the design team is dissolved after commissioning. However, in an ever-changing environment the design process should be continued in order to keep the terminal fulfilling it requirements. In the inductive cases, we have learnt that many terminals do no remain up-to-date, which leads to decreasing a service level or a less competitive position because internal and external factors such as labour cost, dwell time and vessel call pattern change.
The (re-)design effort might be at a less intensive level after commissioning, however, the evaluation and improvement process should be continued in order to know whether changes are required or improvements can be made.

2.3.9 The design of hard- and software should be integrated in order to design a cost-effective automated system

A robotised, automated container terminal consists of both hardware (the infrastructure and the equipment) and software (the controls at many different levels, mostly executed automated though some manual control remains necessary). Usually, these two completely different component types are integrated only after the component design has been carried out. However, the inabilities of one component type can be taken care of by the other component type and vice versa. For instance, an advanced routing algorithm can compensate for the lack of space on a terminal site. Alternatively, different curve movements of the AGVs can solve deadlock problems that occur in a certain routing algorithm.

In order to find solutions that are not only feasible from both the hard- and software side, but are also the best solutions from a technical and economical point of view, integration of both design processes is required. In a simulation environment, both component types – hardware and software – are represented by software; the hardware as finite state machines, which can be in a number of pre-defined states, and the software in an explicit way by representation of the real algorithms. Because the different component types can already be integrated at such an early stage, the possibility arises to integrate the hard- and software design. Important here, again, is to evaluate the component design from a holistic perspective and not from an isolated albeit it integrated point of view. Only then, design decisions can be taken in line with the terminal design objectives.

2.3.10 The architecture of the simulation environment should mirror the system architecture

It is common to model in accordance with the scope and purpose of the analysis for which the simulation is used. Often, this results in models that are more or less different from the system that will be implemented in reality in terms of structure and processes. That is not a problem in itself; it can even reduce costs of model development, because the representation of reality in the model was easier to realise and still valid for the purpose for which the model was developed. However, it does not contribute to the re-usability of models within a design project where the same system components are redesigned multiple times. Nor does it support the use of the same models throughout an entire design-engineering process, because there multiple purposes are inherent to the various activities in the design process. Finally, yet importantly, creating a model whose architecture is similar to the real system is beneficial during the implementation process, where it can serve as system environment for function and technical testing. Therefore, we propose to develop simulation models that have an architecture, which is similar to the real system, both hardware and software.

3 Developing PCS software using simulation

As said earlier, the PCS’s importance is much larger at an automated terminal, because many decisions are automated. Therefore, simulation should be able to support the design process of the PCS. Following guideline 2.3.10, the following requirements to the simulation environment can be defined (see also Suanen and Franzke, 2000a):

- The simulation environment should have an identical architecture as that of the PCS, so that components can be plugged in and out and exchanged between real software and simulation environment.

- Within the simulation the communication between PCS components and equipment (AGVs, RMGs) has to be modelled explicitly so that in the case of on-line interaction between
simulated components of the PCS and real ones, no problems occur regarding communication delays or asynchronous communication.

- The dynamic behaviour of the physical resources has to be modelled in a valid and detailed way and most preferably the equipment communicates via the same interfaces with the PCS as in the real system.
- The information software components need should be available at the same locations as in the real PCS, i.e. the information architecture should be identical.

Furthermore, it means that typical PCS components need to be present in the simulation environment as well:

- Planning and scheduling of transportation and transhipment jobs (timing, sequence, co-ordination).
- Job assignment to equipment (minimisation of resources).
- Real-time job co-ordination (rendezvous planning, interchange zone occupation, sequence guarding, progress monitoring).
- Routing.
- Collision and deadlock avoidance (see Figure 3).
- Job execution.
- Job monitoring (status, equipment properties, job progress, failure management).

Recently we have gained experience with software testing and development in two cases. In the first case we developed a simulation of an existing terminal in order to improve the terminal performance. Because the main performance gains were thought to be reached with software improvements and because the improvements had to be compared starting from the current system and current PCS, the simulation had to provide a valid representation of the PCS to a high detail level. In order to accomplish this we made a detailed comparison between an emulation of the PCS and the simulation model, using identical loading lists, an identical initial stack, the same equipment configuration and specification and a similar landside arrival pattern. After reach a point that the simulation performed with 5% accuracy the same as the real PCS, we started to improve the software. Because the simulation model has a modular architecture, we were able to exchange certain functionalities with alternatives, such as job assignment, sequence control, collision avoidance and job scheduling. By prototyping the improved modules, first of all a distinction could be made between should have, nice to have and should not have, so a priority list for implementation could be defined. Furthermore the quantification of performance improvements could be compared with the cost concerned with the improvement. Thirdly the prototype provided more than a functional specification only; it provided a prototype implementation – already tested as part of the entire system – that could be used as basis for the real software.

In the second project we used the simulation model as test environment of real software, especially because the entire system environment was still lacking when a number of modules were ready, for instance routing, job assignment, collision avoidance and deadlock avoidance. By providing a whole system environment first of all one can see the contribution of improvement of a single module on the whole, in order to avoid sub-optimisation or not efficient improvements. Secondly one can test the module under all kind of controlled conditions, even in a reproducible way. So when fine-tuning a module, one can test it again under the same circumstances so that an assessment can be made of the relative improvement. Finally the on-line linkage of software and simulation supports the understanding the logistic consequences of certain algorithms and choices, especially because the dynamics within the system sometimes lead to unexpected results, even for the designers themselves.

We can say that those recent experiences with software specification, prototyping and testing were successful in the sense that results were obtained in a relatively short period of time, and that decision-makers as well as designers understood the dynamics as well as the effect of improvement.
measures better. Simulating the terminal processes at such a detailed level furthermore created the possibility to assess the quality of software solutions much faster than traditionally.

4 Conclusion and future outlook

Robotisation and automation can no longer be ignored when designing a cost-efficient high-speed container terminal. The design process of robotised terminal differs from traditional design processes, because of the central role of software design and engineering. In order to avoid gaps between functional specifications and the implemented (hard- and software) system, advanced object-oriented simulation, applied during the entire design and implementation process, can support and improve the design process. As we have experienced in two recent cases, it can also improve and speed up the software design and development process, and simultaneously give the decision makers better insight in what is going on in the “soft department”.

In order to be able to support an entire design process, simulations of different aggregation levels are required, which are applied in an iterative way, in one direction providing input for other models, in the other direction providing feedback for assumptions and choices.

Figure 4: Possible visualisation of a virtual terminal using Automated Guided Vehicles and Overhead Bridge Cranes.

The virtual terminal, fully operational without any equipment running yet, is about to be available, including a distributed control system, containing all functionalities in a prototyped form. One can even start training the people in the control tower in controlling the system, and taking care of disturbances and exceptional situations. After commissioning the real terminal, the virtual terminal can play a supporting role during the operation by providing the following functionalities:

- Replay of events, in order analyse and possibly fine-tune the control system.
– Pre-plan the operation by running a number of alternative vessel and yard planning and determine which planning works out the best in the operation.
– Test environment for extensions, small improvements during the terminal’s lifetime, et cetera.

References

DOBNER, M. ET AL. (2001), Cost Performance Study APM Terminals, intermediary report, internal report Demag Mobile Cranes, Düsseldorf

DOBNER, M. ET AL. (2002), Cost Performance Study APM Terminals, final reports, internal report, Demag Mobile Cranes, Düsseldorf


RENGELINK, W.; SAANEN, Y.A. (2002), Improving the quality of controls and reducing costs for on-site adjustments with emulation: An example of emulation in baggage handling. 2002 Winter Simulation Conf.


SAANEN, Y.A. (1999), *Examining the potential for adapting simulation software to enable short-term tactical decision making for operational optimisation*, IIR Conf. on Simulation and Optimization in Container Ports, IRR, pp.300-p312, London.


SAANEN, Y.A.; DE WAAL, A. (2001), Simulatie PCS Nieuw (in Dutch), report ordered by ECT, TBA Nederland, Delft


ZEIGLER, B. (1976), *Theory of Modelling and Simulation*, John Wiley & Sons
Material Management in Shipbuilding

Ivan Siksne-Pedersen, Logimatic Software A/S, Denmark, ISP@logimatic.dk

Abstract

Shipbuilding features concurrent engineering, at best short repetitive series, frequent interaction between design and production, and short planning horizon for the detailed work. These characteristics determine the strategies to deal with the material management process and requirements for supporting information systems. These are outlined with their specific implementation in Logimatic.

1. Introduction

This paper focuses on why Material Management in shipbuilding is conceptually different from material management in other types of industry, e.g. manufacturing cars, TV, mobile phones. The description is given through a listing of the main characteristics the material management processes have in shipbuilding.

By using these characteristics a set of requirements for the conceptual important functions needed in a Successful Material Management System are described. Based on these functions reflections are made on the core functions available in the generic ERP Systems available. Finally general considerations on the use of this type of systems in shipbuilding are given.

The information in this paper is based on more than 15 years experience of implementing the MARS system, which is an own-developed Material Management and Production & Quality Control System in different shipyards worldwide.

The experience comes from the work on design, development and implementing of dedicated solutions on systems to support the material management & production control processes in shipyards. No direct reference is made to any of the yards involved and the paper should be looked at as a ‘Pragmatic description of an area of processes in a pragmatic business’.

2. Why Focus On Material Control?

The price competition in the shipbuilding industry is traditionally strong and has become even stronger during the last few years. To keep the market position and gain new contracts every shipyard is working hard to become more productive and efficient, cutting down any possible overhead and not least reducing the labour cost. In all the attempts to become more competitive there is often much to little focus on the importance of effective and accurate control of the materials required although this is probably one of the most important cost factors. Not only the potential direct savings in purchasing the “right” material at the right time and thereby reducing the financing cost is important. But also the importance of avoiding material shortage and interrupting/holding back the production is vital indeed.

As a general rule of thumb, the split of the cost for a general commercial new building is as visualised in Fig.1. However, some types of ships can have a slightly different split of the cost related to the areas indicated, Fig.1. A 10% saving in Labour Cost gives a total saving of 2% whereas a 10% saving in material costs gives a total saving of 6% of the total newbuilding cost. For navy ships the material cost is even higher and sometimes even in the proximity of up to 80% due to special equipment and related weapon systems.

The high material cost improves the competitiveness for high labour rate countries while it on the contrary makes it almost equally important for the cheap labour countries to optimise the material management routines.
3. Characteristics of Shipbuilding Business in Relation to Material Management

What makes it complex and difficult to deal with material control in shipbuilding? The primary reason is that an essential characteristic of material control in shipbuilding is high degree of uncertainty. This becomes critical in the light of the early commitment, which the shipyard has to make. Usually, the shipyard is requested to sign a contractual commitment at a time where the uncertainties in connection with the considered order are at the greatest e.g. procurement contract has to be made based on preliminary design detail.

In relation to this it is also a challenge to avoid purchase of material that ends up being surplus. The goal can be summed up: To ensure:

“The right material, at the right time, at the right place, in the right quantity”

When reflecting on the shipbuilding business in relation to material management the following specific characteristics can be highlighted:

- The project identification is the prime bearer of information.
- Shipbuilding is an interactive, concurrent time-wise process between design, procurement and production.
- Due to short project period the detailing level of the information is rather low in the beginning of the project.
- Shipbuilding companies are specialised in design, construction and production of individual plants where in principle all information except from general stock control is project/order related.
- All production is dedicated to a specific order/project
- The functions of the ship is described through a system breakdown structure
- Cost control performed up against a system breakdown structure
- No prototyping
- Requirements to traceability of material
- Deadlines for purchase and production controlled by the project plan (activities)
- The technical design is a basis for the purchase process
- The purchasing process is technically oriented complex and comprehensive
- Early purchasing of “key” items
- Project/order specific material > 80% of total material cost
- Special routines required for handling of steel material
- Outfitting process managed by palletising of materials
3.1. System Breakdown Structure and Cost Control

For every new building the functions in the ship are described through a system breakdown structure. The system breakdown is normally (and should be) functional related to support the design process, which is often done system wise. The different systems are split into smaller design tasks.

The specifications, estimates, material definition, purchase and material handling is all related to the different system and the pre-calculation, budgeting and follow up on material costs are based on systems structure.

The system structure will normally be totally different from the accounting structure used in the financial system.

3.2. Design

The design process is to a great extent performed as concurrent engineering with limited (none) prototyping. The challenge is to build the prototype and then to repeat it in a relatively limited series of sister project/ship.

The design is performed system oriented where the hull itself represents one system and the other systems covers the outfitting of the hull.

For a newbuilding project up to 80% of the material is project specific. This means that technical design is a basis for the purchase process and often the designers are in direct contact with the suppliers during the design process. For some key-items the designers performs the purchase work except for the actual commercial detail like payment term and delivery conditions, which are managed by the purchasing department.

3.3. Material Definition

For the strategic material needed for a project only a very small part can be considered as ordinary stock material and reordered based on regular consumption. Most of the material is defined specifically for each individual project although they might be standard materials. The material definition is performed as a part of the engineering process and concurrently when building up the part lists.

When the engineering process is performed as a concurrent process it is not possible to complete the material definition and material requirements before the purchase process can be initiated. By reason of this it is important to focus on the right material and quantity rather focusing on trying to bulk the material requirement which most likely will change at a later stage due to change in plan and/or design.

Material in a shipyard can be divided into two main categories: Outfitting Materials and Steel Material. The outfitting material are material like different types of equipment to be purchased. Steel material are divided into raw material to be purchase (Plates and profiles) and different steel parts, panels, pre-fabrications for building the hull.

3.4. Planning

The planning of the work is performed as a building plan defined through the creation of a work breakdown structure.

Much planning is based on experience and assumptions and requires therefore to be performed in steps starting with a rough schedule that is refined during the phase for the initial design and more elaborated during the detail design and production design phase.
In the final stage where the actual production is performed the plan is often changed caused by late deliveries or late changes in design.

3.5. Procurement

The most difficult procurement processes for shipyards are those related to project specific components. The process is difficult because:

- Need for early purchasing of “key” items
- Information required is created iterative and final information is often only available after a purchase contract is placed.
- The purchasing process is very technically driven and requires teamwork between designers and buyers.
- Purchase of material is often made before the final need-date in production is decided.
- Steel material (plates and profiles) often need to be treated are project specific material

It is extremely important to focus on the key-items for a project. The key-items are the project specific material that is defined, purchased and used for a project and are often the materials with the longest delivery time and are the most critical items for the material handling. Buying in a wrong quantity will affect the material costs.

![Fig.2: The Material Triangle](image)

Fig.2 shows a typically split between the quantities of material in relation to the values they represent. It underlines the importance of being able to uniquely identify and focus on the project specific materials as these represents up to 80% of the material costs.

3.6. Material Handling

The material flow inside a shipyard is performed according to the project plan. Two different material flows exist: One for handling the different steel parts for building the hull and one for handling the deliveries to production of the different outfitting material.
Delivery of material to hull production
The delivery of the plates and profiles is based on the planned start date for building a steel section. The materials are delivered to the shops where the planned cutting process is performed. The building structure and deadlines for each section control the following flows of material between the steel shop.

Delivery of material for outfitting
The deliveries in production consist of packages of material needed for a certain activity, ‘palletising of material’. The material flow is not controlled by the use of routing and lead-time information. The foreman requests for the delivery of the material, which means that the materials are not pushed into production according to a predefined schedule, but are pulled according to a planned and confirmed need.

The planning of the production is very much de-centralised. The detailed planning of the production activities are performed according to the status of other activities, but also to a high degree of the availability of material e.g. a change of plan because of late deliveries of material.

4. Practical IT-solution

Based on the characteristics for the ship building industry the work process can be mapped into general requirements to a conceptually correct IT-solution. In the following the key functions needed in a practical IT-solution for a successful material control process are described.

4.1. Work Process Areas

The material control process with special requirements in relation to shipbuilding are:

- System structure and Cost Management
- Design and material definition
- Part Listing
- Purchase
- Material handling
- Subcontractor handling & follow-up

The main work processes focuses on the material control process. As a result of the identification of the characteristics of shipbuilding the production planning is performed as a work break down structure as an activity plan.

Fig.3: Material Control and related Strategic Important Systems

The important issue is that the timing of e.g. deliveries of material, completion of certain parts of systems can be defined without the plan is made in detail. It is therefore important to be able to relate the activities (milestones and jobs) to the material requirements and when a change of plan is introduced
the timing of the delivery deadlines for materials are changed accordingly. The planning process is therefore not needed to be an integrated part of the material control system but as a vital tool like the CAD system and financial system, Fig.3.

4.2. System structure and cost management

In the project set-up process the definition of the System Structure of the new building is performed. The system structure serves two purposes, 1) the designers’ tool for planning and dividing the work between the different design groups, 2) basis for Cost Management.

For serving these main purposes the system accounts must be an integrated part of the material control system. In relation to design/engineering areas such as material definition and part listing the relation to the different systems must be established, and in relation to the follow-up on the committed and final costs.

4.3. Design and material definition

The design process is performed as a concurrent process. Next to that the purchase of certain material and production of certain systems must be initiated before the design is completed. This sets up some requirements to the tools to use.

Material Definition
Traditionally, only one material catalogue is available. When only one catalogue is available it contains all the different material used in design, purchases, production etc. Material that can be characterised, as project specifics are often used only once but stays in the catalogue forever.

A use of only one material catalogue does not support the necessary flexibility in design process where a material can start up as being different items e.g. 2 different pumps, but after changes in design a different standard pump can be used instead.

Another important issue is that often the specification of material develops during the negotiation with the supplier and the material can therefore not be considered a standard material for other designers to use. Material like this needs to be separated from the standard materials.

What designers need is a project specific catalogue for defining project components. The project Component catalogue makes it possible to define project specific material. A project component is identified by: Project Id. + System Number + Free Number. This identification makes it possible to achieve full tracing of each component from the early definition through procurement, receiving, issuing of material. Further it is automatically organised systemwise.

During the design process of a system the materials needed are defined concurrently. The materials are defined as project specific materials, Project components e.g. the designer needs a certain amount of valves in a system. The valves are then defined as project Components and defined for a specific system. For some of these project components, e.g. main engine, an early purchase is needed. For other project components the designer concludes that the valves can be purchase as a standard material. The designer then relates a standard material number to the project component.

Fig.4: Project Component and Standard Material Catalogues
As Fig.4 illustrates, the project components are placed in a separate catalogue. It is possible to relate a standard material to the project component. For timing of the requirements of these project components an activity can be related to each component describing the deadline for use for purchase or production.

**Early Purchase needed**

For some project components it is necessary to start up the purchase process before the design of the system is completed. Various reasons for this exits e.g. design is dependent on the suppliers ability to supply, long delivery time. At this stage these components are not yet finally placed in the final outfitting part list but the activity related to the components defines the timing of the material requirement.

The function to support this early start up of the purchase process is the Technical Purchase Order. In the technical purchase orders all technical details on the components to buy are defined. When technical purchase process is completed the order is transferred to the mercantile handler for completing the purchase order by adding various commercial information.

**4.4. Part Listing**

The part listing process in shipbuilding is split in two different areas: Part list for building of hull, and part list for outfitting. Both part list types are used for supporting the material control process in relation to purchase requirements and delivery of material to production.

**Part list for Hull**

The part lists created for the hull describe a hierarchical break down of the entire steel structure into elements such as blocks, section, sub-assemblies and panels. This parts list supports the building of the hull.

Traditionally material like steel plates and profiles has been considered stock material and purchased in bulk. For supporting the purchase process for buying steel and decrease the surplus amount, it is necessary to have a separate steel estimating function for defining and reservation of steel requirements pr. section/block. This estimation is then used when buying steel and when planning the delivery of plates and profiles to the different shops.

**Part list for Outfitting**

The part listing process consists of building up the material needed for the different design units. A design unit can be an assembly, a drawing, a pre-fabrication or similar. By relating the material from the catalogues to the parts list the material list for the design unit are defined.

It is possible to create a hierarchy of different design unit by relating these to each other. Example: A part list describes the material needed to fabricate a pipe spool. The part list the pipe spool is related to is the part list describing the material to install the pipeline the pipe spool is a part of.

The part lists and their relations, together with the activity, describes the material needed and the deadlines.

What is important is that by building the part list up in different design units the design and planning can support the production in a parallel process.

The part lists for outfitting have two main purposes:

- Defining the material requirements.
- Support of the outfitting process
For material and components for which early purchase is not needed the part lists defines the material requirements, which is presented in the material status.

To every part list line an activity is related and the timing of the material requirement is defined. The start date of the activity defines the date of need in production. By relating the activity to the different lines it is specified in which activity the material is to be used and a relation between the system breakdown structure and the work breakdown structure is established.

4.5. Procurement

The procurement process is dependent on the category of material to purchase. For project components and standard project material the purchase process is initiated by the designers creating the technical purchase orders. The buyer can then decide to perform an enquiry process.

For other category of material the purchase process is initiated by dealing with requisitions from designers and by checking the material status.

The overall philosophy is that this process is partly manual performed. The challenge is not how fancy the tool deals with reordering of standard catalogued items but to focus on the critical items. The purchase process is seen as an on-going process that collects the information from all different parts of the material control system, Fig.5.

![Purchase Process Diagram](image)

**Fig.5: Purchase Process**

4.6. Material Handling

In shipbuilding, the production plan is never that detailed that routing information can support a delivery of material to a specified workshop. Since the production planning is highly decentralised and performed by the foremen in production the materials are not delivered to production without any ‘Request for material’ made.

Fig.6 shows the entire outfitting process.
Before an activity/job is performed the material needed for this job can be identified by the use of the pallet list. The pallet list is the ‘work package’ of material for an activity.

The foreman in production request for the material to be delivered. Prior to the actual request for material the foreman must check the availability of the material in order to make changes to the outfitting plan. To avoid stops in production the goal is to install/outfit using the material available.

When the foreman in production approves the pallet list is becomes a ‘Delivery Request List’ and initiates the stock handling for creating the picking list and the final issuing of material to production.

When planning the outfitting work often the result is that a job involves material from different design unit (drawings, systems etc.). It is therefore needed to be able to relate different activities to lines in the same part lists. This function is shown in Fig.7.

4.7. Subcontractor handling & follow-up

In shipbuilding the general tendency goes more and more in the direction of utilisation of subcontractors at the shipyard site itself or at the subcontractor site, and this put some requirement for
possibility of handling of sub-contractors as well as the possibility of monitoring the progress of the subcontracted work.

The key to follow-up of this work is logically the purchase order itself, as this is where the work is procured, and therefore there must be some functionality built-in, where the progress of the delivery of the work can be reported into the system as e.g. when a delivery registration of some goods are made, and the stock department can see if the delivery has been made partly or totally.

4.8. ERP within ship building

Based on the facts that the shipbuilding industry demands different type of IT-solutions than the ordinary manufacturing industry many shipyards did earlier develop their own IT-systems. Today this is against the general IT-strategy as most companies try to replace ‘home-made’ solutions with ready-made systems. This is one of the reasons why general-purpose ERP solutions have experienced explosive business development, especially within the ordinary manufacturing industry.

For shipyards it might also be easy and in some way attractive to follow the main stream and implement one of these ERP concepts. Some shipyards have even decided to go that route and have started up the practical implementation process.

The implementation of these systems has often failed and the result has been a burden to the whole organisation. Further the later operation of the system is troublesome even though considerable resources and costs have been spend on adapting the system to the company’s working procedure.

The main shortcomings of ERP in relation to shipbuilding can be summarised as follows:

- Most ERP systems are based on MRPII theory which is developed for the general manufacturing industry (mass productions)
- ERP systems are fundamentally not designed for Project Driven Industry (One of a kind production) and their data structure will never become suitable for shipbuilding. Consequently the shipbuilding industry will never become their core business.
- The data structure of traditional ERP systems presupposes input of comparatively much information before valuable output can be extracted. This is in contradiction to the work process in a shipyard that is precisely characterised by an iterative process between design, procurement and production.
- The material control functions in ERP systems are normally based on a “predefined” and stable Bill Of Material (BOM). This provides the user with information for lead-time for purchasing well defined detailed material requirements and lead-time based on detailed and correct information on each assembly stage in a well-defined line for production. All this information requires the required final product to be totally defined up front both from a material view and a construction view when the order for the product is signed.
- There is a tendency that ERP systems put a heavy attention on work order generation, stores handling, stock control and production of standard items. These functions often create a lot of transactions that from a strategic point are less important for a shipbuilding enterprise.

Example of or ‘Best in Class’ System

During the last 15 years the Danish Company Logimatic (LMC) have continuously developed and implemented the MARS (Material- and Production Management System) in shipyards world-wide.

The MARS system was originally a spin-off product based on a customer specific system that was developed in close co-operation between several Danish shipyards and the Finnish shipbuilding group
Kvaerner-Masa Yards. On basis of this specific system it was decided to develop a standard system available for shipyards however designed and developed for this specific market segment.

Today the MARS system has achieved a reputation and position as the best available system for Material Management & Production Control for shipbuilding & shiprepair world-wide, with basically no “well proven” competitors. The reason of current position of the system relates not only to the system functionality available, but also to the general knowledge of the business processes in shipyards in general.

4.9. Going for ERP or ‘Best in Class’

Today the demands from the market are changing towards more modularised concepts where each core process is supported by a “Best in Class / Best of Breed” solution, partly due to the fact that the competitive margins are getting more and more narrow, but also the fact that there are a request for general improvement of efficiency.

In contrary to the integrated ERP solution the objective of the “Best in Class” concept is to provide the optimum IT-solution for each core business process. For the shipbuilding industry the main business processes are:

- CAD / CAM
- Material Management & Production Control
- General Project planning
- Financial & Human Resource

These systems should be based on open database technology. Each of them providing the necessary standard interfaces, which can be adapted to fulfil the need of each particular customer. If possible they should operate on the same relational database which makes it easier to extract the necessary management information into an Executive Information System.

Some of the main strengths of the “Best in Class / Best of Breed” concept are:

- It provides an optimum solution for each core business in the shipyard
- No compromises in functionality in order to achieve “one” integrated product
- Flexibility to upgrade each application as appropriate and required
- Less depended on one supplier
- It complies with the requirements to open and flexible systems

After all it is totally vital that the shipyard makes the optimum and correct decision on systems to support the core businesses which are the processes directly related to building of the ship. The optimum Material Management & Production Control System is therefore probably the most important subsystem within the shipyard IT-environment and no compromises are allowed when choosing that software. Only the ultimate best functional solution is acceptable.
Methods of Artificial Intelligence in Navigational Safety Assessment of Ship Encounters

Zbigniew Pietrzykowski, Maritime University of Szczecin, Szczecin/Poland, zbij@wsz.szczecin.pl
Janusz Uriasz, Maritime University of Szczecin, Szczecin/Poland, lubat@wsz.szczecin.pl

Abstract

The assessment of navigational safety of ships’ encounter situation is an essential element of on-board decision processes. This assessment forms a basis for planning adequate manoeuvres and the right moment to execute them. It is also indispensable in decision support systems. The paper presents methods and tools of the representation of navigators’ knowledge for the assessment of navigational safety of passing and overtaking manoeuvres. The results of knowledge acquisition as well as its possible applications in a decision support system are discussed.

1. Introduction

The basic objective of marine navigation, executed in the decision process of steering a ship, is to safely conduct a vessel from the place of departure to its destination. In practice, in this process the navigator recognises and considers encounters with other vessels.

Navigation in a restricted area is subject to various hazards due to many limitations, such as geometrical parameters of the area, technical parameters and manoeuvring characteristics of a ship, characteristics of a position determination system, and hydrological and meteorological conditions. In addition, manoeuvring enhances certain negative phenomena, e.g. increased squat or decreased manoeuvring capabilities. Navigation in a restricted area inevitably brings about encounter situations. These necessitate certain manoeuvres to be made: overtaking, passing or stopping.

A complex character of the phenomena occurring during these manoeuvres makes their analytical description very difficult. And so is an assessment of a navigational situation. The application of methods and tools of artificial intelligence allows to utilise navigators’ knowledge in the process of navigational situation assessment.

2. Characteristics of ship encounter situations in a restricted area

The manoeuvres of passing and overtaking are the most frequent manoeuvres performed by ships in a restricted area. The passing of a ship has to be planned well in advance due to its relatively short time. The two ships involved are often required to co-operate with each other (Fig.1a). Certain adverse phenomena occur during such manoeuvres, that is pushing and attracting forces in particular stages of a manoeuvre. The navigator has to be aware of these forces and counteract appropriately. Association of the knowledge indispensable for a passing manoeuvre to be safely performed can make it possible to analyse and attempt at automatic evaluation of such a manoeuvre.

When an overtaking manoeuvre takes place in a restricted area, the navigator has to be aware of associated phenomena and situations. In most cases overtaking can be performed only with a consent and co-operation of the vessel to be overtaken. (Fig.1b). The following facts affecting the safety of manoeuvring are to be taken into account: substantial duration time of the manoeuvre and asymmetric water flow along the vessel’s sides making it more difficult to steer.

The full awareness and understanding of existing hazards and hydrological phenomena allows to safely execute an overtaking manoeuvre. Therefore, the knowledge of these hazards and phenomena is a prerequisite for safe manoeuvring. So, an attempt of its association is purposeful and desired.
Fig. 1: Encounter situations in a restricted area: a) a passing manoeuvre; b) an overtaking manoeuvre

3. Research methods

The problem of vessel traffic in a restricted area, particularly in view of navigational safety, is investigated by marine traffic engineering. This science uses various research methods. The navigational situation assessment is performed through simulation, measurement, expert, statistical and artificial intelligence methods. These methods make it possible to solve an examined problem in a comprehensive manner and allow to carry out current identification of navigational safety assessment. The methods are briefly described below:
- computer simulation – enables simulation of possible navigational situations in a relatively short time,
- measurement – enables the acquisition of parameters (in the simulation time) affecting navigational situation assessment,
- expert research – makes it possible to acquire the knowledge (facts) of navigator-experts indispensable in the correct process of navigational process assessment,
- statistical – allows to analyse expert assessments in view of their usefulness in the process of navigational situation assessment,
- artificial intelligence methods – have tools enabling the use of expert knowledge in the problems of navigational situation assessment.

The methods mentioned have been applied, inter alia, for the assessment of navigational situations occurring during a ship’s passage in a restricted area, Pietrzykowski (1997), Dziedzie et al. (2000), Pietrzykowski and Uriasz (2001). The assessment of passing and overtaking manoeuvres is more difficult as it additionally requires taking into account dynamically changing parameters of the other ship’s state vector and mutual interaction of the two ships.
3.1. Simulation research

The simulation method used the ship handling simulator NMS-90 of the Maritime University of Szczecin. The following area and vessel models were used in the experiment:
- A straight section of the fairway 100 m wide,
- A "Freigh" bulk carrier model with length 95.5m, beam 18.2m, draft 5.5m.

The following conditions were simulated:
- the ship was handled with the helm (neither thruster nor anchor employed)
- ship’s speed 4.1 [m/s] (8 knots)
- very good hydro-meteorological conditions (good visibility, no current, no wind, no wave action)

The simulation research consisted of a number of passages in which both passing and overtaking of ships were executed. All vessel movement parameters were automatically recorded during the simulation (Figs.2 and 3):

\[ \Delta y \] – distance from the fairway centre line,
\[ \Delta K D d \] - deviation from a pre-set course, determined by the fairway centre line
\[ \omega \] - ship’s rate of turn (Fig.1).
\[ d_{\text{min}} \] - distance between two nearest points of the examined ships.
\[ KK' \] - heading angle to the other vessel – bearing on an obstruction (other ship)

![Fig.2: A ship passing manoeuvre on the fairway](image)

![Fig.3: A ship overtaking manoeuvre in the fairway](image)

Dangerous situations were simulated during the experimental passages, such as sudden turns, beaching, collisions with a vessel. Besides, very safe situations were simulated, the so called model passages.
3.2. Expert research

Expert research supplements simulation research. The former consists in recording navigational situation assessments performed by navigators during simulated passages.

Participants were navigators with diversified qualifications and experience: deck officers and master mariners. The experts evaluated navigational situations according to their own individual criteria at 15 second intervals using a 0 to 10 scale:
- 0 - a very safe situation,
- 10 – a very dangerous situation (collision or imminent collision situation).

The individual assessments of experts made up a basis for determining mean values of individual assessments of recorded situations that were standardised to the <0, 1> range. In this way the gathered facts equivocally described the recorded situations and their assessments, based on navigators’ knowledge and experience.

3.3. Knowledge representation

The research has resulted in collecting data /facts/ referring to the process of navigational situation assessment by navigators during passing and overtaking manoeuvres. This is one of the forms of knowledge representation. As it is practically impossible to gather facts concerning all possible situations, it became necessary to attempt at generalization of that knowledge. An approximating tool in the form of artificial neural network with fuzzy logic was used for the purpose, Osowski (1996), Rutkowska et al. (1997). The network structure was assumed to have the following three layers:

- layer 1 (input): 5 neurons; the number of neurons depending on the size (dimension) of the input data vector x,
- layer 2: 243 inference rules
- layer 3 (output): 1 neuron; the number of neurons depending on the size (dimension) of output data vector (one-dimensional vector – navigational situation assessment).

The collected data were used in the process of learning of neural network with fuzzy logic, separately for each of the manoeuvres. The supervised learning was applied – the method of backward propagation with a momentum.

4. Encounter situation assessment

After the learning process had been completed, the correctness of network operation was verified. This enabled the evaluation of navigational situations occurring during analysed manoeuvres for the full range of changing input parameters: distances from the fairway centre line (\(\Delta y\)), deviation from the pre-set course determined by the centre line of the fairway (\(\Delta KDD\)), ship’s rate of turn (\(\omega\)), distance between two nearest points of the examined ships (\(d_{\text{min}}\)) and the heading angle to the other vessel – bearing on an obstruction (other ship) (\(KK^+\))

4.1. Passing manoeuvre

Fig.4 shows an example of a passing ship’s movement trajectories and standardised expert assessments ranging <0,1>. The obtained responses of the network are correlated with the expert assessments at the level of 0.992. Fig.5 shows the network responses (navigational situation assessments). Due to difficulties in presenting how five variables depend on the assessment of a navigational situation, the diagram shows the relation between two variables with the others being constant. The diagram shows an assessment of a navigational situation of a passing manoeuvre for a case in which the ship maintains the course KDD=095°, does not make a turn while the ship being passed bears –10°).
Fig. 4: Trajectories of ships making a passing manoeuvre in the fairway in passage no 1 and experts’ mean assessments

Fig. 5: Assessment of a navigational situation during a ship passing manoeuvre for KDd=095°, \( \omega=0°/\text{min} \), KK’= -10°

Positive distances between ships (the closest points) denote a situation when the two ships approach each other (approaching phase), while negative distances denote the fact that the ships, after passing, move away from each other. The line “d” expresses potential positions of the two ships’ closest points of approach for the assumed parameters of assessment. The navigational situation assessment presents areas of adequate safety levels within which the ship can be situated. These areas make up specific swept paths. They are formed so that maintaining a ship within such a path will result in the ships passing each other at an adequate safety level. The diagram analysis leads to a conclusion that the passing manoeuvre should be started at the range of 4L-5L distance between ships (ship moves away from the centre line to the starboard side of the fairway), whereas the last moment to start the manoeuvre is when the distance between the ships equals 3.5L (one should bear in mind that during the
passing the ship being passed does not make any manoeuvres). The most dangerous area stretches ahead of the ship being passed at a distance of 1L to 3.5L (L=95.5 m). The safest area is found in the vicinity of the fairway centre line astern of the ship being passed ( navigational situation in which a hazard has been avoided).

The applied research methods and the designed tool, a neural network with fuzzy logic, make it possible to assess a navigational situation at various phases of the passing manoeuvre. The assessment can be performed on a continuous basis.

4.2. Overtaking manoeuvre

Fig.6 shows an example of a ship movement trajectory for an overtaking manoeuvre and standardised expert assessments ranging <0,1>.

![Ship movement trajectory during an overtaking manoeuvre on the fairway in passage no 11 and experts’ mean assessments.](image1)

![Assessment of a navigational situation during an overtaking manoeuvre in the real fairway for KDd=270º, ω=0º/min, KK*=45º.](image2)
Fig. 7 shows an example of the network response. Similarly to the case of passing manoeuvre, the influence of two most important parameters is presented, i.e. the deviation from the fairway centre line (assessment-deviation correlation coefficient equals 0.42) and the distance between ships (assessment-distance correlation coefficient equals −0.82) with other parameters being constant. The diagram of the network response presents the influence of the deviation from the centre line and of the distance between ships on the assessment of a navigational situation at constant values of the remaining parameters: KDd=270° (parallel to the fairway centre line), \( \omega = 0 \), KK′=45°. Positive values of distances between ships mean that the ship being overtaken is located ahead of the overtaking ship, while negative values mean that the ship being overtaken is sailing astern of the overtaking ship.

The assessment of navigational safety presents a ‘virtual’ path within which the overtaking ship should be proceeding. The most dangerous area (narrowed safe area) refers to a situation in which the distance between the ships is approx. 0.5 L. The overtaking ship, while carrying out an intended manoeuvre, has to be handled properly sailing between the left-hand side fairway limit and the port side of the ship being overtaken (keep the ship within a path determined by experts’ assessments). The degree of safety of an overtaking manoeuvre will depend on the ship’s position in the fairway (distance from the fairway centre line), and on the positions of the ships involved in relation to each other.

An overtaking manoeuvre in a restricted area is assessed as dangerous. The situations presented in the diagram referred to a case in which the ship does not make a turn (rate of turn = 0). In this case the lowest assessment of ship positions (alongside) produced by the network was 0.5-0.6 (standardised assessment range 0 to 1). In overtaking the relative speed between the ships is small. This makes the manoeuvre last some considerable time and calls for much caution, continuous assessment of a navigational situation and maintaining the ship within a strictly defined path. The path narrows substantially as the overtaking ship closes up on the ship being overtaken. Such a situation is shown in Fig. 8, where the heading angle is 85°. The line “d” marks potential locations of points of the closest approach between the two vessels.

![Diagram](image)

Fig. 8: Assessment of a navigational situation of an overtaking manoeuvre in a traffic lane for KDd=270°, \( \omega = 0°/\text{min} \), KK′=85°.

Analogously to the passing manoeuvre, the applied research methods and the designed tools, namely a neural network with fuzzy logic, allow to assess a navigational situation at various stages of a given overtaking manoeuvre. The assessment can be made on a continuous basis.
5. Conclusion

The research has proved the methods presented herein to be useful in the assessment of ship encounter situations in a restricted area. The application of artificial intelligence methods and tools made it possible to acquire and use navigators’ knowledge for assessment of a navigational situation.

The execution of passing and overtaking manoeuvres requires the navigator to be cautious and calls for proper classification and assessment of a navigational situation as well as earlier preparation.

During a passing manoeuvre a hazard may appear when a navigational situation changes suddenly due to a large value of the ships’ relative speed. A hazard in an overtaking manoeuvre appears due to lasting adverse effects resulting from long time of the manoeuvre. Therefore, preparation for a manoeuvre well in advance is of much importance. A properly planned, prepared and started manoeuvre in an encounter situation of two ships brings fruit – navigational safety is maintained at an accepted satisfactory level.

It is possible to carry out continuous assessments of either of the examined manoeuvres.

The assessment of a navigational situation is an essential element of decision-making process. It forms a basis for working out appropriate manoeuvres and defining the right moment to start them. Such a basis is necessary in decision support systems. The presented methods and tools can find applications in such systems.

References


PIETRZYKOWSKI, Z.; URIASZ, J. (2001), Analysis of how professional experience of navigators affects the assessment of a navigational situation, 1st Int. Symp. Human’s Factor on Board, Bremen

Artificial Neural Network and Genetic Algorithms in Identification Problems and Ship's Behaviour Forecast in Real Time Intelligence Systems

Yury Nechaev, Alexander Degtyarev, Ilia Kirukhin
St. Petersburg University, deg@fn.csa.ru

Abstract

The identification problem of a complex dynamic object state in extreme situation is considered. The analysis is carried out on the basis of combination of artificial neural network and genetic algorithms. Various types of neural networks are considered. Fuzzy neurons are used for improvement of inference. Compression of measuring information about object dynamics is achieved using cognitive structures. The procedure of identification is realized with the help of logic conclusion in decision support system on unsinkability monitoring of ships and ocean vehicles.

1. Introduction

The control of floating dynamic objects (DO) in conditions of fast interaction with their environment is related to the solution of some problems concerning data analysis and interpretation with the help of adaptive algorithms. Realization of such algorithms on the basis of fuzzy information is complicated for damaged ship and requires mathematical modelling methods. Using such modelling in the design of new generation intelligent systems (IS) allows increase there functional efficiency, reliability and survivability. Algorithms and software reflect increasing sophistication of IS. They are now one of the main conceptual elements for control and decision acceptance.

Real time IS operation is provided in conditions of continuous variation of damaged DO and environment parameters. Uncertainty and incompleteness of initial information result in additional complexity in solution of computational problems, Nechaev (2002a). Realization of effective algorithms for data analysis and interpretation in complex on-board IS is an actual problem of using of new ideas and methods requiring large computer resources in conditions of strong time limits. The methodical base for design of such algorithmic procedures of decision acceptance in fuzzy conditions is the concept of optimal solution and available resources (fuzzy goal and limits).

We consider here one of the complicated problem of assessment of DO behaviour in waves. This is identification of extreme situation in conditions of ship damaging and compartment flooding. The novelty of the obtained solutions consists in the following:

- The algorithm of information transformation during damaged DO operation in conditions of continuous variation of object dynamics and environment is carried out.
- Ways to improve the identification problem solution are proposed. These are based on new principles of information analysis and interpretation combining traditional methods with technologies of artificial neuron networks (ANN) and genetic algorithms (GA).

2. Problem to solve

Let us consider the mathematical model of a damaged ship motion as non-linear matrix equation. The general view of such equation is in Cauchy form

$$x' = \Phi(x, U, W, t)$$  \hspace{1cm} (1)

$x, U, W$ are the matrix-columns (vectors) of state variables, governed and excited actions; $\Phi$ is a non-linear vector function (matrix-column of scalar non-linear functions)
\[
X = \begin{bmatrix}
  x_1 \\
  \vdots \\
  x_n \\
  u_1 \\
  \vdots \\
  u_r \\
  w_1 \\
  \vdots \\
  w_m \\
\end{bmatrix},
U = \begin{bmatrix}
  u_1 \\
  \vdots \\
  u_r \\
\end{bmatrix},
W = \begin{bmatrix}
  w_1 \\
  \vdots \\
  w_m \\
\end{bmatrix},
\Phi = \begin{bmatrix}
  \phi_1 \\
  \vdots \\
  \phi_n \\
\end{bmatrix}.
\]  

(2)

Let us consider projections of angular and linear velocities, yawing, rolling, pitching and linear coordinates of central of gravity as state variables. State vector characterizes whole space motion of the ship, and it is consist of the following components:

\[
X = (x_p, y_p, z_p, V_x, V_y, \theta, \psi, \varphi, \omega_x, \omega_y, \omega_z).
\]

(3)

The mathematical model concretisation for damaged ship – environment interaction involves going from the general model (1) to the non-linear differential equations system. Such system is based on the hydrodynamic concept of “elongated (slender) body” and two-parameter series expansion on powers of small parameters, Nechaev (2001).


Behaviour of damaged ship in waves could be presented in the context of general theory of systems as Cartesian product of two spaces (initial and result), Nechaev (2002b):

\[
S \subset U \times V,
\]

(4)

\[
U \text{ is the set of initial values } \{U_i\};
\]

\[
V \text{ is the set of results } \{V_i\}.
\]

Considered system S in conditions of uncertainty and incompleteness of initial information is described by fuzzy equation in relations, Zadeh (1994):

\[
B = A \cdot R;
\]

\[
A = \{\alpha, \mu(\alpha)\},
\]

\[
B = \{\beta, \mu(\beta)\}.
\]

A and B are fuzzy sets of input and output; \(\alpha, \beta\) are fuzzy sets of linguistic variables; \(\mu(\alpha), \mu(\beta)\) are correspondent membership functions; \(R\) is fuzzy relation on \(U \times V\) as governing rule with the structure “If … Then … Else”, Nechaev (2001):

\[
X \rightarrow Y(Z),
\]

(6)

In terms of membership function we have the following view, Nechaev (2002b), Zadeh (1994):

\[
\mu(\beta) = \max_\alpha \min \{\mu(\alpha), \mu(\alpha, \beta)\}.
\]

(7)

The experience of analysis and forecast models development shows that we can construct satisfactory model on the basis of ANN even in conditions of data deficiency. Such model could be defined more accurately as new data becomes available. It is very important especially at initial stage of extreme situation development when we have not enough initial data for solution with the help of traditional analysis and forecast methods.


One of the complicated problems in IS is extreme situation identification. This problem is related with assessment of flooding case of compartments when DO receives the holes. We shall consider five typical cases of flooding ((1) positive GM, zero static heel; (2) positive GM, asymmetric flooding; (3)
negative GM, symmetric flooding; (4),(5) variants of negative GM, symmetric flooding), e.g. Nechaev (2001). Results of identification permit to control variation of flooded waterline and forecast of critical time interval of DO in extreme situation, Nechaev (2001), Nechaev et al. (2002).

Solution of identification problem related to class of not well-posed problems is sufficiently complicated by conditions of emergency situation development. Traditional identification procedures result in unstable solutions if dynamics of the object and environment are continuously varied. Obtaining of relatively stable solution could be achieved with the help of regularization methods. However the main contradiction of identification problem in conditions of complex dynamics of damaged DO could be solved with the help of analysis of alternative solutions only. These solutions obtain on the basis of various methods of measuring information processing.

The rational approach to solution of this problem uses the competition principle, Nechaev (2002a,b). This is a development of a “soft computing” concept given as described in Nechaev and Sick (1998), Zadeh (1994). In the framework of this principle, we simultaneously consider different computational technologies:

- standard algorithms based on classical mathematical methods;
- artificial neural networks.

The aim is now to obtain those results reliably by the shortest way. The competition principle applied here to the problem of extreme situation identification shows its efficiency in on-board IS, Nechaev et al. (2002b).

4. Cognitive structures

Solution of the recognition problem with the help of cognitive paradigm, Nechaev et al. (2002a), was carried out by way of information transformation in frameworks of formal logical description. Such approach permits to obtain appearance and development of emergence ship motion in waves in depends on character of flooding and excitation level. Cognitive spirals were chosen as model of ship motion mapping. Such spiral permits to “compress” initial information about non-linear motion with random period. Contraction of cognitive spiral on stochastic function results in periods “adjunction” and there transformation to uniform assigned value by affine compression or expansion of considered function intervals.

Cognitive spiral as alternative to phase portrait carries more information that one can choose visually orienting on image structure. Such information is breadth of stripe in spiral, colour saturation, frequency of changing of different colour stripes. Asymmetry of upper and lower parts of cognitive spiral and colour distribution at angles near 0° and 180° could play key role.

![Fig.1: Cognitive structures for flooding cases 2 and 3](image)

Considered problem of recognition is related with classification of “difficult-to-separate” typical flooding cases among five classical situations. Second, third and fourth cases fall into these categories. As mentioned the first case is relatively trivial (symmetrical flooding, no static heel, positive GM). It
permits clear separate first case without complicated inference procedures using. As for fifth case, it is usually considered as subset of fourth case with similar character of motion. Examples of cognitive structures for most character flooding cases (2 and 3) are shown in Fig.1.

Cognitive structures analysis permits to obtain necessary information for ANN learning and further classification problem solution. Correlation functions of some cuts of cognitive structures constructing on quasi-stationary intervals are used for these purposes, Bogdanov et al. (2001), Nechaev et al. (2002a). In result constant number of input neurons is determined, ANN structure is found, and the key moment – we have freedom in choosing of realization length (this length is determined from conditions of IS operation reliability in extreme situation assessment). Moving average was removed from using realizations, and calculation of correlation function was carried out for “treated” data. It increased accuracy of recognition and reduced time of ANN learning.

5. ANN topology and GA

ANN consecutively links $K$ layers of formal neurons in the form of a cone, Fig.2. Each layer has $n_i$ ($i=1,\ldots,k$) neurons. The number of neurons at each layer uniformly decreases with layer number increasing. Each neuron is characterized by consecutive combination of two components. The first is a linear transformer (adder) of multivariate input vector $r_j$ in one-dimensional vector $S$ with weight coefficients $W_{ij}$ ($i=1,\ldots,k, j=1,\ldots,n$). The second component is non-linear transformation of adder output $S$ in output signal

$$q = f \left( \sum_{j=1}^{n} w_j r_j \right) = f(S).$$

(8)

The structure of the multi-layer ANN provides a complicated non-linear transformation of input vector $r = \{r_i\}$. In symbol form such net could be presented as

$$\text{NET}_{0,n}^{1n} (i=1,\ldots,k-1),$$

(9)

$K$ is the number of layers; $n_0$ is the number of inputs; $n_i$ is the number of neurons in $i^{th}$ hidden (intermediate) layer; $n_k$ is the number of neurons in $k^{th}$ hidden layer (simultaneously it is the number of outputs $q_1,\ldots,q_k$ of multi-layer ANN); links between neurons in layer are absent, outputs of neurons of $i^{th}$ layer goes to neuron entrances of next $i+1$ layer only.

![Fig.2: Neural network in a task of identification of an extreme situation](image)

(A – input, B,C – hidden, D – output layer)

Relation between input and out of ANN is determined by non-linear recurrent equation

$$Y = q^{(k)} = f^{(k)} \times$$

$$\left( W_0^{(k)} + W_1^{(k)} f^{(k-1)} \left( W_0^{(k-1)} + W_1^{(k-1)} f^{(k-2)} \left( \ldots W_0^{(1)} + W_1^{(1)} f^{(0)} \left( \ldots W_0^{(2)} + W_1^{(2)} f^{(1)} \left( W_1^{(0)} + q^{(0)} \right) \right) \right) \right) \right.$$  

$$= F(r)$$

(10)
$F(r)$ is a non-linear function. The sigmoid function is chosen as activation function for the hidden layers,

$$y(x) = \frac{1}{1 + e^{-x}}, \quad (11)$$

the output layer uses modified linear function

$$y(x) = \begin{cases} 
  0, & x < 0 \\
  x, & 0 \leq x \leq 1; \\
  1, & x > 1 
\end{cases} \quad (12)$$

ANN learning was carried out on the basis of genetic algorithms (GA), Goldberg (1989), Davis (1991), Nechaev and Siek (1998). GA has some advantages in real time systems in comparison with standard learning algorithms based on back propagation procedure, Nechaev (2002a), Wasserman, F. (1992). Gray code was used as initial code in chromosome structure, Goldberg (1989), Nechaev et al. (2002a,b).

6. Identification and forecast

Initial data for estimation of emergence situation and forecast of such situation development are presented by realizations of non-linear rolling for five classical flooding cases (see section 3). Realizations are obtained by the way of simulation. Each of them includes 3000 periods of non-linear oscillations. Analysis is carried out on the basis of non-linear differential equations describing rolling, pitching and heaving of damaged ship in random waves of 6 and 8 state. Each fragment of realisation is used as class representative. Image (cognitive spiral) is cut on 20 non-overlapping parts. Each part is approximately 150 periods. The first 75 periods are used as learning sample for this class. The residual 75 periods are used for testing.

Standard algorithm for forecast of appearance and danger of extreme situation on the basis of mathematical modelling data is realized in the following sequence:

- Random wave field of given intensity is generated for various values of spectral density (real interval of wave characteristics variation);
- Dynamics of emergence ship – environment interaction is modelled for given random waves. This procedure is repeated for different wave fields realizations.
- Statistical processing of obtaining sets of experimental data is carried out. Values of equilibrium position parameters of considered kinds of damaged ship motion on waves determining current position of emergency waterline are calculated.
- The forecast model of extreme situation development is constructed. This model takes into account results of statistical processing of modelling results. On the basis of this model the time of damaged ship operating until beginning of critical conditions (in accordance with buoyancy and stability which determine dangerous waterline position) is determined.

Practical using of results of the work of constructed algorithm is carried out in modification of logical models of decisions acceptance of IS dynamic knowledge base. In process reliability of obtained practical recommendations has great influence on IS efficiency.

Standard algorithm of forecast of damaged ship characteristics on waves (equilibrium angles of heel and trim) in conditions of extreme situation development uses linear adaptive model in basis of variable size:

$$y(x) = \sum_{k=0}^{M} \beta_k P_k(x) \quad (13)$$

$P_k(x)$ are basis functions (polynomials), $\beta_k$ are coefficients (including random) to be determined. Fig.3 shows a graphical interpretation of the considered algorithms.
Fig. 3: Real time system modelling damaged ship characteristics on the basis of adaptive component: A – observed data (solid line) and forecasting values of output characteristic (dash line); B – general structure of adaptive model; \( x=\Delta t, 2\Delta t, \ldots, N\Delta t \) is running time of situation development; \( \tau \) is quasi-stationary interval; \( y(\Delta t), \ldots, y(N\Delta t) \) are output characteristics.

The described algorithm was compared with results of situation modelling with the help of neural network model in competition principle realization. The results of analysis are shown in the table as applied to non-linear interaction between damaged ship and environment (pitching for unfavourable loading state):

<table>
<thead>
<tr>
<th>( N_0 )</th>
<th>Standard algorithm</th>
<th>Neuron network algorithm</th>
<th>Real data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.048 / -0.049</td>
<td>-0.047 / -0.048</td>
<td>-0.047 / -0.050</td>
</tr>
<tr>
<td>2</td>
<td>6.43 / 6.47</td>
<td>6.24 / 6.53</td>
<td>6.38 / 6.62</td>
</tr>
<tr>
<td>3</td>
<td>-1.68 / -1.67</td>
<td>-1.81 / -1.71</td>
<td>-1.63 / -1.73</td>
</tr>
<tr>
<td>4</td>
<td>-6.78 / -6.64</td>
<td>-6.57 / -6.74</td>
<td>-6.72 / -6.78</td>
</tr>
<tr>
<td>5</td>
<td>4.83 / 4.96</td>
<td>4.95 / 5.05</td>
<td>4.87 / 5.01</td>
</tr>
</tbody>
</table>

The experimental results show a practically acceptable quality of solution for identification and forecast problems in conditions of uncertainty and incompleteness of initial information. It permits a conclusion about the possibility to concentrate measurement information in few characteristics determining the structure of cognitive image. So it is possible to reduce ANN resources to solve practical problems of damaged ship dynamics in waves.

7. Conclusions

Our results show that rational use of ANN and GA permits to ensure flexibility and adaptability with respect to variable external conditions, retention of stable high quality of operation and realization of unattainable earlier high level of computational power. Adaptive properties, parallelism and possibility of non-linear data transformation open perspectives of practical realization of computational technology based on new principles of information processing in on-board real time IS.

Synergetic effect of joint use of fuzzy systems, neural network theory, evolutionary modelling and cognitive paradigm forms scientific basis for deep knowledge integration in complex IS ensuring identification and forecast of damaged ship behaviour on waves.
References


The Application of Fire and Evacuation Simulation in Ship Design

Edwin R. Galea, Steve Gwynne, Peter J. Lawrence, Darren Blackshields, John Ewer, Zhaozhi Wang, Nathan Hurst, Nigel Mawhinney
Fire Safety Engineering Group, University of Greenwich, Old Royal Naval College, 30 Park Row London SE10 9SL, UK, s.gwynne@gre.ac.uk

Abstract

Fire and evacuation models with features such as the ability to realistically simulate the spread of heat and smoke and the human response to fire as well as the capability to model human performance in heeled orientations linked to a virtual reality environment that produces realistic visualisations of the modelled scenarios are now available and can be used to aid the engineer in assessing ship design and procedures. This paper describes the maritimeEXODUS ship evacuation and the SMARTFIRE fire simulation model and provides an example application demonstrating the use of the models use in performing fire and evacuation analysis for a large passenger ship partially based on the requirements of MSC circular 1033.

1. Introduction

However remote the possibility or difficult the task, ship evacuations do occur. In the wake of several prominent maritime disasters and in light of the growth in the numbers of high density, high-speed ferries and large capacity cruise ships, there is a growing interest in the marine industry in issues of evacuation of passengers and crew at sea. The High Speed Craft (HSC) Code introduced the concept of performing critical path analysis of the evacuation arrangements, SOLAS regulation II-2/28 1.3 required Ro-Ro passenger ships built after 1 July 1999 to have an early design stage evacuation analysis performed and in June 1999 IMO issued Guidelines for a Simplified Evacuation Analysis on Ro-Ro passenger vessels in the IMO MSC Circular 909.

IMO has established prescriptive times for the performance of an evacuation from passenger vessels, and owners are required to demonstrate compliance through full-scale tests. Demonstrating compliance with evacuation requirements through full-scale evacuation exercises poses considerable ethical, practical and financial problems that bring into question the value of their overall contribution to passenger safety. Consequently this form of proof of concept is impractical as a routine aid to assessing ship safety and at best provides an index allowing comparative evaluation, but providing little indication of what might happen in a real emergency. In response, IMO introduced prescriptive design regulations for passenger ships and defined some maximum target times for steps in the process. E.g., in the SOLAS Chapter III Regulation 11, the maximum time from the time the abandon ship signal is given to prepare all survival craft for abandonment is set at 30 minutes and the maximum abandonment time (i.e. for mustered passengers) is set in SOLAS III/21.1.4 as 30 minutes. However, it is recognised that these times are arbitrary and fail to address the unique situations that can arise in a given situation.

More sophisticated evacuation analysis through the use of computer simulation is both desirable and achievable. In recognition of this, IMO - through a Correspondence Group (CG) of the Fire Prevention Sub-Committee FP46 - has been working to set standards for the adoption of sophisticated evacuation modelling techniques. In addition to reviewing the simplified analysis with a view to drafting guidelines for application to Ro-Ro passenger and other passenger ships, the CG has worked towards generating standards for more sophisticated evacuation simulation tools, developing validation guidance for these tools, and addressing how these analyses are to be applied to existing ships. In February 2002, such a methodology was put to FP46, intended to cover the analysis of all new built passenger ships with a recommendation that it be applied to all passenger ships. The methodology outlined covered both simplified (i.e. hydraulic evacuation analysis) and advanced (i.e. the use of evacuation modelling tools) techniques for the simulation of evacuation. This approach was accepted by FP46 and in the 75th Session of MSC (in May 2002) was formally adopted as the Interim
Guidelines for evacuation analysis of new and existing passenger ships including RO-RO, IMO (2002). These guidelines define two benchmark scenarios (along with two variants) that must be simulated as part of the certification process. These are defined as the “night” and “day” scenarios. While arbitrarily defined, they establish a baseline performance for the vessel and crew allowing comparison with both the set target time and alternative designs. The scenarios only address the mustering or assembly phase of the evacuation and involve conditions of dead calm (i.e. zero list, heel and roll) and do not explicitly take into consideration the impact of fire. A safety factor is added to the predicted muster time to allow for these omissions. The guidelines also set out validation/verification requirements that the software must satisfy before it is considered suitable for use in certification applications, which include 11 test cases and software documentation requirements, IMO (2002).

In the building industry, performance based fire safety engineering requires the determination of essentially two factors, the time available to the occupants for safe egress, sometimes referred to as the Available Safe Egress Time (ASET) and the time that is required by the occupants to evacuate from the structure, often referred to as the Required Safe Egress Time (RSET). For the building to be considered acceptable, the RSET (plus some safety factor) should be less than the ASET, ISO (1999). For a particular fire/evacuation scenario, these calculations are typically performed in isolation, with the ASET being determined through fire simulation - either zone or field, Galea (1989a) - and the RSET being determined through an evacuation calculation or simulation, Gwynne et al. (1999). For the scenarios under consideration, fire simulation is typically used to determine at what time non-survivable conditions develop within the enclosure, e.g. when the smoke level reaches a particular critical height deemed detrimental to evacuation or when the radiative fluxes reach a critical value leading to the onset of flashover. The evacuation calculation would then be used to determine if the people within the structure can evacuate before these critical conditions develop. However, today’s Computational Fire Engineering (CFE) tools such as fire field models and complex evacuation models are capable of much more, such as the prediction of toxic gas and smoke generation and distribution, oxygen depletion, etc, and the reduction in travel speed due to smoke obscuration, incapacitation due to the inhalation of toxic products, impact of irritant gases on individual evacuation etc. Using such CFE tools it is possible to combine the calculation of the ASET and RSET and determine if the outcome of a given scenario is acceptable by using more sophisticated measures, e.g.: zero fatalities result from the simulated scenario or occupant exposure to fire effluent are limited to acceptable threshold values. Using such an approach a more meaningful comparison could be made between alternative structural configurations, active/passive systems, evacuation procedures or even between the nature of the building materials.

Here we demonstrate the current capabilities of this approach using the state-of-the-art CFE tools maritimeEXODUS, Galea (2000), Galea et al. (1994,2001a,b) and SMARTFIRE, Taylor et al. (1997), Ewer et al. (1999), Wang et al. (2001).

2. The ship evacuation process

The term “evacuation” when applied to maritime applications refers to all activity from the sounding of an alarm to leaving the ship. It is generally divided into mustering/assembly (passengers assemble at pre-defined assembly points onboard the ship) and abandonment phase (passengers actually leave the vessel using various means to transfer to other vessels, survival craft, or the water). From the modelling perspective, there are at least five components to ship evacuation: recognition by passengers of the need for emergency action, preparatory actions/behaviours, progressive evacuation to place of relative safety, preparation/deployment of escape system and eventually abandoning the vessel. To truly model ship-based evacuation it is essential to address all of these components. To perform the required simulation reliably requires an evacuation model with the appropriate set of capabilities and access to the necessary data. Furthermore, the scenario under consideration may be under conditions of calm or involve situations with list or roll. This will affect not only the nature of the data required but also the capabilities of the model. Throughout the evacuation process complex contra-flows can develop within the passageways and staircases. These can be formed by flows of passengers with different goals, e.g. by passengers attempting to find companions, collect life jackets and warm clothing and locate muster stations. Crew members can also create contra-flow situations as they attempt to tackle the cause of the
emergency or reach assigned duty stations. When modelling human behaviour during evacuation, it is essential to represent the enclosure geometry, population and population behaviour.

In order to fully assess the potential evacuation efficiency of any structure (ship, building, or aircraft), it is essential to address the configurational, environmental, behavioural and procedural aspects of the evacuation process, IMO (2002). Evacuation models must tackle each of these issues if they are to truly address the evacuation situation. Traditional methods for evacuation design fail to address all these issues quantitatively preferring to rely almost totally on the designer “judgement” and a set of “prescriptive rules”. As these “prescriptive rules” have an almost total reliance on configurational considerations such as travel-distance and exit width they can prove to be too restrictive. Furthermore, as the traditional methods are generally insensitive to human behaviour or likely emergency scenarios, it is unclear if they indeed offer the optimal solution in terms of evacuation efficiency.

3. The maritimeEXODUS software

EXODUS is a suite of software tools designed by the Fire Safety Engineering Group of the University of Greenwich to simulate the evacuation of large numbers of people from a variety of complex enclosures. Research and development on EXODUS began in 1989. Today, the family of models consists of buildingEXODUS, Gwynne et al. (2001c), airEXODUS, Blake et al. (2002), with the most recent addition being maritimeEXODUS. The software allows designers, certification authorities and operators to incorporate human performance and environmental factors into the evacuation analysis.

The EXODUS software takes into consideration people-people, people-fire and people-structure interactions. It comprises five core interacting sub-models: the Passenger, Movement, Behaviour, Toxicity and Hazard sub-models. The software describing these sub-models is rule-based, with the progressive motion and behaviour of each individual being determined by a set of heuristics or rules. These submodels operate on a region of space defined by the GEOMETRY of the enclosure. The spatial and temporal dimensions within EXODUS are spanned by a two-dimensional spatial grid and a simulation clock (SC). The spatial grid maps out the geometry of the structure, locating exits, internal compartments, obstacles, etc. and can involve multiple decks, connected by staircases. The structure layout can be specified automatically using a DXF file produced by a CAD package or manually using the interactive tools provided. Internally the entire space of the geometry is covered in a mesh of nodes that are typically spaced at 0.5m intervals. Each node represents a region of space typically occupied by a single passenger. In addition to the representation of the structure itself, the abandonment system can also be explicitly represented within the model, enabling individual components of the abandonment system to be modelled individually, Galea et al. (2001a). By providing a modular system, a variety of complex and diverse abandonment systems can be represented within maritimeEXODUS from several LSA (life saving appliances) components.

The MOVEMENT SUBMODEL controls the physical movement of individual passengers from their current position to the most suitable neighbouring location, or supervises the waiting period if one does not exist. The PASSENGER SUBMODEL describes an individual as a unique collection of defining attributes and variables such as name, gender, age, maximum unhindered fast walking speed, maximum unhindered walking speed, response time, agility, etc. The HAZARD SUBMODEL controls the atmospheric and physical environment, distributing pre-determined fire hazards such as heat, radiation, smoke concentration and toxic fire gas concentration throughout the atmosphere whilst controlling the availability of exits. EXODUS does not predict these hazards but accepts hazard data either from experimental measurements or numerical data from other models including a direct software link to the CFAST fire zone model, Peacock et al. (1993). A link to the SMARTFIRE fire field model will be demonstrated below. The TOXICITY SUBMODEL determines the effects on an individual exposed to toxic products distributed by the hazard submodel. These effects are communicated to the behaviour submodel, which in turn feeds through to the movement of the individual. To determine the effect of the fire hazards on passengers, EXODUS uses a Fractional Effective Dose (FED) toxicity model, Purser (1996), which considers the toxic and physical hazards associated with elevated temperature, thermal radiation, HCN, CO, CO₂ and low O₂ and estimates the time to incapacitation. In addition to this behaviour, the passengers are able to respond to the environmental conditions by adjusting their behaviour. The BEHAVIOUR SUBMODEL incorporates adaptive
capabilities that include, structural knowledge, reaction to communication, affiliative behaviour, occupant motivation and reaction to fire hazards. The submodel determines the passenger’s response to the current situation and passes its decision on to the Movement Submodel. The Behaviour submodel functions according to global behaviour (which involves implementing an general escape strategy), which may be modified or overridden through local behaviour, which includes such considerations as determining the occupants initial response, conflict resolution, overtaking, the impact of list and heel upon passenger movement and the selection of possible detouring routes. In addition a number of localised decision-making processes are available to each individual including the ability to customise their egress route according to the levels of congestion around them and the environmental conditions and the social relationships within the population, Gwynne (2000). With regard to the environmental conditions, passengers will stagger through smoke filled environments and may also elect to crawl. As the smoke concentration increases and visibility decreases, the travel speed of the occupants is reduced according to experimental data, Jin (1978), Jin and Yamada (1989). The user is also able to assign passengers and crew a list of tasks to perform via the behaviour model.

Another important aspect of human behaviour is the manner in which passengers react to the ship orientation. The movement rates of passengers in corridors on stairs and through doorways at various angles heel/trim is represented within the model and based on data generated from large-scale trials using the BMT Fleet Technology SHip Evacuation Behaviour Assessment (SHEBA) facility, www.shebafacility.com, fseg.gre.ac.uk/fire/net21.html, and small-scale data using the TNO Ship Motion Simulator (SMS), Bles et al. (2001). Within maritimeEXODUS the impact of the TNO/SHEBA data is influenced by several factors (age, gender, direction of movement, terrain, trim and heel angles, presence of a lifejacket). All of these factors are considered in the calculation of the impact heel/trim on travel speed. The data is presented as a reduction factor over the normal movement rates at 0° heel. The TNO data is used for trim, the SHEBA data for heel. In situations involving both heel and trim, the greater of the two reductions in travel speeds is used.

For ease of use, maritimeEXODUS has three types of pre-defined emergency itineraries that can be selected from a pull-down menu. These are intended to represent the type of procedures typically encountered in evacuations involving (i) Cruise ships, (ii) RORO ferries and (iii) High Speed Craft. The three types of evacuation procedure differ as to the number and type of the specific stages involved. In cruise ship evacuation, passengers are first required to collect their life vest from their predefined location (i.e. cabin) before moving to their relevant muster station prior to abandoning the vessel via an LSA. In RORO ferry evacuation, passengers are required to assemble at a relevant muster station, where life vests are issued before they abandon the vessel. Crew members are typically required to perform additional tasks during emergencies such as damage control, fire fighting, perform searches of given areas, oversee mustering, enable LSAs etc. To achieve this within maritimeEXODUS crew members can be assigned their own user-defined itinerary.

maritimeEXODUS produces both graphical and textual output. Interactive two-dimensional animated graphics are generated as the software runs allowing the user to interrogate occupants and events as they occur. The graphics can be displayed in individual mode, population density mode (where a colour contour fill is used to represent the number of people per square meter) or in footfall contour mode (which provides an indication of the number of passengers to have used a particular route during the simulation). askEXODUS has also been developed that enables large data output files from numerous runs to be analysed selectively and efficiently. The post-processor virtual-reality graphics environment vrEXODUS provides an animated three-dimensional representation of the evacuation, Fig.1. maritimeEXODUS is stochastic in nature as certain behaviour rules (e.g. conflict resolution), are probabilistic in nature. Thus every time a simulation is repeated a slightly different evacuation time will result, necessitating the repetition of many simulations to generate a distribution of results. It is the distribution of results that must be analysed when determining the outcome of an evacuation.
4. SMARTFIRE

The SMARTFIRE V3.0 software was used for the fire simulations in this study. SMARTFIRE is an open architecture CFD environment written in C++ that is comprised of four major components: CFD numerical engine, Graphical User Interfaces, Automated meshing tool and the Intelligent Control System. The CFD engine in SMARTFIRE has several special CFD features required for fire field modelling, Galea (1989b). These include a six-flux radiation model, Kular et al. (1991), a multiple ray radiation model, Raithby and Chui (1990), provision for heat transfer through walls, a volumetric heat release model or gaseous combustion model (using the eddy dissipation model) to represent fires, Lewis et al. (1990), smoke modelling and turbulence (using a two equation k-ε closure with buoyancy modifications), Galea (1989b). Within SMARTFIRE the user can define a range of scalar variables. These can be used to represent the transport of products such as toxic gases and smoke. SMARTFIRE uses three-dimensional unstructured meshes, enabling complex irregular geometries to be meshed. The code uses the SIMPLE algorithm for pressure-velocity coupling and can solve turbulent or laminar flow problems under transient or steady state conditions. The software has undergone considerable validation.

The fire environment to which the passengers are exposed in maritimeEXODUS can be determined by means of the CFAST zone model. As part of on-going research, a software link to CFD based fire models is developed. In this paper we explore the linkage of maritimeEXODUS to the CFD fire model SMARTFIRE. The link is achieved through the use of a SMARTFIRE routine that converts the fire data generated by SMARTFIRE into a format that can be read by maritimeEXODUS. To harmonise the three-dimensional control-volume discretisation used in SMARTFIRE with the meshing and zoning system used within maritimeEXODUS, a volume averaging technique is used. This effectively groups together potentially large numbers of CFD cells, averaging the data within them to allow easy use within maritimeEXODUS. The averaging is also extended over time, so as to provide the evacuation simulation with the data required at times appropriate to the evacuation simulation. Within each of these zones the information relating to two separate user defined heights (e.g. at head and waist height) is combined so as to provide an indication as to the conditions within the specified zone at those heights. This technique was developed as part of a European Union funded project called Fire Exit which is part of the Growth Programme. The information provided will relate to the temperature, the gas concentrations, thermal radiation from any hot layer and smoke concentration.

Fig.1: vrEXODUS output showing the top three decks of a 10 deck passenger ferry during passenger mustering. The lowest deck shown (deck 8) is the muster deck.
5. Demonstration application involving a large passenger ship

5.1 Test case

To demonstrate the operation of maritimeEXODUS, the software is applied to a hypothetical ship layout using the IMO night scenario specification as a guide to the analysis. The analysis is repeated using a fire case.

A large passenger ship consisting of 10 decks is defined within maritimeEXODUS using CAD drawings. Fig.2 shows two decks. The ship has a capacity of 650 passengers. The ship is divided into three vertical fire zones accommodating 348 passengers in the first fire zone, 52 passengers in the second fire zone and 249 passengers in the third fire zone. Fig.1 shows views of top three decks generated using vrEXODUS.

![Diagram of Deck 7](image1)

(a) Deck 7

![Diagram of Deck 8 (Muster Deck)](image2)

(b) Deck 8 (Muster Deck)

Fig.2: Sections of the 10 deck hypothetical ship

The lowest passenger deck is Deck 6 while Deck 10 is the highest. The assembly areas are located on Deck 8 and there are two for fire zone 1, two for fire zone 2 and four for fire zone 3. There are two passenger decks below the muster deck accommodating 400 passengers and two passenger decks above, accommodating 250 passengers, Table 1. Each deck of the first fire zone is serviced by four staircases located within the far corner of the fire zone connecting each deck. The second fire zone only possesses a single staircase centrally located within the fire zone. Fire zone 3 has a similar layout to fire zone 1. All the stairs are similar in construction and are narrow, capable of allowing only a single lane of passengers to use the stairs. The only exception is the dual lane staircase in fire zone 2. All stairs are dog-legged and have a landing located at their mid-point. Passenger cabins are located on both decks 6 and 7 in fire zone 1 and 2 and both decks 9 and 10 in fire zone 3. A large theatre is located on deck 7 in fire zone 3, dinning areas and bars are located throughout deck 8 and within fire zone 2 on deck 9.
Table I: Number of passengers in each fire zone per deck

<table>
<thead>
<tr>
<th></th>
<th>Fire Zone 1</th>
<th>Fire Zone 2</th>
<th>Fire Zone 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deck 6</td>
<td>172</td>
<td>28</td>
<td>-</td>
</tr>
<tr>
<td>Deck 7</td>
<td>176</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td>Deck 8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Deck 9</td>
<td>-</td>
<td>-</td>
<td>150</td>
</tr>
<tr>
<td>Deck 10</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
</tbody>
</table>

IMO stipulates that two main benchmark evacuation scenarios be examined, the “day” and “night” cases. There is also a variation of each of these scenarios that must also be considered making a total of four different benchmark scenarios that must be investigated. Here we only consider the main night scenario and two other variations of this scenario not currently required by IMO. In the cases considered here, all the passengers are initially located in their cabins and have response times varying from 7-13 minutes as required by the IMO specification. This time is intended to represent the time required to arouse a sleeping passenger and get them dressed and ready to move to the assembly station. The passenger attributes are defined according to the IMO (2002) specification. In Scenario 1 the vessel is level (0° heel). Scenarios 2 and 3 are like Scenario 1 but 10° and 20° heel, respectively. Only the quantitative changes in the movement speeds of the evacuees due to heel are represented, but not any behavioural changes that may result in conditions heel.

The results are presented in two sections, one dealing with the standard IMO type scenario and another dealing with the combined evacuation and fire scenario. All of the simulations presented here were run on a 1.9 GHz Pentium 4 PC with 2 Gb of RAM. When run in maritimeEXODUS interactive graphics mode each simulation requires approximately 1’35” to complete. The simulations presented here were run in batch mode which significantly reduces the time required to complete the simulations.

5.2 IMO compliant scenarios (without fire)

To be fully compliant with IMO (2002) regulations, it is necessary to perform a total of 50 repeat simulations for each of the four IMO benchmark scenarios. These repeat scenarios are made up from five different random mixes of people and each random mix is run 10 times. The representative assembly time for the vessel for each scenario would be determined from the distributions by selecting the time that is larger than 95% of the generated values. Finally, the representative assembly time for the vessel is taken as the largest of the four values. Once this time has been determined, a safety margin of 10 minutes is added to the calculated time, IMO (2002). This is intended to account for all of the assumptions involved in the modelling approach. For the cases presented here, we present the minimum, average and maximum values from which we determine the 95% value representing the assembly time for the vessel.

(a) Scenario 1 (0° heel)

The average total time to muster is approximately 15’32” with a maximum muster time of 15’58”, Table II. Fire zone 1 is the last to muster in all of the cases considered. Of the three fire zones, fire zone 1 has the largest number of passengers and all the passengers are below the muster deck, requiring them to travel upstairs, thereby incurring the slowest travel speeds. The deck clearing times, Table III, represent the time required to completely clear the deck of all passengers, those starting on the deck in question and those passing through the deck on their way to the muster deck. The lower decks take longer to clear than the upper decks. This is because there are more passengers located on the lower decks and passengers take longer to climb stairs than to descend stairs.

Fig.3 shows the cumulative muster times generated by the 50 simulations. Depicted are the maximum time to muster for each passenger for all 50 simulations, the minimum time to muster and the average time to muster. The muster time graph depicts a characteristic “S” shaped appearance. The slow start of the curve is due to the long and widely distributed response times exhibited by the passengers in the
night scenario. Once the flow of passengers begins, there is a rapid increase in arrival rate into the muster zone over a 50 s period. After this initial build up, the arrival rate settles down to an almost uniform value as there is a constant supply of passengers into the assembly area. Over the last 100 s of the assembly, the arrival rate begins to drop off as the supply of passengers begins to diminish. The 95% value for the mustering time for these simulations is 15’57”. Thus, the predicted time to muster for this vessel under the IMO night conditions is 15’57”. With the IMO safety margin of 10 minutes added, the vessel will require 25’57” to muster and so satisfies the IMO standard.

Table II: Range of assembly times generated by maritimeEXODUS for 0° heel

<table>
<thead>
<tr>
<th></th>
<th>Fire Zone 1</th>
<th>Fire Zone 2</th>
<th>Fire Zone 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>14’ 59”</td>
<td>13’ 34”</td>
<td>13’ 42”</td>
</tr>
<tr>
<td>Avg</td>
<td>15’ 32”</td>
<td>14’ 00”</td>
<td>14’ 32”</td>
</tr>
<tr>
<td>Max</td>
<td>15’ 58”</td>
<td>14’ 43”</td>
<td>15’ 24”</td>
</tr>
</tbody>
</table>

Table III: Clearing times for each deck at 0° heel

<table>
<thead>
<tr>
<th>Deck 6</th>
<th>Deck 7</th>
<th>Deck 9</th>
<th>Deck 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>14’ 01”</td>
<td>14’ 28”</td>
<td>14’ 04”</td>
</tr>
<tr>
<td>Avg</td>
<td>14’ 18”</td>
<td>14’ 43”</td>
<td>14’ 19”</td>
</tr>
<tr>
<td>Max</td>
<td>14’ 39”</td>
<td>15’ 02”</td>
<td>14’ 41”</td>
</tr>
</tbody>
</table>

Fig.3: Cumulative muster time for Scenario 1 (0° heel)

According to the IMO regulations, regions in which the local population density exceeds 4 persons/m² for a duration exceeding 10% of the overall assembly time are identified as congestion regions. As part of the analysis, these congestion regions must be identified and removed, either through procedural or structural alterations. As the assembly time for this vessel is 957 s, congestion should not exceed 4 persons/m² for periods of 96 s.

One of the outputs generated by maritimeEXODUS for each passenger is a measure of the amount of time the passenger wastes in congestion. This is known as the Cumulative Wait Time (CWT). This can be averaged for each passenger in the simulation and a representative average CWT generated for each person and for each simulation. The average CWT for these 50 simulations was 6 s. As the average mustering time was 15’32”, it is clear that significant congestion does not develop within these simulations. So the protracted passenger response times result in low levels of congestion throughout the vessel for the duration of the mustering process.

A more detailed analysis of the congestion can be derived using the maritimeEXODUS population density mode visualisation. Using this feature it is possible to generate a coloured contour map of the population density throughout the vessel as the evacuation takes place.
The most heavily congested areas occur at the base of the staircases on deck 7 leading to the muster deck, Fig.4(a). Congestion in this area achieved values of between 2.2 and 3.5 persons/m² of a period of 19 s. For the remainder of the simulation, congestion in this area never exceeded 2.2 persons/m². As expected, the staircases also proved to be areas of heavy congestion. On the landings, the longest periods of congestion lasted for approximately 28 s where concentrations of people reached levels of between 2.2 persons/m² to 3.5 persons/m². For a brief period of time lasting approximately 2 to 4 s, congestion on the staircase landing reached 4 persons/m². At all other times the congestion levels throughout the vessel were less than 2.2 persons/m².

The vessel is therefore deemed to comply with the requirements of the IMO night-time assembly scenario. If we repeat this scenario using an instant response time distribution (i.e. all of the evacuees respond immediately and simultaneously) rather than the IMO specified response time distribution, we find a very different situation, Fig.4(b). There are significant congestion regions developing in the vicinity of all the stairs, we also note that the average CWT reaches 42 s. Thus if in the night scenario, passengers react immediately to the call to evacuate, significant congestion can be expected to develop.

(b) Scenarios 2 and 3 (10⁰ and 20⁰ heel)
At 20⁰ heel, the average time to muster is increased to approximately 16 minutes with the maximum muster time reaching 16’22”, Table IV. Applying the 95% ruling to this vessel would suggest that the representative assembly time for this vessel at 20⁰ heel is 16’30”. With the IMO safety added, the vessel will require 26’30” to muster and so meets the required standard.

![ IMO night scenario](image1) ![ Instant response time scenario](image2)

Fig.4: Population density contours

Table IV: Range of assembly times generated by maritimeEXODUS for each muster zone at 20⁰ heel

<table>
<thead>
<tr>
<th>Zone</th>
<th>Fire Zone 1</th>
<th>Fire Zone 2</th>
<th>Fire Zone 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>15’ 21”</td>
<td>13’ 21”</td>
<td>13’ 55”</td>
</tr>
<tr>
<td>Avg</td>
<td>16’ 00”</td>
<td>14’ 30”</td>
<td>14’ 58”</td>
</tr>
<tr>
<td>Max</td>
<td>16’ 32”</td>
<td>15’ 51”</td>
<td>16’ 14”</td>
</tr>
</tbody>
</table>

Fig.5 shows the cumulative muster times for 20⁰ heel, the equivalent of Fig.3 for Scenario 1. The impact of increasing the heel angle on the overall evacuation efficiency can also be seen by looking at the rate at which passengers arrive at the various muster zones. E.g., after 10 minutes, 83 passengers have assembled in Fire Zone 1 with 0⁰ heel, 81 passengers have assembled with 10⁰ heel and 70 passengers have assembled with 20⁰ heel. Thus 20⁰ heel has a significant effect on the mustering time, while a 10⁰ heel has little effect. As the heel angle has increased we generally find that the time to muster also increases. At 0⁰ heel, we find that the time to muster is 15’57”, at 10⁰ heel the muster time is 15’56” and at 20⁰ heel the muster time is 16’30”.

63
Note: maritimeEXODUS V3.0 does not include possible changes to passenger behaviour as a result of heel or trim. The only impact is to the travel speeds. At large heel angles it is possible that corridors that allowed the passage of several passengers abreast could effectively become single lane passage ways, thereby further reducing the flow rate and increasing the time required to muster.

5.3 Scenario involving fire

The only new factor in this simulation is the presence of an evolving fire, located in a cabin on the lowest passenger deck, deck 6. The data for this incident was produced using the SMARTFIRE CFD model. For the purposes of this example, only the fire hazards associated with elevated temperature, thermal radiation and smoke concentration are considered. While maritimeEXODUS is capable of including the impact of toxic fire gases they are ignored here. The SMARTFIRE fire simulation produced 22’32” worth of fire data. The fire is assumed to have started in a cabin on deck 6 within fire zone 1 and the fire effluent is allowed to spread throughout the fire zone through decks 6, 7 and 8. No attempts at controlling the fire are considered in this simulation however, the fire effluent is prevented from spreading to other fire zones. As in the previous examples, the muster stations are located on deck eight. As fire effluent spreads to the muster deck in fire zone 1, passengers in this zone are allowed to move to the neighbouring fire zone when they reach the muster deck.

Given that this scenario is beyond the IMO requirements, some minor modifications have been made in the nature of the results presented. The times presented relate to the time for the entire population to muster. This was felt more appropriate when dealing with the fire scenario. Only a single scenario is presented involving zero heel or list angle. The evacuation simulations were repeated 10 times with the location of the individuals swapped at the end of each simulation.

(a) The Fire Data Produced by SMARTFIRE
SMARTFIRE was used to determine the evolution of the fire incident within the vessel. The geometry used for the CFD simulation consisted of one vertical fire zone (fire zone 1) and three decks forming a volume 50.4 m long by 28.0 m wide by 8.8m tall. It was assumed that the door to the fire compartment was open and that the doors to all other compartments were closed. While SMARTFIRE can include the impact of forced ventilation these simulations did not include the action of mechanical ventilation. Thus, for the purposes of this demonstration simulation, it was assumed that the passenger cabins were not affected by the fire effluent and that the effluent could spread freely throughout the fire zone. Therefore heat and smoke generated by the fire escaped from the compartment of fire origin and travelled along corridors, up stairwells and through open public spaces.
The fire was simply modelled as a volumetric heat and smoke source. The fire source was given the rough dimensions of a single bed (2.1m long by 0.8 m wide by 0.5 m high) and placed at the back of the cabin, Fig.6(a). The heat release rate used in the simulations are equivalent to that of a mattress with a peak Heat Release Rate (HRR) of around 1 MW, fire.nist.gov/fire/fires/fires.html. The HRR curve, Fig.6(b), consists of a \( t^2 \) fast growth rate \( (c = 0.0469 \text{ kW/s}^2) \) fire which grows for 150 s to a peak HRR of approximately 1 MW and then remains constant. The smoke production rate was defined following the same distribution; smoke production in kg/s grew as \( t^2 \) with a proportionality constant of \( c = 5 \times 10^{-7} \text{ kg/s}^3 \) for 150 s to a peak mass release rate of 0.01125 kg/s and was then constant, Fig.6(b). Using the above geometry and heat and smoke release rates, SMARTFIRE determined the state of the fire zone at 1 s time intervals for approximately 20 minutes. The smoke concentration predicted by SMARTFIRE on deck 6, Fig.7(a), 150 s into the simulation yielded the representation in maritime-EXODUS, Fig.7(b), and vrEXODUS, Fig.7(c).

For the purposes of the data transfer link between SMARTFIRE and maritimeEXODUS, output zones were applied across the domain. These zones were used to calculate and export averaged values over two set heights, a ‘low’ height of between 0.3m and 0.8m, and a ‘high’ height of between 1.5m and 2.0m. In total some 79 non-uniform zones were defined, varying in length from 4.9m up to 17.6m. On the fire deck the zone length did not exceed 9.0m.

![Fig.6(a): Fire zone 1 deck 6 showing fire compartment](image)

![Fig.(b): Smoke and heat release rates used in SMARTFIRE fire simulation](image)

(b) Evacuation Results produced by maritimeEXODUS
While a vast amount of detailed data can be generated for these simulations, only a summary of the results will be presented here. The average time to muster is approximately 31’03”, this compares to 15’32” without the fire, Table II. The presence of the fire is predicted to double the required muster time. Fire zone 1 is the last to muster in all of the cases considered. This is to be expected as this is the fire zone that contains the fire. In addition, the passengers in fire zone 1 are forced to muster in fire zone 2. This was necessary as fire effluent reached the muster zone on deck 8.

Table V shows the deck clearing times. As already seen in the earlier cases, the decks below the muster deck take longer to clear than the upper decks. The upper decks in the fire case take approximately the same amount of time to clear as they did in the non-fire case. This is again due to the location of the fire. The fire is located on the lower decks and only affects the lower decks. The passengers passing through the smoke are slowed down to a crawl, delaying their passage through the deck and prolonging their exposure to the life threatening fire hazards. The 95% value for the mustering time for these simulations is 32’37”. Thus, the predicted time to muster for this vessel under the IMO night conditions including a fire is 32’37”. However, this time does not include the IMO specified safety factor of 10 minutes. While it may still be appropriate to add a safety factor to account for issues excluded from these predictions (e.g. heel), it is not appropriate to add the entire safety factor to the predicted muster time as this simulation includes some of the features that the safety margin is intended to compensate for i.e. fire. While it may be argued that the vessel is capable of
meeting the IMO target muster time of 40 minutes – even in situations involving fire – there are other factors that should be considered in judging the success or failure of this vessel under the scenario conditions.

<table>
<thead>
<tr>
<th></th>
<th>Deck 6</th>
<th>Deck 7</th>
<th>Deck 9</th>
<th>Deck 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average non-fire case</td>
<td>14’ 18”</td>
<td>14’ 43”</td>
<td>14’ 19”</td>
<td>13’ 58”</td>
</tr>
<tr>
<td>Average fire case</td>
<td>18’ 18”</td>
<td>20’ 06”</td>
<td>14’ 18”</td>
<td>13’ 47”</td>
</tr>
</tbody>
</table>

In the fire scenario there are a number of fatalities. Over the 10 repeat simulations, the number of fatalities varied from 26 to 27 passengers. The first fatality occurred at around 7’6” into the simulation while the last fatality occurred at around 19’30” into the simulation. All these fatalities occurred on deck 6 with the passengers involved being initially located close to the compartment of fire origin, Fig.7(d). Each of the fatalities travelled on average only 5 m before they succumbed to the intensity of the fire. The main reason for their demise was their proximity to the fire and their lengthy response time. The fatalities had an average response time of 10’20”, with the shortest response being 7’10”. The only way in which these individuals could have survived the incident would have been by responding much sooner to the alarm.

In addition to the fatalities it is also possible to determine the level of likely injury that the survivors have sustained. Although the majority of the surviving passengers had little or no interaction with the deteriorating conditions, several of them could be considered to be near death. Two passengers are expected to have severe exposure to thermal radiation having received a dose of thermal radiation greater than 70% of that which is required to cause incapacitation. All those exposed to severe doses of thermal radiation were initially located on the fire deck (i.e. deck 6). In total 25% of the passengers on deck 6 received doses of thermal radiation greater than 10% of that required to cause incapacitation. Had toxic fire gases been included in the simulation, the death toll and the injury level could be expected to be much worse. Conversely, it should be remembered that the scenario did not include any active fire fighting measures such as the use of sprinklers or forced ventilation.
This scenario has demonstrated the manner in which maritime EXODUS and SMARTFIRE can be used to examine scenarios beyond the IMO regulations in order to investigate the success of the procedures on board, the impact of specified fire scenarios and suggesting potential solutions to these problems. The results presented are only a sub-set of that expected during a full analysis, indicative of the insight that could be gained.

6. Conclusions

Ship evacuation models will have a profound impact on safety at sea. They will be used by ship designers during the concept phase, classification societies for the certification of ship design and by ship operators for training both on shore and at sea. In the early stages of the design process these models will bring important issues of safety, evacuation, staffing and procedures to the fore of ship design in a manner that will be reliable, quantifiable and reproducible. In a similar process, classification societies will be able to quickly assess a proposed design, including the crew procedures and determine whether proposed designs meet acceptable standards. Finally, ship operators will be able to assess safety provision on-board as conditions including number, type and location of passengers, number of crew etc., change with each sailing or during a sailing. On-board versions of the software will allow ship operators to train crews and even re-direct fire fighting and passenger evacuation activities in response to the on-board situation. The software will have a similar impact on naval vessels, where issues such as lean manning, optimisation of crew movements during emergency and non-emergency situations as well as safety and evacuation are key to the design of efficient and well managed fighting machines.

By combining detailed fire simulation with evacuation simulation, it is possible to obtain detailed insight into the performance of both man and machine under emergency conditions involving fire. This insight can be used in order to investigate the success of the procedures on board, the impact of specified fire scenarios and suggesting potential solutions to these problems. In this way the ship of the future will be safer by design.

Acknowledgements

The authors are indebted to the Canadian Transportation Development Centre and the European Union Growth programme under Frame Work 5 for their financial support in the development of maritime EXODUS. Prof. Galea is also indebted to the UK CAA for their financial support of his chair in Mathematical Modelling.
References


IMO (2002), MSC Circular 1033

ISO (1999), Document ISO/TR 13387-8

JIN, T. (1978), Visibility through fire smoke, J. Fire and Flammability 9, pp.135-155


68


Speech-Interactive Virtual Environments for Ship Familiarization

Kenneth Wauchope, Stephanie Everett, David Tate, Tucker Maney, Naval Research Laboratory, Washington DC/USA. \{wauchope|everett|maney\}@aic.nrl.navy.mil, tate@itd.nrl.navy.mil

Abstract

When US Navy personnel first report on board a ship, they are generally unfamiliar with the overall layout and the locations of the offices and other spaces they will be required to access in performance of their daily duties. The use of computer-aided design in the development and construction of new ships provides a partial solution to this problem by making it possible to build interactive 3D virtual environments (VEs) that can be navigated and explored in advance of boarding the actual ship. Such environments, however, must be augmented with intelligent query answering and route finding capability to be fully useful. This in turn requires that the purely navigation-oriented VE interface be integrated with a symbol-based interface that can provide information such as compartment names and numbers. While such an interface is typically implemented as a GUI (Graphical User Interface) in most computer applications, speech interaction provides a highly natural alternative that offers minimal interference with the eyes/hands-busy task of virtual navigation. This paper describes two such speech-interactive VE interfaces we have developed, one for a Navy firefighting research and test ship and the other for a newly commissioned aircraft carrier.

1. Introduction

When US Navy officers and crew first report on board a ship, they are generally unfamiliar with the overall layout and the locations of the offices and other spaces they will be required to access in performance of their daily duties. The problem is compounded in the case of new ships, where even the pre-commissioning crew must learn its way around. Through an inefficient and lengthy process that may span several months, personnel learn the locations of areas important to them by asking directions, following others, or simply exploring. While diagrams and floor plans can be useful, they are no substitute for the visual experience of navigating the compartments, passages, stairways, ladders and decks of the actual physical space itself.

The increasing use of computer-aided design (CAD) in the development of new ships provides a partial solution to this problem, since CAD files can often be converted with little or no difficulty to three-dimensional (3D) graphical models that can be used to generate virtual walkthroughs, though often in the form of non-interactive movie files rendered in non-real time. If the model can instead be viewed interactively in real time with sufficiently persuasive visual effects, the technology is often referred to as Virtual Reality (VR), although we prefer the term Virtual Environment (VE) as being more inclusive and domain-neutral. By taking advantage of our innate human ability to rapidly process and interpret 3D visual and audio information, VEs provide a highly effective way of acquiring working mental models of new physical spaces without actually being there, Tate et al. (1997a), Witmer et al. (1996), Bliss et al. (1997).

The conventional VE, however, only provides a passive navigable space that gives the user the option of exploration mentioned earlier, but not the ability to ask for directions, be shown a path to follow, or be escorted about. To provide those additional capabilities, the VE must be augmented with a software agent able to respond to queries and commands about named entities such as compartments, levels and spatial orientations (e.g. “left”, “starboard”). This symbol-oriented interface must also be properly integrated with the existing VE display so that users can query or refer to their current location and orientation, whether stationary or in the middle of a walkthrough. Finally, the new software must include an automated route finding routine that can plot reasonable paths from one location to another and translate them into VE outputs such as animated walkthroughs or marked routes. We have named the resulting hybrid “Augmented VE” (AVE), since it is an exact parallel to so-called Augmented Reality (AR) technology in which an interactive information system is overlaid on the (real) 3D world via a device such as a see-through head-mounted display.
Over the past two decades, symbolic command- and query-oriented interaction with computers has typically been implemented using graphical user interface (GUI) technology. In a VE, however, the user’s hands and eyes are already occupied navigating and viewing the scene, making the equally eyes- and hands-busy GUI a relatively incompatible means of interaction. While it is possible to embed a GUI in the virtual world as a kind of “virtual PDA” or portable menu for issuing commands and getting information, recent advances in commercial speech technology provide the alternative of a completely eyes- and hands-free mode of symbol-based interaction offering minimal interference with the VE experience. Thus a “speech-interactive VE” can be defined as an AVE in which speech is an available interaction medium. This paper describes speech-interactive VE ship familiarization systems we have developed that integrate a spoken language interface and route finder with 3D models of portions of two US Navy ships, the Ex-USS Shadwell firefighting research and test ship and the USS Ronald Reagan (CVN-76) aircraft carrier. We believe such systems could be used to improve the efficiency of a crewmember’s familiarization process, either shore-side before reporting for duty or as an embedded training system on board ship.

2. Multimodal Ship Familiarization Tool

In a previous study, we conducted an experiment aboard the Shadwell to determine if VEs could be used to aid shipboard firefighters, Carhart et al. (1987). Our results showed that a measurable performance improvement can be achieved by exposing shipboard firefighters to immersive virtual environments as part of their mission preparation, Tate et al. (1997b), Williams et al. (1997). The metrics used in this test were the time required to complete the task and the number of wrong turns taken. Firefighters that used VE during their mission preparation showed measurable improvements in both metrics when compared to the control group that did not use VE (see Table I). These results suggested that the broader problem of general ship familiarization is also an area that could benefit from training in virtual environments, which led us to develop the Multimodal Ship Familiarization Tool (MSFT), Tate et al. (2000), Everett et al. (1998, 1999, 2000).

<table>
<thead>
<tr>
<th>Group</th>
<th>Navigation Task</th>
<th>Firefighting Task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (mm:ss)</td>
<td>Wrong turns</td>
</tr>
<tr>
<td>VE</td>
<td>1:54</td>
<td>0.6</td>
</tr>
<tr>
<td>Control</td>
<td>2:38</td>
<td>1.6</td>
</tr>
</tbody>
</table>

2.1. System Architecture

The key code and data modules of MSFT are the VE viewer application, 3D ship model, speech recognition and synthesis components, language command interpreter, ship compartment database, and route finder. Of these only the first two are typical of most VE applications; the additional components are what extend the VE system into a queriable tool capable of providing the user with ship compartment, directional and route information.

As shown in Figure 1, the VE viewer component and speech interface run as two separate processes communicating via TCP/IP sockets. The VE application runs on an Intergraph 500MHz Pentium III computer, and the speech interface runs either on the same machine or on a Dell 366MHz Inspiron 7000 laptop. All software was written using commercial off-the-shelf software development kits (SDKs) rather than proprietary products. The VE application was built using the World Toolkit from Engineering Animation, Inc., a cross-platform software development system for building 3D applications. The speech interface uses IBM’s ViaVoice ’98 Executive, a speaker-dependent continuous speech recognition system that requires minimal enrollment (training the system on a particular user’s voice). The speech interface was written using the ViaVoice SDK’s Speech Manager
API (SMAPI) and tags each recognized utterance with so-called grammar annotations, simplifying the implementation of the command interpreter by eliminating the need for a separate utterance-string parser.

![Fig. 1: Multimodal Ship Familiarization Tool architecture](image)

**2.2. 3D Model**

The *Shadwell* was built before CAD tools were available to naval architects, so no model was readily available that we could import into our system. Because of the ship’s ongoing role as a research facility, however, accurate CAD files had been developed for many of the areas of the ship. For our project, these files were expanded and combined to create a complete 3D model.

To be truly useful, MSFT required a model that included all the areas of the ship to which the crew has access, including state rooms, berthing spaces, mess areas, storerooms, electronics spaces, machinery rooms, weather decks, heads and passageways – a total of 147 compartments on the *Shadwell*. A highly detailed model of a structure this size could potentially create millions of polygons, which would require a high-end graphics computer to display effectively. Since we envisioned this system being used as a self-paced training system running on a desktop computer, we needed a model that was relatively simple in terms of geometric complexity, yet provided a realistic representation of the ship (see Figure 2).

![Fig.2: A view from the quarterdeck of the virtual Shadwell](image)
To reduce the load on the graphics-rendering engine and thereby maintain an acceptable frame rate on the display, some effort was taken to create simple yet realistic models of various components. Fine details of the ship’s construction, such as support beams and stiffeners, were omitted to help reduce the complexity of the model. We also reduced model complexity by simplifying the representation of the approximately 400 doors in the Shadwell. An example of this is the modeling of watertight doors, which are oval with a wheel in the center that operates a complicated latching mechanism. Texture map images that showed the wheel and latching mechanism were applied to the faces of the doors to give the correct appearance, and the corners of the doors were chamfered instead of rounded to significantly reduce the geometric complexity of each door. This created a realistic looking door that was not a burden to the graphics processor, even when replicated hundreds of times. The resulting model of the ship contains approximately 65,000 polygons, and can be rendered at a minimum of six frames per second on an Intergraph 500 MHz Pentium III.

Terrain following and collision detection are both enabled during program initialization. Terrain following causes the user viewpoint to remain a fixed distance above the decks and ladders, giving the appearance of walking around. Collision detection prevents the user from walking through walls, which means they must open doors and move through the model as they would have to move through the real ship. This may help to reinforce the cognitive modeling of the terrain more than if the user were allowed to “fly” freely about because it is more like the experience of actually walking through the real ship.

The user interface for the virtual environment supports both the desktop monitor and the head-mounted display (HMD). When using the desktop monitor, the mouse and keyboard control interaction with the environment. The left mouse button controls the user’s speed and direction of motion. The middle mouse button acts as a push-to-talk button to signal the beginning and end of an utterance, enabling the user to control both the VR application and the speech interface with a single mouse. When using the HMD, a hand-held joystick provides navigation and interaction with the VE. The joystick has buttons to control movement, to operate doors, and to signal the speech recognizer.

2.3. Speech Interface

Speech interaction allows the user to issue commands and queries to the VE system in a natural and intuitive way and receive informative feedback via synthesized speech, thus remaining fully engaged in the virtual experience even while wearing an HMD. The user can ask questions such as Where is Repair 2?, Which way is forward?, or Which deck am I on now?, and issue commands such as Put me on the forecastle or Show me how to get to the Control Room. While similar interface capability could be implemented graphically using a heads-up display and touch-activated popup menus embedded within the VE, that would introduce artificial visual distractions as well as the additional effort involved in scanning and selecting items using multiple point and touch operations.

The interface accesses the ViaVoice speech recognizer as a so-called “command and control” application, which requires that an application-specific language model be provided defining the set of utterances able to be recognized. The language model is in the form of a context-free grammar written in a simple BNF (Backus-Naur Form) notation called SRCL (Speech Recognition Control Language), and specifies both the syntactic patterns and the specific vocabulary that will be accepted. The speech grammar developed for MSFT uses a vocabulary of approximately 170 words and recognizes over 16,000 different utterances. The user can request information about the name, number and/or location of the space s/he is currently in (What’s the number of this compartment?), or about any of the other spaces in the database (Which deck is the Communications Center on?). Users can also request information about their current location (Where am I?, Which way am I facing?) or ask to be transported directly to any space (Put me in the Crew Mess).

Commands and queries identified by the speech recognizer are mapped to one of 20 function tags according to specifications written into the grammar. Table II shows some examples of utterance
patterns and their corresponding tag structures. The SRCL notation \([a \mid b]\) indicates that either \(a\) or \(b\) is required; parentheses indicate optional words or phrases. For example, \([\text{Take} \mid \text{Fly}] \text{ me (from here to) there}\) denotes that “Take me there”, “Take me from here to there”, “Fly me there” and “Fly me from here to there” are all valid utterances.

<table>
<thead>
<tr>
<th>Utterance Pattern</th>
<th>Tag Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>(How do I</td>
<td>(Show</td>
</tr>
</tbody>
</table>

After recognition, the tag structure and utterance string are passed to the command interpreter, which (depending on the function tag type) may access the compartment database, route finder, or VE application to get additional information before generating an appropriate action. To determine which space the user is in, the command interpreter gets the coordinates of the user’s current location from the VE process and uses the \(z\) coordinate (altitude) to determine which deck s/he is on. It then uses a ray-crossing algorithm to calculate which compartment on that deck (as defined by the set of vertices comprising its perimeter) contains the \(x,y\) point. Each time an utterance is processed, the index of the space that is the current topic of conversation is stored to allow the user to ask follow-up questions about the same space using anaphoric reference (\(it, there\)) rather than having to specify its name or number each time.

The verbal responses generated by the command interpreter are customized to match the syntactic structure of the user’s question. For example, if the user asks \(\text{Which compartment am I in}\? \tag{\text{ThisNameIn}}\) the system responds “You are in the X”; if the user asks \(\text{Which compartment is this}\? \tag{\text{ThisName}}\) the system responds “This is the X”. Using complete sentences as feedback both encourages the user to also speak in complete sentences (as required by the recognition grammar), and helps the user realize when a recognition error has occurred.

2.4. Compartment Databases

A typical CAD-based VE model is simply a collection of graphical elements (possibly grouped) containing no information about object properties, behaviors, or relationships. The Shadwell model contains a minimal amount of this type of information about each of the doors so that they can be operated individually, but does not contain any information about the compartments or how they are arranged relative to one another. To support the spoken natural language interface and automated route planning, it was therefore necessary to construct two supplemental databases. The first contains the name, number, location, and perimeter vertices for each of the compartments in the ship, and the second contains adjacency and connectivity information about the doors and compartments for use by the route finding algorithm.
2.5. Automated Wayfinding

The user can ask to be shown a path to a particular space (Show me how to get to the Control Room) or between any two spaces (How do I get from the Control Room to the Communications Center?). An automated route finding algorithm determines the best path between the specified locations by accessing a waypoint connectivity database representing the compartments and doorways in the Shadwell as nodes in an adjacency graph. 326 nodes are needed to accommodate all the possible paths between the 147 compartments. An additional database of path correction points is provided for cases where straight-line paths between nodes would cause the user to traverse non-navigable areas such as through walls or across floorless spaces.

The route finder is based on an implementation used in the US Air Force’s LACE military simulation system to generate on-road routes for ground vehicles, Anken (1989). It does a breadth-first search of the adjacency network starting from the point of origin, generating incrementally longer sub-paths until a path to the destination is found. Route quality is improved by a user-defined cost function that ranks intermediate results and expands the most promising sub-paths first. The cost function used in this project simply minimizes the total cost of all the nodes comprising the route. Assuming that the nodes are more or less evenly spaced throughout the ship, this would always find the shortest path between two points if all nodes had equal cost. By making some nodes more expensive than others, however, we can fine-tune the search to yield slightly longer paths that meet additional quality criteria, such as avoiding certain compartments or going out on deck only when necessary. For example, the nodes corresponding to doors that connect interior and exterior spaces are assigned a relatively high cost so that routes originating indoors tend to stay indoors, and those originating on the deck tend to remain outdoors whenever practical.

Once the complete path has been determined, the VE system displays the path as an animated sequence of arrows about 60 centimeters above the deck, pointing toward the destination. The arrows can be selectively hidden or shown so the visual aid will only be in effect when desired. Any compartment in the database can be the origin or destination of a path, and the path correction points incorporate additional segments into the path to avoid any obstacles. Finally, the command Take me there walks the user along the path under computer control.

3. Interactive Ship Familiarization System

The Interactive Ship Familiarization System (ISFS) is a recent transition of the MSFT system to a new application domain and software environment, Wauchope (2003). The new domain is a 3D model of the Island House structure (Levels 08 & 09) of the aircraft carrier USS Ronald Reagan (CVN-76), and includes not only ship structure but also furnishings, electronic instrumentation, utility piping and ductwork, fire extinguishers and other machinery, all represented in considerable detail. The system is written in VRML (Virtual Reality Modeling Language), a scriptable, text-based 3D modeling language that can be viewed on a variety of hardware platforms and software environments and that has been widely used for industrial and scientific applications by numerous academic and US government agencies.

3.1. System Architecture

Over the past several years we have developed speech interfaces to an increasing number of US Defense Department applications written in the Java language. Java provides a high-level and easy-to-use software interface called JSAPI (Java Speech API) that is supported by a number of speech recognition systems, including ViaVoice. Since VRML supports Java as a scripting language, it thus became possible to integrate our Java-based speech recognizer directly into a VRML program. The close syntactic parallels between the C language and Java also made it a fairly simple matter to translate MSFT’s route finder from C to Java. As a result, the entire ISFS system can run as a single VRML program without the necessity of inter-process socket communications; otherwise, the system architecture is identical to MSFT’s (Figure 1). It currently runs on a Dell 2.0GHz Xeon computer
running Windows 2000, in a standalone VRML viewer built using the Cortona VRML Client SDK from ParallelGraphics Inc.

3.2. 3D Model

The 3D model of the CVN76 Island House Levels 08 & 09 was provided to NRL by the Newport News, Virginia facility of Northrop Grumman. The model was originally developed in the FTL file format on a proprietary 3D modeling software called Vivid, and was converted to VRML for delivery to NRL. Initially the model was delivered as a single 2GB VRML file which (on our 512MB machine) was too large either to run in a VRML viewer or to edit in our model development software, Autodesk’s 3D Studio VIZ 3i. Subsequently the model was redelivered broken down into nine smaller files each containing a different layer of the original. These VRML files were now of a suitable size to import into 3D Studio, edit in its native format, and re-export as new VRML files acceptable both to the VRML viewer and to several postproduction development tools for data and geometry complexity (polygon) reduction.

![Image of a 3D model](image)

Fig. 3: View forward from the 09 Level port weather deck, virtual *Reagan*

From the standpoint of creating a recognizable and navigable space containing a fairly realistic clutter level of furnishings and equipment, the essential files for our purposes were Structure (floors, walls, ceilings, doors and hatchways), followed by Hull Outfit (chairs and other furniture, stairs, railings, ladders, hatchway doors, electrical equipment mounts, binoculars) and finally Electrical Equipment (computer monitors and keyboards, other device consoles, lights, electronics cabinets). While initially the Structure file was one of the largest (despite modeling features like floors and walls that should only require relatively few simple polygons), we discovered experimentally that by converting the VRML file to AutoCAD DXF format and then re-exporting the file as VRML we were able to reduce its size by a factor of 40, from 322MB to only 8MB, with no loss of detail and only a few minor disruptions in geometry. (VRML files generated by 3D modeling programs have a reputation for being bloated in size, and apparently this was one such case.) Data reduction (redundancy elimination) on the remaining files typically decreased their size by a factor of five or so. Since the Electrical Equipment file was by far the largest we were using, it was additionally submitted to 50% polygon reduction without showing any significant deterioration in visual appearance.

After postproduction we were able to load the Structure and Hull Outfit files for both levels and the Electrical Equipment for one level (09) without exceeding the 320MB or so of RAM available to the VRML viewer, thus keeping the computer’s total memory usage within its 512MB RAM limits and
avoiding the virtual memory paging that otherwise would interrupt the walkthrough animations with long pauses and leaps ahead in time. Since that particular computer can be upgraded to as much as 2GB RAM, more if not all the Island House geometry could potentially be loaded, although it is uncertain whether the resulting animation frame rate would remain at the current minimally acceptable level of about five frames per second.

While detailed in geometric appearance, the modeled objects are not realistically colored but rather are color-coded by layer (Structure: gray, Hull Outfit: cyan, Electrical Equipment: magenta, etc.) making for a somewhat surreal visual environment (Figure 4). While it would be a simple matter to change the color for an entire layer in the model editor, the absence of object types makes it impossible to, say, select all fire extinguishers to assign a particular color, so colorization would instead have to be done laboriously on an object-by-object basis. Similarly, attempting to optimize the model by using the editor to delete individual bits of geometry considered non-essential to the VE display would be prohibitively time-consuming and would only end up “nickel-and-diming” the overall polygon count by a few tenths of a percent at most.

![Fly Control, virtual Reagan](image)

The original model is lit by three DirectionalLights, which illuminate all geometry of the same orientation equally throughout the world. While inexpensive for the VE browser to render, such lights produce an artificial “glare” that is not well suited for realistic interior scenes but is entirely consistent with the somewhat otherworldly color scheme that currently exists. We also added a Background node providing sky, sea and horizon since otherwise the ship would appear to be floating in the blackness of space, and that addition did much to improve the believability of the scene and ameliorate any perceived harshness in the lighting and colors.

While the most familiar VRML viewers are Web browser plug-ins, that runtime environment imposes security limitations on what system operations the embedded Java is permitted to perform. In our case, JSAPI was unable to establish a connection with the speech engine because browser security was enforcing restrictions on the opening of arbitrary system resources. (This is understandable since one would certainly not like to be browsing a remote VRML file on the Web that has the ability to open a microphone connected to your machine and start listening in on your office conversations.) While it is possible for Java to interactively request security override permissions from the user, different vendors and releases of Java handle this in substantially different ways, including the older version of Java (Microsoft JVM 1.1) that the Cortona VRML viewer uses. We finally decided that the simplest and most manageable solution would be to run the VRML program in a stand-alone viewer.
built using the Cortona SDK, thus bypassing browser security issues completely. This approach had the further advantage of allowing us to customize the interface to the VRML viewer itself if we so desire.

3.3. Speech and Graphical Interfaces

Unlike MSFT, ISFS provides not only a speech interface but a functionally equivalent graphical user interface as well (Figure 5). While this may seem to contradict our earlier-stated objections to the use of GUIs in VEs, ISFS is currently a strictly non-immersive or “desktop VE” application occupying a window on the computer screen, with the mouse and keyboard available to either navigate the VE, operate the VE viewer’s controls, or use the separate GUI window as an alternative to speech interaction. In particular, the GUI provides a fallback in case the speech recognizer is either unavailable or having trouble accurately transcribing the user’s utterances. It is built using the older Java AWT (Abstract Windowing Toolkit) rather than the newer Swing Set toolkit since Microsoft JVM 1.1 is a pre-Swing Java release. It provides for instant transport to any compartment in the model, escort between any two compartments, the ability to show or hide a trail of arrows delineating the path from an origin to a destination, and identification of the current compartment (including level, compartment number and name) and direction of view (forward, aft, port or starboard). A button is also provided to enable and disable speech recognition so the operator can carry on side conversations.

Fig. 5: Interactive Ship Familiarization System graphical user interface

The Java Speech API accepts rule-based speech input grammars written in a notation called JSGF (Java Speech Grammar Format) that is remarkably similar to SRCL, including the ability to specify parse tag annotations. Spoken commands and queries generally correspond one-to-one to the controls in the GUI and update them when executed, e.g. the query Where am I? not only elicits a synthetic speech response identifying the current space (“This is the Pilot House”) but also updates the value of the “Here” (current space) text field in the graphical interface. The only difference from the GUI is that the speech interface provides separate queries for current level (Which level am I on?) and compartment number (What’s the number of this compartment?), whereas in the GUI the compartment number (which begins with the level) is always displayed alongside the name so as to identify it unambiguously, e.g. “08-162-1-L Officer WC” and “09-163-1-L Officer WC”. The speech synthesizer pronounces compartment numbers in US Navy fashion (“oh eight tack one sixty two tack one tack L”) and the speech recognizer is prepared to recognize numbers read in the same way. Finally, the command Stop listening puts the speech recognizer into “sleep” mode in which it ignores all further microphone inputs other than the command Resume listening, which restores it to full command recognition mode.

An ambiguous name reference produces a tag structure containing an embedded list of alternative compartment number strings:

Where is the Officer WC?  show from here to {08-162-1-L 09-163-1-L}
Escort me from the Bath to this compartment.  escort from {08-168-3-L 09-167-1-L} to here

If the ambiguous reference is the destination, the command interpreter chooses the one on the same level as the origin as being the most likely intended goal; if the origin, it chooses the one on the level
the user is currently on. In either case it announces “There is more than one <name>, choosing the one on the [current, same] level.” Since these heuristics are not guaranteed to yield what was actually intended, a better approach might be to engage the user in a dialogue to resolve any ambiguities. To avoid any confusion to begin with the user can also preface the name with its level, e.g. How do I get to the 09 Level Officer WC? (Also note that Where is <compartment>? is interpreted in this implementation as a request to be shown a path there, unlike MSFT which treated it as a request to be told the compartment number.)

Since the route finder only associates one key waypoint with each compartment, referencing the exterior weather decks becomes problematic because they wrap around the entire Island House and we would like a command like How do I get to the weather deck? to find a path to the nearest segment of the deck. For that reason we retain the approach taken in MSFT by treating the port, forward and starboard weather decks on each level as different “compartments” and requiring that the user reference them specifically (Take me to the forward weather deck), subject to the same level disambiguation routine just described.

3.4. Automated Wayfinding

The route finder in ISFS is a direct port to Java of the C language version used in MSFT. It loads a waypoint connectivity file that defines the adjacencies of key coordinate locations (waypoints) in the model, such as doorways, corners (turn points) and the centers of compartments. When a route is requested, a separate Java resource file is first consulted to map origin and destination compartment numbers to their key waypoints, and then a best-first search finds the most cost-efficient route between the two points. A route can have a high cost either because of its overall length, or because it includes one or more waypoints that have been defined as being less preferable to use, e.g. a hatchway that requires stooping and crawling through, or doorways that take one from inside the ship to outside and vice versa. Once a route has been computed, the system either draws a chain of arrows along the path for the user to navigate (GUI command Show Path or verbal command How do I get to...), or escorts the user along the path at a walking pace (GUI command Escort or verbal command Take me to...).

3.5. Escort Animation

The escort animation code takes the path (sequence of waypoint coordinates) generated by the route finder and converts it to a VRML PositionInterpolator node. As a clock signal is routed to the node, a continuous stream of interpolated coordinates is generated and routed to the position field of a dedicated Viewpoint node to which the user’s viewpoint has been temporarily bound, generating the animation. The clock is set and the interpolation computed in such a way that each clock tick results in an equal coordinate displacement, producing a steady transition at a walking pace. The route path is also used to generate an OrientationInterpolator node that is routed to the viewpoint’s orientation field to control the viewpoint’s horizontal (left-right) viewing angle. Since the orientation at each waypoint of the path is in the direction of the next waypoint, the OrientationInterpolator produces a continuous stream of interpolated rotations as if the user’s head is gradually turning in the direction s/he is about to move, rather than always looking straight ahead even at turns. The view remains level unless the user is ascending or descending, in which case the orientation is multiplied by a ±22.5° vertical vector causing the viewpoint to momentarily peer up or down and then level out again as the ascent/descent is completed. (While many people might just look straight ahead when using stairs or a ladder, the vertical orientation dipping gives the user a preview of the space about to be entered so as to enhance their spatial awareness of the transition.)

If the path’s origin waypoint corresponds to the same space the user is currently in (e.g. Take me from here to...), the user’s orientation is first rotated toward that waypoint. If the user is located more than a half meter from the waypoint s/he is then walked to its coordinates, at which point the path traversal itself begins. One problem with this approach is that even if the user’s current location is closer to a subsequent node on the path than to its origin, the user is still escorted to the origin and then has to
“double back” past their original position. The straight-line walk to the path origin is also incapable of navigating around obstacles like furniture or equipment, and in an irregularly shaped space might even go through an intervening wall. The solution to these problems would require predefining many more waypoints per space and modifying the route finder to associate multiple waypoints with each one, finding a path that originates from the closest such waypoint to the user.

If the user requests to be escorted from some other space to a destination, s/he is first teleported to the origin waypoint in that space (facing in the same direction as originally) and after a one second pause – intended to provide time to adjust to the “shock” of teleportation – is rotated in the direction of the path and the escort is begun. It might be argued that the user’s orientation after the teleportation (or at the end of an escort, for that matter) should instead be in some pre-determined direction that provides a “good” view of the space, since under the current approach the user might arrive simply facing a blank wall. This was accomplished in MSFT by assigning each waypoint a predefined orientation, and that approach could also be adopted here. The advantage to the current approach, however, is that if the user is aware of which direction they are facing before the teleportation, they can take advantage of that information afterwards without having to make an additional orientation query (Which way am I facing now?).

When an escort has been completed, the system signals the arrival with synthesized speech (“You have arrived at the destination”). MSFT provided an additional visual cue (a large floating red ball) indicating the end of the path, but ISFS relies just on the speech cue and cessation of motion (and/or end of arrow trail, see next section) to signal arrival at the destination. Finally, at any time during an escort the user can halt the walkthrough (e.g. Stop the escort) to pursue independent navigation or to request a subsequent escort or path display.

![Fig. 6: Arrows show path computed by route finder](image)

3.6. Path Display

The user can request to be shown a path from one compartment to another, and then navigate the path manually using the computer’s mouse and/or arrow keys (Figure 6). The path is shown as a sequence of bright yellow 3D arrows floating at about chest level and spaced 1.5 meters apart, each arrow pointing directly toward the one in front of it. (Since from the browser’s viewpoint the arrows are virtual physical objects just like any other, they are grouped in a Collision node with collision disabled so they will not block the user from navigating the path if collision detection is turned on.) As with virtual escort, if the origin waypoint of the path does not correspond to the space where the
user is currently located, the user is first teleported to that waypoint. Instead of rotating the user’s viewpoint toward the first arrow of the path, however, synthesized speech instead announces “Follow the arrows <direction>” where <direction> is “to your left/right”, “in front of you” or “behind you”. This gives control over any initial reorientation to the user, who in requesting a marked route most likely intends to navigate it manually rather than be given an automatic escort. It should also be noted that in a large irregularly shaped space the initial arrow of the path might still not be visible even after a viewpoint rotation, in which case the user would have to do some navigation just to locate the start of the path. We experimented with adding arrows from the user’s current position to the path origin, but this created unnecessarily confusing paths in the case of the “doubling back” routes described earlier.

If the user subsequently issues an escort command, s/he will be led along exactly the same sequence of positions as the arrow path, although during turns the arrows may momentarily go out of sight because the viewpoint orientation is being gradually interpolated away from the current straight-line path segment toward the direction of the next path segment. The user can also request at any time that the arrows be hidden or redisplayed. In the current implementation the arrow path remains visible even after the user has navigated or been escorted to the destination, in case they wish to retrace their steps.

3.7. Teleportation

Rather than be escorted at a walking pace to a new location, the user can request to be transported there instantly using commands such as Teleport (beam, transport) me to Fly Control. This is identical to the instant transport that occurs at the beginning of an escort or path display that originates at a location other than the user’s current location.

3.8. Viewpoint Queries

At any time (including during an escort) the user can query the name of the compartment they are currently in, the ship level they are on, or the direction (forward, aft, port, starboard) they are currently facing. Queries can be posed either as Wh-questions (What compartment is this? Which way am I facing? Which direction is starboard?) or yes/no questions (Is this the Pilot House? Am I facing aft? Is starboard to my right?) The system determines the current compartment from the viewpoint coordinates by consulting a separately defined compartment perimeter database, just as in MSFT. The ship level is determined from the viewpoint’s vertical coordinate, and viewing direction from the viewpoint’s orientation field. It should be noted that the current compartment and direction fields of the GUI do not automatically update as the user navigates or is escorted around, but only when explicitly queried, keeping the semantics of the GUI aligned with the semantics of the speech interface which similarly only provides such information when asked.

3.9. Database Queries

The user can query the compartment name, number and ship level for any particular space either by naming it (What’s the number of the Pilot House?), giving its number (What’s the name of compartment oh eight tack one sixty eight tack three tack L?), using an anaphoric reference (How do I get there? What is that compartment’s number? What level is it on?), or a reference to the current space (What’s the number of this compartment?). Most such queries can also be couched as yes/no questions, e.g. Is the Pilot House on level 08? If the name reference is ambiguous, the system chooses the one on the user’s current level and responds accordingly:

What’s the number of the Passage? The one on this level is number 08-168-1-L.
What level is the Officer WC on? There is one on this level, 09.
Is the Bath on Level 08? There is one on that level.
4. Future Considerations

4.1. Interaction Technique Compatibility

Having been involved for many years with the integration of spoken language interfaces into existing systems already having graphically-oriented interfaces (GUIs and VEs), we have regularly had to deal with the issue of interaction technique compatibility. The GUI and VE interface both rely primarily on the technique of direct manipulation, in which the user causes things to happen through direct personal action typically using a mouse or other manual pointing/triggering device. Spoken language, however, is a communicative interaction mode in which the user addresses commands or queries to an implicit helper agent, who then performs the action on the user’s behalf. Conventional GUIs are to some extent already a hybrid of these two techniques, since menu choices (though selected using direct manipulation) are typically labeled by words or short phrases and can often be thought of as “commands to the computer” rather than direct personal actions. As human-computer system interfaces become increasingly multimodal, care must be taken that the two interaction techniques integrate in an intuitively natural manner. Such research results would also benefit other technology areas such as Augmented Reality (superimposition of information systems on actual rather than virtual reality), collaborative systems, smart house and vehicle technology, etc.

4.2. Object and Information Representation

As we have seen in these two projects, adding an information system to a VE required that the VE database (3D model, behaviors and sensors) be augmented by additional databases representing abstract entities such as compartments, their perimeter coordinates, adjacency information, and symbolic references (names, numbers), all of which are irrelevant to the VE viewing experience and thus not included in the 3D model. For example, a CAD-derived 3D model need contain no representation of a “compartment” at all since that is just the empty space enclosed by the visible geometry representing the walls (which might not be modeled as discrete entities either, but just as an undifferentiated mass of polygons). Handcrafting these additional databases is labor-intensive, error-prone and unlikely to scale up well to larger and more complex projects. There is thus a need for additional model building tools that can model the virtual world as a collection of objects, each of which has both a visual representation (coordinates, geometry, and appearance) accessible by the VE system, as well as a symbolic or conceptual representation (object type, name, and relationship to other objects) accessible by the information system.

As a case in point, while VRML is an object-oriented representation in which shapes can be hierarchically grouped into named nodes representing individual objects, those internal names may not be adequate to automatically identify entities for reference by a symbol-oriented interface. For example one of the deck chairs in Fly Control is represented by a VRML Transform node named _76_5033_1331, but without a concordance between such identifiers and an equipment list it would be impossible to incorporate referential capability such as that chair, the nearest fire extinguisher, etc. into the interface.
References


Escape Route Optimization through Evacuation Analysis of a Recent Ro-Pax Design

Ásmund Hellesøy, Flensburger Schiffbau Gmbh, Flensburg/Germany, hellesoy@fsg-ship.de

Abstract

IMO requires an evacuation analysis to be performed for vessels carrying passengers. New guidelines in force from 1st July 2002 open for the use of computer based evacuation simulation. This paper presents one of the available evacuation simulation software and also escape route optimization using this tool for a recent Ro-Pax design developed by Flensburger Schiffbau Gesellschaft Gmbh. The paper is focused more on the practical use in optimization than on the background theory of the program. One of the important reasons for conducting evacuation analysis is to identify congestion points and to take action to solve possible problems as appropriate. Based on the discussion in this paper it is concluded that care has to be used when conducting analysis according to the new IMO guidelines. Especially IMO’s use of reaction times might hide possible significant congestion problems. The simulation software, the optimization approach and the results of an escape route improvement is presented in this paper.

1. Nomenclature and Definitions

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>Circ.</td>
<td>Circular</td>
</tr>
<tr>
<td>FSG</td>
<td>Flensburger Schiffbau Gesellschaft mbH &amp; Co. KG</td>
</tr>
<tr>
<td>GA</td>
<td>General Arrangement Plan</td>
</tr>
<tr>
<td>GL</td>
<td>Germanischer Lloyd</td>
</tr>
<tr>
<td>HSC</td>
<td>High Speed Passenger Craft</td>
</tr>
<tr>
<td>IMO</td>
<td>International Maritime Organization</td>
</tr>
<tr>
<td>MFZ</td>
<td>Main Fire Zone</td>
</tr>
<tr>
<td>min</td>
<td>Minutes</td>
</tr>
<tr>
<td>MSC</td>
<td>Maritime Safety Committee</td>
</tr>
<tr>
<td>MVZ</td>
<td>Main Vertical Zone</td>
</tr>
<tr>
<td>PS</td>
<td>Port Side</td>
</tr>
<tr>
<td>SB</td>
<td>Starboard Side</td>
</tr>
<tr>
<td>sec</td>
<td>Seconds</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>TraffGo</td>
<td>TraffGo GmbH</td>
</tr>
</tbody>
</table>

2. International attention to evacuation analysis

Several tragic disasters at sea have put international attention to the matter of evacuation analysis. The HSC Code was the first IMO publication to request an evacuation analysis to be performed. Later a regulation considering ro-ro passenger ships came into force, and all such vessels built after July 1st 1999 are required to undertake evacuation analysis at the early stage of design. The MSC, at its 75th session in May 2002 adopted MSC/Circ.1033 which for the first time opens for computer based advanced evacuation simulation. MSC/Circ.1033 (referred to as the guidelines) lets the designer choose from either simplified or advanced evacuation analysis. Evacuation software complying with the new IMO guidelines for advanced analysis are currently available from several different suppliers.

3. Short theoretical description of an evacuation simulation software

FSG is using the evacuation simulation tool AENEAS derived from the cooperation between TraffGo GmbH and GL. The background theory and approach of the various available software programs differs. In the following a short description of the AENEAS software is provided.
3.1. AENEAS

The vessel’s GA is imported from the users CAD system to the AENEAS Editor. Walls, doors, stairways, escape routes, and locations of passengers are automatically identified by the editor, and manual adjustments to the imported layout can be made if necessary.

The approach used in AENEAS is the so called microscopic or advanced simulation model. The decisions and movement of every single person is simulated. A multi-agent-model in discrete time and space is used. Therefore the floor-plan of the investigated ship is divided into square cells of a size of 0,4 by 0,4 meters as represented by the grid on Fig.1. Each square represents approximately the space occupied by one person when standing in a queue. The direction in which a person has to move in order to reach her goal is contained in the cells. A person moves along the escape-route by the given cell-information and then jumps from one cell to the next one, avoiding obstacles and other persons.

Fig.1: Part of the GA for a Ro-Pax vessel being edited in the AENEAS Editor, (TraffGo)

Fig.2: Room Geometry, (TraffGo)  Fig.3: Grid, (TraffGo)  Fig.4: Potential, (TraffGo)

Fig.2. is showing a cabin and a small corridor as a simplified geometry example. The geometry is transformed into cell information as seen in Fig.3. The model is considering the five cell-types; walls, obstacles, doors, stairs and free cells. Persons are represented with dots traveling on the grid of cells.
Potentials for orientation spread out from the appointed exits according to the given escape-routes as indicated in Fig.4. The gradient of the blue color marks the potential value and thereby the resulting direction of movement.

Fig.5: AENEAS Demographics, C: TrafGo

Fig.6: Multiple Escape Routes, C: TrafGo

Variation of personal abilities is taken into account by assigning every person a set of parameters influencing her behavior. The list of factors affecting the movement of a person is nearly endless. However, from a physical point of view the movement of a person is only characterized by her speed and direction.

Fig.7: AENEAS Simulation, C: TrafGo

Fig.8: Distr. of multiple simulations, C: TrafGo

The number of parameters characterizing the abilities of a person in AENEAS was reduced to the six parameters shown in Fig.5. All parameters are assigned by normal distributions with cutoffs for minimum and maximum values. The parameters are: Speed (the speed of the person in cells/s), Patience (maximum waiting time until a person seeks an alternative route), Look (maximum number of cells that a person is looking to decide the direction of movement), Reaction (the reaction time of the person), Dawdle (the probability that a person stands still for the rest of the time-step) and Sway (the probability that a person diverge from her direction of movement). Various escape routes can be assigned as demonstrated in Fig.6. Along these routes, the editor automatically spreads the potentials for directional information. An egress route can be seen as a self-contained layer, by which the person following this route orientates, but at the same time interacting with persons following alternative escape routes.

The evacuation simulation generates a large amount of data. This comes from the many simulation runs which are necessary for a statistical analysis and because the data of every single person is collected. The distribution of evacuation times shows you how much the result spreads and how it is distributed. An example is provided in Fig.8.
4. Short description of the vessel and the simulation model

The escape routes of a Ro-Pax ferry carrying 1000 persons have been investigated. The ferry will be classified for 966 passengers and 34 crew. An extract of the GA is enclosed in Appendix 1. The passenger decks are divided in the 2 fires zones MVZ 4 (aft) and MVZ 5 (forward). Fig.9 shows the geometrical model of the passenger decks as they are represented in the simulation software AENEAS. The dots are the persons and the rectangles aft on 2\textsuperscript{nd} House Deck are the assembly station.

![Diagram of passenger decks](image)

Fig.9: The layout of the vessel’s passenger decks as modeled in AENEAS

Individuals on 1\textsuperscript{st} House Deck climb one deck and exit to open air on 2\textsuperscript{nd} House Deck, and proceed aft to the assembly station. Persons on 2\textsuperscript{nd} House Deck stay on this deck and travel as far as possible the shortest route on open-air decks aft to the muster station. Passengers on 3\textsuperscript{rd} House Deck forward in MVZ 5 follow the stairway to 2\textsuperscript{nd} House Deck, and then move to open-air and aft. Crew on 3\textsuperscript{rd} House Deck in MVZ 4 follow outside stairs down to 2\textsuperscript{nd} House Deck, and crew on 4\textsuperscript{th} House Deck follow the outside route aft and down to the assembly station on 2\textsuperscript{nd} House Deck.
5. Escape route optimization and conclusions

Four evacuation scenarios are considered according to IMO/MSC Circ. 1033. The initial distribution of passengers and crew are shown in Table 1. The four scenarios can shortly be described as follows:

- **CASE 1**: Night scenario where all cabins are occupied with full berthing capacity, and the remaining persons are located in the public spaces.
- **CASE 2**: Day scenario where public spaces are occupied to 75% of maximum capacity, and the remaining persons are located in the cabins.
- **CASE 3**: Similar to CASE 1 but the stairway capacity is reduced by 50% in the most critical fire zone.
- **CASE 4**: Similar to CASE 2 but the stairway capacity is reduced by 50% in the most critical fire zone.

<table>
<thead>
<tr>
<th>CASE 1 - Night Case</th>
<th>CASE 2 - Day Case</th>
<th>CASE 3 - Night Case</th>
<th>CASE 4 - Day Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passengers in cabins</td>
<td>241</td>
<td>Passengers in cabins</td>
<td>126</td>
</tr>
<tr>
<td>Passengers in public spaces</td>
<td>725</td>
<td>Passengers in public spaces</td>
<td>590</td>
</tr>
<tr>
<td>Crew in cabins</td>
<td>5</td>
<td>Crew in cabins</td>
<td>22</td>
</tr>
<tr>
<td>Crew in service spaces</td>
<td>6</td>
<td>Crew in service spaces</td>
<td>6</td>
</tr>
<tr>
<td>Crew causing counter flow</td>
<td>3</td>
<td>Crew causing counter flow</td>
<td>9</td>
</tr>
<tr>
<td>Crew not modeled *</td>
<td>8</td>
<td>Crew not modeled *</td>
<td>3</td>
</tr>
<tr>
<td>Total crew modeled</td>
<td>31</td>
<td>Total crew modeled</td>
<td>31</td>
</tr>
<tr>
<td>Total passengers modeled</td>
<td>966</td>
<td>Total passengers modeled</td>
<td>966</td>
</tr>
<tr>
<td>Total persons modeled</td>
<td>997</td>
<td>Total persons modeled</td>
<td>997</td>
</tr>
</tbody>
</table>

* The crew members not to be modeled are according to the guidelines assumed to be in place at the emergency stations at the time of the alarm.

The simulation software calculates the travel time T. This is the time it takes to gather all persons at the assembly station. The time T includes safety margins to account for simplifications and assumptions stated in the guidelines. Items like rolling of the vessel or static list are not considered in the new guidelines, and this is a major shortcoming. However, IMO has stated that the guidelines are to be used for benchmarking, and development of improved guidelines will follow in the years to come. It is therefore clear that we are not simulating reality. Nevertheless the currently available evacuation simulation software are very good tools when it comes to comparing different layouts of our vessels, and thereby optimizing the escape routes.

The performance standard for ro-ro passenger ships is based on the calculated total evacuation time. 
\[ T + 0.667(E+L) \leq 60 \text{min} \]
\[ E+L = 30 \text{min} \] if results from full scale trials or data from the manufacturer are not available. T is obtained based on multiple simulation runs. The guidelines requires a minimum of 50 repetitions of the simulation runs, but this is in many cases not adequate for obtaining satisfactory results. A total of at least 500 runs should be performed to reach acceptable correlation according to evacuation analysis done at FSG. The value T is taken which is higher than 95% of all
the calculated values, plus a safety margin of 600sec for case 1 & 2 and 200sec for case 3 & 4.

The analysis show that the performance standard is met for all the four cases. The results are shown in Table 2.

Table 2. Total evacuation time for cases 1 to 4 compared to maximum allowable evacuation time.

<table>
<thead>
<tr>
<th>CASE</th>
<th>T_95%</th>
<th>Safety Margin</th>
<th>T</th>
<th>E+L</th>
<th>T + 0,667(E+L)</th>
<th>T + 0,667(E+L)</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1333</td>
<td>600</td>
<td>1933</td>
<td>1800</td>
<td>3133</td>
<td>52</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>1064</td>
<td>600</td>
<td>1664</td>
<td>1800</td>
<td>2864</td>
<td>48</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>1358</td>
<td>200</td>
<td>1558</td>
<td>1800</td>
<td>2758</td>
<td>46</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>1263</td>
<td>200</td>
<td>1463</td>
<td>1800</td>
<td>2663</td>
<td>44</td>
<td>60</td>
</tr>
</tbody>
</table>

We have seen that the performance standard is met, but when taking a closer look at the analysis it is discovered that significant congestions are occurring during the evacuation. The guidelines utilize uniformly distributed reaction time with the minimum, average and maximum reaction times shown in Table 3. The relatively large deviation causes the evacuating persons to be spread out in time so that hardly any significant congestion is occurring for most scenarios as defined in the guidelines. We are seeing the reaction times as given in Table 3 as somewhat unrealistic. When evacuating a large room with several hundred persons we find it likely that a large portion of the individuals choose to proceed to the exits more or less simultaneously. If some persons reacts to the alarm, and starts evacuating, it is likely that the entire group will follow. Persons in cabins probably obtain more spread reaction times simply because of the lack of visual contact between the cabins.

Table 3. The population’s reaction time as given in the guidelines.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (night)</td>
<td>420</td>
<td>600</td>
<td>780</td>
</tr>
<tr>
<td>2 (day)</td>
<td>210</td>
<td>300</td>
<td>390</td>
</tr>
<tr>
<td>3 (night)</td>
<td>420</td>
<td>600</td>
<td>780</td>
</tr>
<tr>
<td>4 (day)</td>
<td>210</td>
<td>300</td>
<td>390</td>
</tr>
</tbody>
</table>

Fig.9: Density plots for case 1 with (left) and without (right) reaction times

When looking for congestion points the reaction times should be set to zero according to the discussion in the previous paragraph. Then an effective escape route improvement could be
completed for this vessel. To illustrate this the person-density plot for case 1 is included in Fig.9. The image to the left shows the original case 1 density plot, and the image to the right shows case 1 but with the reaction time set to zero. This paper is in black and white, but in the original images the colors red, yellow and green are representing high, medium and low person densities respectively. In black and white the darkest gray color is red. Significant congestion occurs on the image to the right. The major problems are seen on the 1st House Deck entering the stairway halls and the stairs, and also on the 2nd House Deck where the persons are exiting to the open deck from MFZ 5.

The first improvement was to increase the stairway capacities of the main stairways in MFZ 4 and 5 by 50%. Figs.10 and 11 are showing the person-density plots before and after the widening of the stairs. The biggest improvement is seen in the forward stairway on 1st House Deck where the areas of dark gray is significantly reduced in the stairway hall. The problem has however moved up one deck where an increase of dark gray can be seen where the persons are exiting to the open-air deck. The logical action will be to widen these doors in the next round. At the aft stairways some improvement can be seen, but the major congestion before entering the stairway hall on SB side remains almost unchanged. The doors on SB will be widened, and the possibility for an additional door on the PS will tested. This will cause some rearrangement of the AC-Room and galley area, but the benefits of symmetrical entry to the aft stairway hall are in this case seen as the major deciding factor.

![Initial case without reaction time](image1.png) ![Wider stairs compared to Fig.10](image2.png)

Figs.12 and 13 shows the situation before and after the incorporation of the suggested improvements. The widening of the doors on 2nd House Deck shows major reduction of congestion. Once the persons have reached the open-air deck there is sufficient space according to the simulation, and they are proceeding aft to the assembly station. Additionally the aft stairway hall and its entrances on 1st House Deck have been changed. Some interior obstacles have been moved or removed and the new entrance on PS has been added. The congestion in the stairway hall has been lowered to a very satisfactory level, but the dark gray area before entering the stairway hall on SB side is still problematic. This is because very few persons have chosen to utilize the new door on PS. The problem was firstly tried to solve with a guiding railing trying to split the flow of persons forcing about 50% of them to use the PS door, but this was not very effective. Therefore a paragraph has been added to the general alarm and evacuation instructions that crew guidance is needed at this point. The importance of the crew and their actions in possible emergency situations has been shown to be of utmost importance. The best known research on this area comes from the aerospace industry showing the significant effects of crew guidance in passenger aircraft evacuation.
The passengers and crew on 1\textsuperscript{st} House Deck in MVZ 4 will be guided by crew so that an approximately equal number of persons enter the stairway hall through the SB and PS entrances. Figs.14 and 15 are showing the effect of this improvement.
Figs. 16 and 17 show the result of even wider stairs in the forward stairways. It results in minor reduction of congestion, but no large effects are seen. Door-widths were adjusted aft on the 2\textsuperscript{nd} House Deck improving the results further.

We have now performed some rounds of escape route optimization obtaining an improved layout of the vessel. At this point it is interesting to calculate the travel times for the improved layout for comparison with the initial GA. When calculating the old and new scenarios according to the guidelines including the reaction times as shown in Table 3, the improved layout only cases 2.5\% shorter travel time. Excluding the reaction time, on the other hand, shows the significant improvement of 26\% for the optimized escape routes compared to the initial layout. This information tells us that we have to be very careful when presenting evacuation analysis results to ship-owners, classification societies and national authorities. In this case we could possibly get the situation where a ship-owner did not want to incorporate the improvements only to gain 2.5\% on the travel time calculated according to the guidelines. However, when talking about 26\% shorter travel time, the considerations might be very different.

**Closing comment**

If the conclusions from this paper can be reproduced and confirmed by other sources, it might be appropriate to reevaluate the reaction times according to IMO/MSC Circ. 1033. Reaction times are undoubtedly needed, but the current values might hide some possible congestion problems as the case is for the vessel investigated here.

**Acknowledgements**

We warmly thank TraffGo GmbH for their cooperation, interest and help in this project.

**References**

1. IMO. *Interim Guidelines for a simplified evacuation analysis on ro-ro passenger ships*. IMO, 1999. MSC/Circ.909
3. IMO. *Interim Guidelines for evacuation analysis for new and existing Passenger Ships*. IMO, 2002. MSC/Circ.1033
Appendix 1: Extract of the General Arrangement for a Ro-Pax-1000 design
Least Cost Optimisation of a Medium Capacity Gas Carrier

Philippe Rigo, ANAST, University of Liege (NSFR), Liege/Belgium, ph.rigo@ulg.ac.be
Catalin Toderan, ANAST, University of Liege, Liege/Belgium, catalin.toderan@ulg.ac.be

Abstract

To be attractive for shipyards, scantling optimisation has to be performed at the preliminary design stage. It is indeed the most relevant period to assess the construction cost, to compare fabrication sequences and, to find the best frame/stiffener spacings and most suitable scantlings to minimize the production costs. The LBR5 package performs such early design least cost optimisation.

The software contains 3 modules. The “Cost Module” to assess the construction cost which is the objective function (least construction cost). The “Constraint Module” performs rational analyses of the considered structure, and the “Opti Module” contains the mathematical optimiser code.

This paper presents the optimisation of a medium capacity LNG ship designed by ALSTOM France (Chantiers de l’Atlantique). In addition to scantling optimisation, sensitivity analysis on frame/stiffener spacings are presented. This industrial optimisation example shows that 5 to 9% of the construction cost of the hull manufacturing can be saved by performing scantling optimisation at the preliminary design stage.

1. Introduction

LBR5 is a rationally based optimisation module that, in the preliminary stage, allows for:
• a 3D analysis of the general behaviour of the structure (usually one cargo hold);
• to explicitly take into account all the relevant limit states of the structure (service limit states and ultimate limit states) thanks to a rational analysis of the structure based on the general solid-mechanics theory;
• an optimisation of the sizing/scantling (profile sizes, dimensions and spacing) of the structure’s constituent elements;
• to include the unitary construction costs and the production sequences in the optimisation process (through a production-oriented cost objective function).

Design variables are the dimensions of the longitudinal and transversal members, plate thickness and spacing between members.

Extensive information on the proposed model is available in the literature: Rigo (2001a and b), Rigo and Fleury (2001), Karr et al. (2002).

2. LBR5, a tool for least cost scantling optimisation

LBR5 is built around three basic modules, respectively, OPTI, CONSTRAINT and COST (Figure 1). The OPTI module contains the mathematical optimisation algorithm to solve non-linear constrained optimisation problems, Fleury (1989), Rigo and Fleury (2001).

The CONSTRAINT module includes (1) technological constraints (or side constraints) that provide the upper and lower bounds of the design variables; (2) geometrical constraints that impose relationships between design variables in order to guarantee a functional, feasible and reliable structure. They are generally based on “good practice” rules to avoid local strength failures (web or flange buckling, stiffener tripping, etc.), or to guarantee welding quality and easy access to the welds; (3) structural constraints that represent limit states in order to avoid yielding, buckling, cracks, etc. and to limit deflection, stress, etc.

In the COST module the objective function is the construction cost that includes labour costs and material cost (Rigo 2001b).
Fig. 1 shows the basic configuration of the LBR5 software with the 3 fundamental modules (COST, CONSTRAINT and OPTI). It describes the general organisation chart of a structure optimisation process and the LBR5 software chart.

With regard to structural constraints, the user must first choose the types of constraints (yielding, buckling, deflection, etc.) then, for each type of constraint, select the method, the code or the rules to use and finally the points/areas/panels where these constraints will be applied.

- **Selection and initialisation** of design variables \((X_i)\)
  and the lower/upper bounds \((X_{\text{min}} \leq X_i \leq X_{\text{max}})\)

**CONSTRAINT MODULE**
- Geometrical constraints \((C(X_i))\) and their sensitivities \(\partial C/\partial X_i\)
- Structural constraints related to the global structure & sensitivity analysis (stress, displacements, …)
  - **Linear Elastic Analysis** *(Rigo 1992)*
    Computation of deflections, unitary forces and stresses \((\sigma, w, \ldots)\)
  - **SENS**
    Sensitivity Analysis \((\partial \sigma/\partial X_i, \ldots)\)
- Structural Constraints on components & their sensitivities \(\partial C/\partial X_i\)
  - Plate Yielding,
  - Stiffeners Buckling…
  - Stiffened Plates Ultimate Strength…
  - Box/Hull Girder

**COST MODULE**
- Objective Function \((F(X_i))\) & the sensitivities \(\partial F/\partial X_i\)

**OPTI MODULE**
- Link with data from the CONSTRAINT and COST modules
- Research of the Optimum \(\rightarrow\) **CONLIN** \(\rightarrow\) New set of design variables
- Updating the Design (scantling) with the new values for the design variables provided by CONLIN

To a new re-analysis

**OPTIMUM SOLUTION**

Fig.1: Chart of the LBR5 model with CONSTRAINT, COST and OPTI modules
LBR5 is also an efficient tool to assess and compare different alternatives. For instance Figure 2 gives the cost and the weight as functions of the web-frame spacing. Other major capability of the method is to quantitatively assess a change of the production technology on the construction cost. For instance, effect of an improved welding procedure (lower unitary welding cost) can be assessed by comparing the least cost optimum scantling obtained with and without the improvement.

Fig.2: Sensitivity analysis: Cost and weight as a function of the web-frame spacing

Figure 3 (left) shows the simple and fast mesh modelling methodology used by the LBR5 software to optimise a fast ferry (right), which may latter be modelled using standard finite elements (for advanced analysis).

Fig.3: LBR5 Mesh model of a fast ferry (Principia Marine)

Principia Marine (France) and University of Liège have furthermore engaged in cooperation to jointly develop their software and integrate them as part of an early design suite (Goubault et al. 2003).

Figure 4 illustrates the use of AVPRO and LBR5 in early design. Although the level of precision of AVPRO is not as fine as those software, the ability of this software to handle all the tasks in one single environment provides this tool with a unique capability for early design. AVPRO is completed with LBR5 with regard to structural design. Further into the design phase the 3DPRO methodology enables FEA calculations to be carried out at much earlier stage than is currently the case.
Fig. 4: Software used in the design loop

The OPTI module is based on the CONLIN code developed by Fleury (1989) using a convex linearization of the constraints and the objective function combined in a dual approach. With this algorithm, large constrained problems with implicit and non-linear constraints can be easily solved. The main difficulty in solving a dual problem is dealing with the non-linear and implicit constraints. In order to avoid a large number of time-consuming re-assessments of these non-linear and implicit functions, Fleury suggests applying convex approximations. At each iteration all the functions (objective function and constraints) are replaced by an approximation called “convex”. In a word, the complex initial optimization problem is decomposed in a sequence of more simple convex optimization problems (obtained through a convex linearisation) that can be easily solved using a dual approach (Figure 5).

In order to consider non-linear implicit constraints \( C(X_i) \), Fleury proposes replacing these constraints with approximated explicit linear constraints by using convex linearisation with mixed variables \( X_k \) and \( 1/X_j \).

\[
C(X_j) = \tilde{C}(X_j) = C(X_j(0)) + \sum_{k} \left[ X_k - X_k(0) \right] \frac{\partial C(X_k(0))}{\partial X_k} \\
- \sum_{j} \left[ 1/X_j - 1/X_j(0) \right] \left( X_j(0) \right)^2 \frac{\partial C(X_j(0))}{\partial X_j} \tag{1}
\]

with \( \frac{\partial C(X_k(0))}{\partial X_k} > 0 \quad (1 \leq k \leq N) \); \( \frac{\partial C(X_i(0))}{\partial X_i} < 0 \quad (1 \leq j \leq N) \), for i=1,N

Fig. 5: The CONLIN model: Convex approximations and Dual approach
3. Least cost optimisation of a medium capacity gas carrier

The present example of least cost optimisation concerns the optimisation of a medium capacity gas carrier (LNG), designed by ALSTOM - Chantiers de l’Atlantique (France).

Due to the international competition between shipyards on such ships, a lot of valuable information will not be mentioned in the present paper. Nevertheless, the authors acknowledge ALSTOM for its courtesy for allowing use of their results. In this paper, data are mainly presented in terms of ratios to avoid publishing sensitive confidential quantitative data.

The present optimisation concerns one of the four tanks (Figure 6). The goal is to define the optimum tank scantlings corresponding to the minimum construction cost. An additional aim is to assess feasible alternative designs (that is, improved general structural layouts).

![Fig.6: General view of a medium capacity gas carrier (ALSTOM, France)](image)

To calibrate the current LBR5’s cost module with the ALSTOM’s unitary costs, the cost of a standard stiffened panel was assessed using the unitary production costs of ALSTOM. These unitary costs relate to:

- plate assembling and welding,
- longitudinal stiffener assembling and welding,
- transverse frame prefabrication,
- transverse frame assembling and welding (for different assembling sequences as the structure is mainly composed of double bottom, double deck and double side plates),
- slots, brackets, etc. (cutting, assembling and welding),
- ….

This assessment was used to define the LBR5 cost parameters: unitary prices of material (C1, ΔC1); unitary working loads for plate assembling (P10, ΔP10), frames and stiffeners (P4, P5, ΔP4, ΔP5, P9, ΔP9).

The ship is classified by Bureau Veritas and its MARS2000 software was used by ALSTOM to define the initial scantlings to be used by LBR5 (reference values). After optimisation, the new scantlings (optimum) are validated using MARS2000 to confirm the feasibility of the new layout and scantlings. This control fully confirms the LBR5 results and the possibility to save about 8% of the tank’s construction cost (cofferdam excluded).

Five loading cases were considered by LBR5 (Figures 9 and 10). They were obtained combining unitary load cases (three of them being presented at Figure 7). The structural mesh model is shown on Figure 8. Based on structure symmetry, only half of the structure is modelled for structure optimisation with the LBR5 model. The maximal sagging and hogging hull girder bending moments (still water level) were valued by ALSTOM through direct calculation (loading manual). The wave bending moments were obtained from classification rulebook.

98
Fig. 7: Unitary Load Cases

Sea Loads (BV’s rules)  Internal pressure in BALLAST (BV’s rules)  Dynamic Internal Pressure in GAS Tank (BV’s rules)

Fig. 8: LBR5’s Mesh Model of a Medium Capacity Gas Carrier

Fig. 9: LBR5’s Load Case 1: Maximum lateral pressure provided by MARS (BV)
The mesh model of the medium capacity gas carrier (LNG) includes:

- 41 stiffened panels with 9 design variables each (some are not considered as variables);
- 4 additional panels to simulate the symmetry axis (boundary conditions);
- 278 design variables (on average 5 to 9 design variables per panel);
- 106 equality constraints between design variables are used, e.g., to impose uniform frame spac-
ing for the deck, bottom and the side ballast tanks.

- 203 geometrical constraints (about 5 to 6 x 41 panels). For instance, longitudinal web heights are limited by such constraints to control the web slenderness.
- 1900 structural constraints (380 per load case):
  - \( \sigma \), frame (web/plate junction – web/flange junction and flange),
  - \( \sigma \), stiffener (web/plate – web/flange and flange) and \( \sigma \), plate,
    which verified that \( \sigma \leq s_1 \sigma_o \) (with \( s_1 \) a partial safety factor and \( \sigma_o \) the yield stress);
  - local plate buckling: \( \delta_{\text{MIN}} \leq \delta \) (with \( \delta_{\text{MIN}} \) the minimum plate thickness to avoid buckling and yielding);
  - ultimate strength of stiffened panel: \( \sigma / \sigma_{\text{ULT}} \leq s_2 \) with \( s_2 \) a partial safety factor.

In addition side constraints are imposed on the design variables \( (X_{\text{MAX}}, X_{\text{MIN}}) \). For instance, the upper limit for the (\( \delta \)) plate thickness is fixed at 25 mm. Other selected limits (side constraints) are:

\[
\begin{align*}
2.00 \text{ m} & \leq \Delta_{\text{Frames}} \leq 4.00 \text{ m} \\
0.50 \text{ m} & \leq \Delta_{\text{Stiffeners}} \leq 1.00 \text{ m} \\
0.10 \text{ m} & \leq h_{\text{web stiffeners}} \leq 0.50 \text{ m} \\
8.0 \text{ mm} & \leq \text{Web-frames thickness} \leq 25.0 \text{ mm}
\end{align*}
\]

..... etc.

**Actual tanks composed of stiffened panels sharing an unique transverse web-frame (H,d), (i.e. double bottom, double deck, side tanks)**

**Approximated modelling approach used by LBR-**

**Tank modelled with 4 stiffened panels having independent frames (h,d).**

**Fig.11: LBR5’s Principle to Model a Double Sided Stiffened Tank**

Figure 11 points out a major difficulty to use LBR5 for double hull ships. Being a principle of the method \( (\text{Rigo 1989}) \) each web-frame is attached to a unique panel and cannot be shared by all the constitutive panels of a double bottom (Figure 11). Thanks to ALSTOM support, a new methodology is under development to face this problem.

**Fig.12:**

**LBR5’s stress distribution at the junction between framed panels.**

Figure 12 explains another LBR5 assumption, that is, the junction between frames coincides with the panel’s node (A or A’) and not to B (or B’). This means that around the frame’ junction, the stresses
are overestimated. To fix this problem, two approaches may be used; (a) to consider in the constraints the stress at point B (B’) instead of A (A’), (b) to consider points A and A’ but to increase the allowable stress by a ratio ($\sigma_b / \sigma_b$) or ($\sigma_a / \sigma_b$).

3.1. How to Minimise the Construction Costs of the LNG Ship.

Tracks to reduce the construction cost of the medium capacity LNG ship are:

- To increase the web-frame spacing:
  - $(N_w - 2)$ web-frames instead of $N_w$ web-frames → Cost saving: 4.85%
  - $(N_w - 3)$ web-frames instead of $N_w$ web-frames → Cost saving: 6.40%

- To increase the stiffener spacing ($\Delta_L$):
  - $1.09 \Delta_L$ instead of $\Delta_L$ → Cost saving: 1.61%
  - $1.15 \Delta_L$ instead of $\Delta_L$ → Cost saving: 2.40%
  - $1.28 \Delta_L$ instead of $\Delta_L$ → Cost saving: 2.97%

NB: straightening cost are not considered.

Note: $N_w$ and $\Delta_L$ refer to the initial design (before optimisation). $N_w$ is the number of web-frames, $\Delta_w$ the frame spacing and $\Delta_L$ the average longitudinal stiffener spacing.

3.2. Steps of the LBR5 Optimisation Process

Aim of the LBR5 optimisation analysis is to provide a least construction cost and feasible scantlings of the 4 tanks. This optimisation is performed within a series of constraints. There are technological (minimum thickness, corrosion, etc.), geometrical (rule based) and structural constraints (rational based → direct calculation).

In principle, LBR5 directly provides the global optimum. In that case, it is not possible to assess the cost saving of each individual parameter like frame spacing, stiffener spacing, plate thickness, duct-keel layout, etc.

To assess these individual cost savings, the present optimisation was split in several sub-optimisations. So, starting from the Alstom’s initial design, step by step, parameters are released and the layout modified (see Tables 1 to 3). Initially, the upper limit of each design variable is fixed at the Alstom’s initial scantling value. Then, the upper limits of a group of design variables are released (typically starting with the frame spacing and stiffener spacing).

Main sub-optimisations are presented in Table 1. They are:
- Least cost optimisation (starting from the initial scantlings provided by ALSTOM, with fixed frame and stiffener spacings),
- Web-frame spacing ($\Delta_w$) is released: $N_w \rightarrow (N_w - 2)$ frames,
- When feasible, the stiffener spacing is released: $1.15 \Delta_L$ and $1.28 \Delta_L$ instead of $\Delta_L$,
- General structural layout is modified,
- Spacing of secondary frames is modified (typically 2 or 3 secondary frames between web-frames are considered, that is, respectively, $\Delta_C = \Delta_w/3$ and $\Delta_w/4$).

Table 2 assesses the cost saving associated with each sub-optimisation and with the global optimisation (cumulated cost saving). It shows clearly that the way to reduce construction cost of the concerned LNG ship is to increase the web-frame spacing ($N_w-3$) and to standardize the stiffener spacing at $1.15 \Delta_L$ (in average). Such changes induce a cost saving of about 8.50% (material and labour costs).
Table 1: Steps of the Optimisation Process (with sub-optimisations)

<table>
<thead>
<tr>
<th>1- ALSTOM</th>
<th>Search for the least cost design (with continuous design variables)</th>
<th>Steps of the optimisation process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimum Type</strong></td>
<td><strong>Number of Web-frames</strong></td>
<td><strong>Secondary Frame Spacing</strong> ($\Delta_c$)</td>
</tr>
<tr>
<td>MARS BV</td>
<td>$N_w$</td>
<td>$\Delta w/3$</td>
</tr>
<tr>
<td><strong>Initial &quot;ALSTOM&quot; layout used as reference point (before optimisation) With discrete design variables.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2- MET8 E00</strong></td>
<td><strong>Least Cost</strong> (*)</td>
<td>$N_w$</td>
</tr>
<tr>
<td><strong>After optimisation of the ASTOM initial design with $\Delta w$, $\Delta_c$, and $\Delta_l$ unchanged.</strong> The design variables become continuous.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>3- MET8 E90</strong></td>
<td><strong>Least Cost</strong></td>
<td>$N_w$</td>
</tr>
<tr>
<td><strong>The stiffener spacings are released (1.15 the initial value)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>4- MET8 B90</strong></td>
<td><strong>Least Cost</strong></td>
<td>$N_w -3$ (*)</td>
</tr>
<tr>
<td><strong>The web-frame spacing is released (upper limit corresponds to 9 frames).</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>5- MET8 F90</strong></td>
<td><strong>Least Cost</strong></td>
<td>$N_w -3$</td>
</tr>
<tr>
<td><strong>As the web-frame spacing becomes larger, one additional secondary frame spacing is added ($\Delta_c = \Delta w/4$ instead of $\Delta w/3$).</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6- MET8 F</strong></td>
<td><strong>Least Cost</strong></td>
<td>$N_w -3$</td>
</tr>
<tr>
<td><strong>The stiffener s' upper limit is increased to 1.25 the initial value</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*) Shows the modified parameter (or variable) between two successive steps

Table 2: Cost Saving at Each Step of the Optimisation Process

<table>
<thead>
<tr>
<th>Search for the least cost design (with continuous design variables)</th>
<th>Spacings</th>
<th>Duct keel bulkhead, Plate thickness</th>
<th>Least cost</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONFIGURATIONS</strong></td>
<td><strong>Optimum Type</strong></td>
<td><strong>Number of Web-frames</strong></td>
<td><strong>Secondary Frame Spacing</strong> ($\Delta_c$)</td>
<td><strong>Stiffeners</strong> ($\delta_l$)</td>
</tr>
<tr>
<td>1- ALSTOM</td>
<td>MARS BV</td>
<td>$N_w$</td>
<td>$\Delta w/3$</td>
<td>$\Delta_l$ (Alstom)</td>
</tr>
<tr>
<td>2- MET8 E00</td>
<td><strong>Least Cost</strong> (*)</td>
<td>$N_w$</td>
<td>$\Delta w/3$</td>
<td>$\Delta_l$ (Alstom)</td>
</tr>
<tr>
<td>3- MET8 E90</td>
<td><strong>Least Cost</strong></td>
<td>$N_w$</td>
<td>$\Delta w/3$</td>
<td>1.15 $\Delta_l$ (Alstom)</td>
</tr>
<tr>
<td>4- MET8 B90</td>
<td><strong>Least Cost</strong></td>
<td>$N_w -3$</td>
<td>$\Delta w/3$</td>
<td>1.15 $\Delta_l$ (Alstom)</td>
</tr>
<tr>
<td>5- MET8 F90</td>
<td><strong>Least Cost</strong></td>
<td>$N_w -3$</td>
<td>$\Delta w/4$</td>
<td>1.15 $\Delta_l$ (Alstom)</td>
</tr>
<tr>
<td>6- MET8 F</td>
<td><strong>Least Cost</strong></td>
<td>$N_w -3$</td>
<td>$\Delta w/4$</td>
<td>1.28 $\Delta_l$ (Alstom)</td>
</tr>
</tbody>
</table>

(*) Stiffener spacing too large $\Rightarrow$ cost savings of 0.5% but increased straightening work $\Rightarrow$ not efficient !!

(1 Variation induced by the changes occurred between two configurations.

Unfortunately, the global optimum (MET8-F90) is characterised by an increase of the weight of 3.4%. To avoid this negative effect, ALSTOM proposed some layout improvement to keep the hull weight almost unchanged. One alternative to avoid an increase of hull weight is to select configuration MET-12b presented in Table 3. MET-12 is characterized by a new structural layout and ($N_w=2$) web-frames (instead of $N_w$, see Table 3). Cost saving still reaches 6.9 % with, in addition, a small weight reduction (99.68% on the initial weight).
Table 3: Additional optimum solutions including weight constraint

<table>
<thead>
<tr>
<th>CONFIGURATIONS</th>
<th>Optimum Type</th>
<th>Number of Web-frames</th>
<th>Second. Frame (Δc)</th>
<th>Stiffeners (Δl)</th>
<th>Duct keel bulkhead. Plate Thickness (mm)</th>
<th>LEAST COST</th>
<th>WEIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Shown change(s) between 2 successive steps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALSTOM</td>
<td>MARS BV</td>
<td>Nw</td>
<td>Δw/3</td>
<td>Δc</td>
<td>(Alstom)</td>
<td>100%</td>
<td>100.00%</td>
</tr>
<tr>
<td></td>
<td>Least Cost</td>
<td>Nw -2</td>
<td>Δw/3</td>
<td>Δc</td>
<td>(Alstom)</td>
<td>105%</td>
<td>98.34%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MET8 E-78</td>
<td>Least Cost</td>
<td>Nw</td>
<td>Δw/3</td>
<td>Δc</td>
<td>(Alstom)</td>
<td>122%</td>
<td>100.21%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MET8 C-78</td>
<td>Least Cost</td>
<td>Nw -2</td>
<td>Δw/3</td>
<td>Δc</td>
<td>(Alstom)</td>
<td>88% (*)</td>
<td>99.68%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MET 12 (*)</td>
<td>Continuous</td>
<td>Nw -2</td>
<td>Δw/3</td>
<td>Δc</td>
<td>(Alstom)</td>
<td>88% (*)</td>
<td>100.88%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MET 12.b (*)</td>
<td>Discrete</td>
<td>Nw -2</td>
<td>Δw/3</td>
<td>Δc</td>
<td>(Alstom)</td>
<td>88% (*)</td>
<td>100.88%</td>
</tr>
</tbody>
</table>

(1) Variation induced by the changes occured between two configurations.

Figures 14 and 15 compare, respectively, the COST and WEIGHT of the different configurations. Sub-optimisations and optimum solutions are marked.

Based on the LBR5’s proposals, ALSTOM modified their initial design and structure layout to define a revised solution that was used for the final design stage.

Expected savings predicted by LBR5 are confirmed through ALSTOM’s additional workload assessment.

Figures 16-18 show typical results provided by LBR5 at the end of the optimisation procedure. These results correspond to the 3D rational analysis of a cargo hold.

4. Conclusions

This paper presents an industrial example of scantling optimisation performed using the LBR5 software. The target is to ease and improve the preliminary design stage allowing least cost optimisation. As the general structural layout is defined during the earliest phases, it is easy to understand why a least cost optimisation tool is attractive, especially one designed for use at the preliminary stage.

Optimum analysis of a medium capacity LNG ship is presented as application of the LBR5 least cost optimisation model. Alternatives like modified web-frame spacing, larger stiffener spacing, and improved layout have been assessed. Starting from the original shipyard’s structural layout, LBR5’s analysis has provided an effective cost saving of 8.5% for the hull tank construction. Main sensitive parameters are frame spacing and stiffener spacing. Using the LBR5 package, optimum scantlings for different configurations were easily compared based on their cost and weight, taking into consideration the shipyard standards (stiffener profiles, shop efficiency and availability, etc.).
Fig. 14: Sensitivity analysis of the stiffener spacing on the construction cost of a medium capacity LNG carrier (one hull tank)
Fig. 15: Sensitivity analysis of the stiffener spacing on the hull weight of a medium capacity LNG carrier (one hull tank)
5. Acknowledgments

The authors thank ALSTOM, Chantiers de l’Atlantique (St. Nazaire, France) and particularly the Steel Hull Department for their support to achieve this research and their assistance and authorisation to publish the paper. The DG TRE Ministry of the Walloon Region of Belgium is also acknowledged for their support (Convention n°215062, First – Spin off).

References


GOUBAULT Ph.; BESNARD N.; RIGO Ph. (2003), AVPRO-LBR.5, An Initial Design Software Suite with Scantling Optimization Capabilities, IMDC’ 03, Athens, Greece.


ODIGO: A Crowd Movement Simulation Tool for Passenger Vessels

Jean-Yves Pradillon, Principia Marine, Nantes/France, j-yves.pradillon@principia.fr

Abstract

Cruise ships are becoming larger and larger and boarding more and more passengers. Both shipyard and ship owner have to deal with the crowd movements onboard to optimise the ship for emergency procedures and for operation as well.

Ship evacuation, in case of disaster, is of a main concern today. This is particularly true for Passenger Vessels, for which it is critical, but will probably also become a requirement for other ship types. The owners and regulations request the yards to be aware of the evacuation process and to take it into account within the design stage. Ship owners have also to take into account the comfort of passengers and, as far as possible, to avoid them queuing during their leisure time. Disembarkation, dinner and end of shows are typical periods when many passengers compete for the access to public areas. A simulation tool allows the designer to optimise the paths identifying bottlenecks, the influence of obstacles and the adequacy of space design together with the role of the crew.

The paper will introduce a quick review of the present international regulations together with a review of various IT techniques that can be used for such a simulation. The implemented method in the ODIGO is based on the multi-agent approach. Each agent is provided with a comprehensive autonomy allowing choice of the direction and speed to reach goals which are updated on a time step basis upon external conditions such as obstacles, other agents vicinity etc. A set of human behaviours is included in the model leading to real rational and non-rational decisions as far as each agent has its own perception of the real world.

The paper concludes with the foreseen test plan and perspectives such as simulation of red code situations on board military vessels.

1. Introduction

On one hand, the demand for passenger shipping has increased in the last years. Both leisure cruise liners and ferries (high speed craft and regular Ro-Ro) are concerned. On the other hand, safety on board is a main concern and becomes a strong requirement. Under market and regulations pressure, the ship owner requires today a better assessment of the capabilities to evacuate the ship and to deal with crowded public spaces. Some rules include a demand for simulating the evacuation process. In these conditions, the designers need simulation tools to check the evacuation procedures and to optimise them as well.

ODIGO (from ancient Greek: I guide) is a French national funded project whose partners are Chantiers de l'Atlantique (shipbuilder for cruise liners and high speed crafts), Bureau Veritas (Classification Society) and Principia Marine together with two subcontractors who also contribute to the project: MFRDC (Consultant) and University of Plymouth. The objective is to provide the yard's designers with a tailor-made tool that can meet their needs and their internal organisation of the design process.

The output from the simulation will focus on two kinds of results:

- Checking the path against geometrical recommendations in the rules,
- Quantitative (evacuation times, people density in critical areas...) and qualitative (bottle necks, misplacement of obstacles...) items.

After an overview of the regulations and the existing methods in other industries or from research works, the paper will introduce the multi-agent approach implemented in the ODIGO project (model, agents description, human factors to be included) and conclude with the perspectives offered by such a development.
2. Regulation overview

We distinguish two different types of regulations:

Prescriptive regulations:
The means by which a certain level of safety can be attained is determined. No choice is left to the operator as to how to attain this level.

Performance-based regulations:
A specified safety goal must be reached, regardless of the technical solutions put into place.

For passenger ship evacuation, two main regulations have to be applied: SOLAS and the HSC Code.

2.2. SOLAS

The most important international conventions dealing with maritime safety is the International Convention for the Safety Of Life At Sea (SOLAS), IMO (1997), which covers a wide range of measures designed to improve the safety of shipping. This convention shall apply to ships on international voyages, entitled to fly the flag of a contracting state.

The SOLAS Convention (chapter I, Regulation 2) defines:
Passenger is every person other than
- the master and the members of the crew; and
- a child under one year of age.

Passenger ship is a ship that carries more than twelve passengers.

In the SOLAS rules, the evacuation process is divided into phases as shown in Fig.1:

![Diagram of SOLAS evacuation process]

Fig.1: Definition of various time names used in SOLAS rules for the evacuation process

In the simulation tool SPECS, the simulation covers "Evacuation time" and possibly "Total evacuation time".

Regulations, which deal with the evacuation process, are:
- Chapter II-2: construction- fire protection, fire detection and fire extinction
  Part B – fire safety measures for passenger ships
- Chapter III: life-saving appliances and arrangements
  Part B- ship requirement
The recommendations mainly refer to the description of procedures, number and functionality of the escape devices, drills, crew training and so on. But part of them deals with the escape route requirements. They point out that the escape routes shall be as direct as possible, not obstructed, and their accessories (handrails, handholds, unlocked doors...) localisation means (“you are here” plans, arrows, deck numeration...) are well described.

Ro-ro passenger ships constructed on or after 1 July 1999 are required to undergo an evacuation analysis at an early stage of design (Chapter II-2).

Present and future (July 1st 2002) SOLAS Codes do not (and will not) include additional instructions about arrangement of means of escape: "the widths, number and continuity of escapes shall be in accordance with the requirements in the Fire Safety Systems Code”.

2.2. HSC Code


The HSC Code defines:

**Category A craft** is any high-speed passenger craft:
- operating on a route where it has been demonstrated to the satisfaction of the flag and port States that there is a high probability that, in the event of an evacuation at any point of the route, all passengers and crew can be rescued safely within the least of:
  - the time to prevent persons in survival craft from causing hypothermia in the exposure in the worst intended conditions,
  - the time appropriate with respect to environmental conditions and geographical features of the routes, or
  - 4 hours; and
- carrying not more than 450 passengers

**Category B craft** is any high-speed craft, other than a category A craft, with machinery and safety systems arranged such that, in the event of damage disabling any essential machinery and safety systems in one compartment, the craft retains the capability to navigate safely.

**High-speed craft** is a craft capable of a maximum speed (m/s), equal or exceeding \( 3.7 \sqrt{V^0.1667} \), where \( V \) is the displacement corresponding to the design waterline (m³).

**Passenger craft** is a craft that carries more than twelve passengers.

Among the main recommendations one can highlight:
- Public spaces should be designed to protect passengers and to help them moving in case of evacuation.
- Seats should not obstruct access to, or use of, any essential emergency equipment or means of escape.
- It deals with the evacuation conditions: position and number of exits, exit doors opening, exits marks, passage dimensions...in day light or in darkness conditions. Recommendations are only vague, not explicit.
- It contains the evacuation procedure and the maximum evacuation time authorised.

\[
evacuation\ time = \frac{SFP - 7}{3} \text{ (min)}
\]

\( SFP \) is the *structural fire protection time* (min). The subtraction of 7 minutes corresponds to the initial period for detection and extinguishing action.
3. Existing methods

We review quickly the most important techniques for simulation of passenger evacuation:

3.1. Hydrodynamics

The first simulation methods for crowd flows or urban traffic were based on hydraulics analogies which consider a continuous flow with variable density. The main problem is to determine this density by solving a system of partial differential equations.

**Hydraulics approach:** The most "simple" or at least most used hydrodynamics method consists in considering the flow as a compressible continuum. For example, a car line taking traffic signs into account can be modelled using a Burger problem, *Lesser and Corkill (1983)*. This method is quite good and realistic for traffic flows with a high density on one line, but it experiences many limitations such as the difficulty to introduce stochastic or statistic drivers' behaviours or a weakness for defining heterogeneous car profiles. As a Eulerian approach, it does not fit well to analyse individual results as point to point running time for a given car. The Italian Classification Society RINA has implemented such an approach for passenger evacuation.

**Particles methods:** The particles methods for solving partial derivative equations have been widely developed in the last ten years. Main applications are dedicated to computational fluid dynamics. For simulating granular continuum, the regularised particles method (SPH - Smooth Particle Hydrodynamics), *Benz (1989)*, is used. Some industrial applications have appeared, e.g. for material damage models, *Benz and Asphaug*, or fast dynamics, *Petschek and Libersky (1993)*. In spite of the "particular" aspect of the method, it relies on a continuum description, and so does not fit very well to the evacuation problem. But some algorithms as seeking neighbours may be useful.

3.2. Cellular automata

A cellular automaton works in a dynamical system where space, time and states are discrete. A state with a known set of possible values is associated to each point of a Cartesian grid, called cell. Local rules allow updating the cell states. At a given time, a cell state is only linked to its own and neighbours' value at the previous time step. All cells of the grid are updated simultaneously, and their state varies upon discrete time steps. Cellular automata have been applied often to simulate traffic flows or crowd evacuation. The research work at the University of Duisburg in Germany, *Esser and Schreckenberg (1997)*, is an interesting reference. In these methods, each individual cell can contain only one actor (person or vehicle). Algorithms are highly parallelizable and very efficient for the geometrical aspect of displacement. But the efficiency decreases quickly when rules of management for actors' interaction become complex.

3.3. Distributed Artificial Intelligence

Traditional Artificial Intelligence (AI) tends to model the intelligent behaviour of one unique agent, Distributed Artificial Intelligence (DAI) focuses on intelligent behaviours resulting from the cooperative activity of many agents, *Huhns (1987)*. It allows a distribution of expertise on a group of agents which have to be able to work and act into a common environment and to solve possible conflicts. Various implementations exist: Parallel Artificial Intelligence, Distributed Problems Resolution and Multi-Agents Systems. For evacuation problems where individual behaviour may be more important than group behaviour, components must be able to "think". Thus they must be provided with perception and action capabilities and must include a given autonomy to behave. Multi-Agents Systems appeared as the best solution, *Durfee et al. (1987)*.

4. Multi-Agent Systems

An agent may be defined as an entity (physically or abstractly) able to act on itself and its
environment, provided with a partial representation of this environment, able to communicate with other agents and with a behaviour coming from its observations, knowledge and relationship with other agents, *Ferber and Ghallab* (1988).

Three different kinds of agents exist:

**Cognitive agents:** They have a comprehensive set of knowledge-based capabilities of thinking, allowing them to solve more or less sophisticated problems and interaction with other agents. Each agent may be regarded as a little Expert System.

**Reactive agents:** For supporters of this method, each agent has not to be fully intelligent to simulate group intelligence. He has just to understand a communication protocol based on a stimuli-action based approach.

**Hybrid agents:** As far as each of the previous methods are not obvious to implement in a whole, it is often better to use a hybrid method combining local and global views, *Chaib-draa and Paquet* (1993).

### 4.1. Functions of a cognitive agent

Agents are provided with various features (intentionality, rationality, involvement, adaptativity, autonomy, anticipation, emotivity). Fig.2 gives the structure of an agent. An agent may acquire knowledge on external environment (perception). He has also some interaction capabilities with the other agents (communication, negotiation). Relying upon knowledge, his beliefs and his goals, the agent must build a plan for action. For that, he uses his expertise to decide which first goal to chose and to complete and which action he plans to reach it. Knowledge may come from initial knowledge (I remember my way to enter the room) or from the environment (Where is the next exit sign?). The beliefs represent the way the agent interpret the information, as far as the perception is not always rational (an agent can miss an exit sign). Planning cannot be decided in a static way. The agent must take into account the vicinity of other agents and to negotiate with them (communication) before deciding his actions (e.g. possibility of collisions or leader followship).

![Fig.2: Typical functions of a cognitive agent](image)

### 4.2. Agents Society

An agent's society may be defined by three main elements, *Gasser* (1990): a set of agents, a set of tasks to achieve and a set of objects related to the environment. An agent may demonstrate his ability to achieve a task and to overtake the responsibility for it. Then he takes a leadership attitude in the group. In such a case to realise a task he may have to inter-operate with objects of the environment or other agents (i.e. helping a lost agent or mustering an agents group). There are various aspects to include in the agents to maintain the group behaviour consistency:

**Control and decision mechanism:** Control is to be included to define the rules of the society creating a kind of master-slave relationship between agents and a super-agent (supervisor). The decision mechanisms must include for instance all relevant conclusions after a negotiation process.

**Cooperation:** In one agent's model of the environment, information coming from other agents must
be included. For instance an agent may suspend a plan to help another or ask for help when his knowledge is not sufficient to find a solution by himself. Cooperation needs a coordination that may help in finding a common solution for a group or in solving a conflicting situation. Coordination may be centralised (supervisor) or decentralised (agents). Cooperation also needs negotiation that includes mechanisms to get the agents coordinated.

**Communication:** Communication includes mechanisms and protocols to support the negotiation. The protocols must include all relevant "words" of the restraint "language" used by the agents to cooperate. Communication may be through a shared memory or on a point-to-point basis. The most efficient approach is to let the agents communicate point-to-point, but using a supervisor as a post office.

5. Communication

Some experiments on communication in case of emergency have been carried out. Interesting results are included here after *Poole and Springett (1998)*.

**Test Experiment 1:** The first experiment relates to the concept of agent communication. Rational agents have the ability to give information to other agents to help them to escape more effectively. Rational communicating agents have been used to supply information to uncertain agents. The percentage of irrational agents that escape is measured against the amount of communicators in the room. As expected, an increasing amount of positive communication within the system allows agents with irrational behaviour to escape the room more easily, Fig.3. This result can be interpreted as meaning that if more rational agents help other less capable agents to escape then these less capable agents will stand more chance escaping the room.

![Communication](image)

Fig.3: Number of *Rational* communicators in the room vs. number of evacuated *Irrational* agents

**Test Experiment 2:** Another manner in which agents can receive information is through sensory information relating to exit sign positioning. Exit signs provide localised information relating to the general position of the main exit. More information provided should enable the agents to evacuate a room more effectively. The percentage of evacuants is measured against the number of exit signs positioned at random in the room. Because of the stochastic effects relating to the initial positioning of agents and signs the results are averaged over 25 runs. Increasing the number of exit signs in the room produces the interesting results of Fig.4. The escape performance of the agents is initially increased, reaching generally high levels between 10 and 25 exit signs, but then performance seems to fall away. The reason for this relates to the nature of agents sensors. These agents will constantly look for exit signs and follow the information until they see an exit to escape from. Beyond a certain amount of exit signs it seems that agents become ‘confused’.
Fig. 4: The percentage of evacuants against the amount of exit signs

**Test Experiment 3:** The final experiment shows how the general escape flow of ‘Rational’ agents slows down when their avoidance behaviour increases. This sensitivity is important because it can model the development of bottlenecks in crowds. If people are sensitive to avoiding other agents the results of Fig. 5 indicate that more bottlenecks emerge and this greatly affects general escape speed. These rules have provided a simple and provisional set in order to demonstrate agent movements through space. For the current project, the agents need to behave as much like a real collection of people on board to accurately model crowd flow through marine vessel architectures. For this, it is important to ensure firstly that the agent model is sufficiently realistic and secondly that it is executed within an accurate computerised representation of a given marine vessel.

![Graph showing percentage of evacuants against exit signs](image)

**Fig. 5:** The flow of *Rational* agents through time at various levels of avoidance sensitivity

### 6. Object-oriented model of the simulator

The simulator is based on a time steps basis that complies with the following schema:
- Initialisation (reading of the database or input files)
- Main loop
  - Time management
  - Updating external world (introduction of hazards)
  - Updating agents (realisation of actions, updating goals)
  - Dealing with conflicts from the current goals
- Possible saving of intermediate or final results
The database or input files contains the description of the scenario (time steps to use, initial position and speed of the agents, time of appearance of events such as hazard or initialisation of a specific agent), description of the space (walls, passages, obstacles and zones with their status such as stairs or muster station) and description of the agents (personality such as maximum speed (physical) or maximum level of vexation (intellectual), initial state), Fig. 6.

6.1. Supervisor

The first supervisor function is to be an interface to the ODIGO database and input/output files. For instance, it is in charge of reading initial data and parameters of the simulation (scenario, space, agents). Then it manages time steps (RunStep method which in turn trig the agents' RunStep method) and appearance of events as initialisation of delayed agents and hazards changing the space state (passage being blocked, fire propagation, ringing alarms). Finally, it is responsible for saving interim or final results from the simulation. The second supervisor function is to help agents unable to deal with a negotiation. Following the same idea, it may be mandated to give an oriented perception of the reality to the agents upon the interest of the overall population. In the first implementation, the supervisor is slightly managerial (reactive agents oriented) but as the agents gather more and more intelligence, the supervisor withdraws to the background leaving more initiative to the agents (cognitive agents oriented).

6.2. Agents

For each time step one single agent executes the following tasks: Feeling, Thinking, Acting. The Feeling function is carried out using a perception object communicating with the supervisor to build beliefs of the external world. The Thinking function is carried out using a planning object that builds a plan and select an immediate goal to achieve. Finally a Goal object selected by the planner deals with the Acting function. To do so it runs a set of behaviour rules and combines the resulting actions to modify the internal states, Fig. 7.

We can highlight some specific aspects of this schema:

**Belief:** To react or decide, the agent needs beliefs of the external world which have not to be rational. Beliefs are maintained by the perception. Some beliefs may result from a combination of various pieces of information based on the external world and internal states. Examples of beliefs are:

Presence of a passage, vicinity of other agents, obstacles on the way or states of an other agent as "lost", "panicked", "wounded"...
**State:** A state represent an internal feature of the agent together with its current value. At each time step a state can be updated by the perception relying upon events. Examples of states are: position, speed, tiredness, moral, stress or vexation.

**Personality:** To decide whether an action can be engaged the agent uses its personality. Usually, a personality is linked to a state and represents a reference value (maximum possible level for instance). The population will comprehend agent with various personalities deduced from a typical set of passengers (percentage of old people, of potential leaders…).

**Goal:** The agents have a stack of goals organised by priority. Some goals are immediate, others are long term. For instance the father of a family may have two major goals: (1) to remain grouped with his family and (2) to reach the next exit door.

![Diagram of the relationships for the Agent object](image)

**Fig. 7:** Schema of the relationships for the Agent object

### 6.3. Space

The Space is an object embedding all geometrical information to be used by supervisor or agents. The top most object used by the space is a Zone. A Zone includes links to other geometrical data as Walls, Passages or Obstacles. A Zone has also a Type (for instance muster station or stair) and States (under fire, non horizontal floor), Fig. 8.

![Diagram of a Zone](image)

**Fig. 8:** Schema and example of a Zone

### 7. Conclusion and perspectives

A sample piece of software of ODIGO has been developed already. An current project is dedicated to add more intelligence and human behaviour in the agent model. One major aspect of such a development is to validate the model. For that, evacuation drills have to be planned and public results from EC funded research projects collected.
One must bear in mind that such simulators would give just approximate values (as evacuation time) that must be obtained only with many different populations and scenarios to give average values. An interesting result would be qualitative, such as identifying bottlenecks or evaluating the fluidity of pathways.

For the future, the French Navy has also voiced interest in the approach for simulating red alert codes, fire extinguishing and embarkation/disembarkation as well.

![Fig.9: A sample visualisation of the simulator](image)

**References**


Multi Objective Optimization for the Master Plan of Cooperative Assembling with Several Shipyards

Marcus Bentin, Hiroshima University, Hiroshima/Japan, bentin@hiroshima-u.ac.jp
Kunihiro Hamada, Hiroshima University, Hiroshima/Japan, hamada@naoe.hiroshima-u.ac.jp
Mitsuru Kitamura, Hiroshima University, Hiroshima/Japan, kitamura@naoe.hiroshima-u.ac.jp

Abstract

The competition in the naval industry is rising. Therefore, productivity has to be increased and planning risk lowered. In order to cope with this challenge, horizontal cooperation with other shipyards in the scope of Supply Chain Management is considered as a good solution for increasing the productivity. In cooperative assembling, compound and sophisticated production planning is demanded with considerations of various factors. A good planning tool for the master plan of cooperative assembling should be able to optimize the profit of the new planned project. Therefore, a new planning tool for cooperative ship assembling is discussed. The objectives of the proposed system are makespan and cost. These are successfully supported by a Simulation System integrated in a Product Model and enhanced by a multi objective optimization system. In order to realize multi objective optimization, a NPGA (Niche Pareto Genetic Algorithm) is used as optimization algorithm.

To determine the objective cost, a cost model is developed that calculates the variable costs based on the data from the Product Model and the cost definition at each cost center that is modeled for each production unit. Several cost types can be modeled. It is possible to analyze the costs according cost types, cost centers and projects or modules.

However, the objectives makespan and cost are influenced by the expected quality and the quality ability of each production unit. Therefore, a quality model is developed to describe the expected quality of each module and the quality ability of each production unit. The quality affects the makespan and the costs and is therefore a restriction of the optimization system.

Moreover a concept for managing multi project optimization is developed. Here it can be distinguished between already scheduled projects as boundary condition for the new project and product mix optimization. Some examples are shown at the end of this paper.

1. Introduction

The competition in the naval industry is rising. The market for new ships and maritime structures is ruled by low prices, reliability of the delivery date and short production time. The customers require higher quality and agreements to individual wishes, which imply flexible manufacturing. The products are designed and manufactured to the customer's notations. In addition, the customer expects a low price. Consequently, it is very important to work economically, i.e. to optimize production and to think about the possibility of cooperation with other shipyards. The effectiveness of cooperation is proved by the experiences derived from Supply Chain Management implementations in other industries, e.g. Fleischer. et al (1999), Taylor (1997), Filling et al (2000). In cooperation, it is possible to produce cheaper and faster and to face the challenges of the market more powerful. Therefore, the cooperation has to be managed well otherwise conflicts lower the potential time- and cost savings. To support this management a good plan and plan generation tool is, beside a trustful relationship, the key to a productive cooperation, e.g. Stadtle and Kilger (2000).

In shipbuilding, the planning starts with an inquiry of a customer. The answer has to be given fast and the processing costs have to be low because in most cases, the contract is not given to the shipyard. Therefore, only some key events are time-phased and the calculations of the costs are done very roughly. In these situations, the experience of the planner is the most important factor. For a binding offer, the planning risks have to be lowered and it is planned a little more in detail. At this stage, the plan has the character of a master plan. The master plan belongs to the operative plans with a mid-term planning horizon. It contains an outline of regular operations, rough quantities and times. In shipbuilding, the planning starts with the assembling of the ship hull because all other
operations, e.g. outfitting, furnishing and piping need the ship hull. Further, it gives time windows for the production plan, e.g. Sladojevic (1996). Often network plans (e.g. Petri Net) are used to generate this master plan. The capacity is checked with typical capacity curves developed from former finished ships. The problem at this level is that the generated CAD model is not connected with a production resource management system. Hence, the makespan for each production is estimated by average working hours per tons steel and the working costs are estimated by multiplying these working hours with an average wage per hour, e.g. Nedelj et al. (1992). For managing and optimizing a master plan for cooperative assembling, this approach is not very useful. To improve this process, the authors developed a new master plan planning tool for cooperative ship assembling Bentin et al. (2002). A discrete simulation system was developed and integrated into a Product Model. Afterwards, it was connected to an optimization system that is capable of optimizing the makespan. However, the parent system solves only a part of the above described problems, the makespan and resource management problems.

In this paper, a more powerful planning tool is introduced based on the former developed system. The target of the new system is to maximize the profit. In order to realize this, the parent system is enhanced by a cost- and quality model as well as a multi objective optimization engine. Therefore, the objectives makespan and cost are optimized by considering some boundary conditions of the master plan, e.g. quality and already allocated resources by beforehand scheduled projects. Moreover, it supports multi project optimization to optimize the product mix.

2. Overview of parent system

2.1. Configuration

The first step in developing the master plan planning tool for cooperative assembling was to connect a Product Model, simulation system and optimization system. It is proved by former studies that the connection of Product Model and simulation system helps efficiently to plan operations and to determine the man-hours for each operation, e.g. Aoyama et al. (1999). The basic configuration of the parent system is presented in Fig.1, Bentin et al. (2002). The system contains a Product Design System to model the Product Model with geometry, parts, rooms, module hierarchy and connecting relations, e.g. Aoyama et al. (1997). The Simulation System consists of three main parts:

- The Production Unit Design System to model the production units (PUs) and their operation speed ability on basis of assigned workers.
- The Operation Information Generation System is used for modeling the operations “Assembly”, “Storage” and “Transportation”. This subsystem is connected to the Product Model to extract the module hierarchy and required information to calculate man-hours.
- The Scheduling System supports the assignment of production units to each operation and the modeling of the order of priority of the operations at each production unit. In the end, this subsystem calculates the makespan and the earliest beginning and ending date for each operation.

```
Product Design System (Product Model)

Simulation System

Optimization System

Operation Information Generation System

Scheduling System

Main Controller

GA

Production Unit Controller

Optimized Gantt Chart

Fig.1: Basic system of the first developing level
```
The Optimization System is connected to each subsystem of the Simulation System by controllers. These controllers manage the design variables of the optimization problem:

- Number of workers at each production unit
- Assignment of production units to each operation
- Order of priority of operations at each production unit

### 2.2. Generation of Operation Information

In cooperative assembling, the consumed time for non-Assembly Operations increase because of the transportations between factories. Therefore, the developed Operation Information Generation System contains a new representation model for operations. They are described as relation between modules in their hierarchy, relation in place and time. Further, the operation is described as event with a target (e.g. length of welding line), a subject (i.e. production unit) and content (e.g. welding). This operation model is combined as event and relation with other operations to describe the operation sequence as described in Fig.2. The modeling of the operation model is done in three levels:

#### Level of Change in Forms for generating Assembly Operation (Fig.2 (1))

The layered structure of the assembly modules is extracted from the Product Model and Assembly Operations are generated in this hierarchy. The operation objects/targets are extracted automatically from the Product Model, too. Then, the production units are assigned to the assembly operations and the man-hours are calculated, according to the capability of their production unit. The place information is implicitly given due to the assignment of the production unit to the operation.

#### Level of Change in Places for generating Transportation- and Storage Operation

When the level of the Change in Forms ends, the places where Assembly Operations are executed are extracted and the remaining operations are modeled continuously. Firstly, Storage Operations are generated automatically after each Assembly Operation and are assigned to a stockyard (Fig.2 (2)). Then, Transportation Operations are generated between the production units. Afterwards, the Transportation Operations are assigned to the transportation units and the man-hours for the transportation is calculated (Fig.2 (3)). The assignment of Storage- and Transportation Operations to the production units has to be decided by the user or the Optimization System.

#### Level of Change in Time for Scheduling

Consecutively, the operations have to be scheduled using the network structure of operations, the man-hours that are generated through the above-mentioned steps and experience. Hence, the order of priority of operations at each production unit has to be modeled by the user or Optimization System.

---

Fig.2: Representation of the Operation Model and its generation process
2.3. The Optimization System

The optimization algorithm at this level is a Simple Genetic Algorithm (SGA), e.g. Goldberg (1999), with two kinds of chromosomes that represent the design variables (Fig.3, Gen and Cheng (1997)). The Main Controller is responsible for the de-and encoding of the chromosomes. The other controllers give this information to each subsystem of the Simulation System and remodel automatically the simulation model. Therefore, it needs some additional information like:

- Routing table
- Grouping of production units
- Production unit hierarchy
- Minimum and maximum of assigned workers to each production unit
- Storage assignment to assembly production units

At this level of development, the objective is minimization of the makespan.

![Employment Chromosome and Scheduling Chromosome](image)

Fig.3: Employment- and Scheduling Chromosome

3. Basic concept of the new developed system

The final aim of optimization is to enlarge the profit, which can be optimized in two ways. The first one is to maximize the “Output”, i.e. sales. It can be achieved by asking for more money for the same product if the market allows this strategy. The shipbuilding marked is characterized by a big competition in price. Hence, it is not possible to ask for more money. In this manner, producing more ships in the same time can only enlarge the “Output” assuming that the production volume is sold. The target of maximizing the “Output” leads to saving as much production time as possible. However, this is only effective if the costs are growing slower than the saved makespan. It depends on how this saved time is used for following projects to compensate the additional costs for saving makespan. This can only be evaluated by a multi project optimization, where several product mixes are compared and optimized.

The second way is to minimize the “Input”, i.e. cost of the “Output”. It can be enlarged by investing money in new production units that allows productions at lower price or by cooperation with partners, who are more prolific in some assembly stages. If the costs for the same “Output” are reduced, the profit is rising. This leads to the target of minimizing costs.

Therefore, the optimization objectives of the new developed master plan optimization system are makespan and costs of each project. Hence, cost and makespan are also influenced by the quality the customer expects and the quality ability of each production unit.

This multi objective optimization has the same design variables than the former developed optimization system (Fig.4) and an already generated Product Model as initial condition.

- Number of workers assigned to each production unit
- Assignment of production unit to each operation
- Order of priority of the operations at each production unit

Furthermore, the system considers several restrictions and boundary conditions like:

- Limited transportation capacities in number and size
- Limited quality abilities of the production units and expected quality
- Limited storage- and erection space
- Limited workers
- Working days and working hours per day for each production unit
o Already scheduled projects
The system builds up on the former developed Master Plan Optimization System. Some new models and changes are developed to realize the multi objective, multi project optimization. Following major changes and add-ons where necessary:
o Cost Model for calculating cost
o Quality Model to describe quality ability and expected quality
o A new reproduction method to implement the Niche Pareto Genetic Algorithm (NPGA) as optimization engine
o Advanced scheduling system that recognizes automatically beforehand scheduled operations and schedules the new operations around the already scheduled operations

Fig.4: Configuration of the new developed system

4. Cost Model

4.1. Cost types considered in the Cost Model
The considered costs are the variable or direct costs that are caused directly by the production of the product. The cost accounting is based on the concept of profit contribution because this concept avoids the mistakes that can be done by distributing the fixed cost to the project, e.g. Dümler and Grabe (2002). The fixed cost has to be covered by the profit contribution that is given by sales minus variable cost.

Table I: Cost types and cost variables at Production Units (PU)

<table>
<thead>
<tr>
<th>Cost Type</th>
<th>Assembly PU</th>
<th>Transport PU</th>
<th>Inter.Shipyard Transport PU</th>
<th>Stockyard PU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personnel Cost</td>
<td>Man-hours</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Material Cost</td>
<td>Weight of Material</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>Consumable Cost</td>
<td>Length of Welding Line</td>
<td>Length of Trans.Way</td>
<td>Length of Trans.Way</td>
<td>--</td>
</tr>
<tr>
<td>Payload Cost</td>
<td>Weight or Volume of Material</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Interest Cost</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>Storage time &amp; Module Costs</td>
</tr>
</tbody>
</table>
Depending on the type of production unit (assembling, stockyard or transportation) different cost types are defined (Table I). The definition of the cost types is based on characteristic cost variables, e.g. length of welding line, weight of material etc. The value for these variables is taken from the Product Model and operation attributes (Table I). The object data model of the Cost Model is shown in Fig.5.

4.2. Calculation of the cost

The user must specify a cost factor, e.g. Yen/working hour that is multiplied with the cost driving variable to calculate the cost. After the time calculation, the costs are calculated for each operation at each production unit depending on the definition of cost types for each production unit. The costs are calculated automatically based on the beforehand-modeled cost factors. The cost driving variable is extracted and calculated for each cost type. Then the cost is calculated by multiplying the cost driving variable with the cost factor of each Production Unit:

- **Personnel cost** is calculated from the man-hour of each operation defined in the simulation system multiplied with the cost factor “Yen/hour”.
- **Material cost** is calculated from the weight of the module parts, calculated from the Product Model multiplied with the cost factor “Yen/Kg”.
- **Energy- and consumable costs** are calculated from the length of welding line taken from the Product Model and multiplied with the cost factor “Yen/meter”.
- The transportation costs for in-shipyard transportation is calculated by the energy- and consumable costs, which base on the length of transportation way taken from the simulation system and is multiplied with the cost factor “Yen/meter”. For inter-shipyard transportation, the costs can be calculated based on payload cost. Therefore, the weight or volume of the module is calculated from the product model and multiplied with the cost factor “Yen/Kg or Volume”. The interest cost at the stockyard is based on the interest as cost factor and on the summed cost of the module that is stored as well as the storage time.

Calculated costs are described in each Operation Model. The Operation Models are managed according their projects. Therefore, it is easy to calculate the costs of each project by gathering all related operations belonging to the project.

5. The Quality Model

5.1. Basic concept

The developed Quality Model consists of two sides. One is the quality ability of each assembly production unit. This describes the quality level of work for each operation the production unit is able to do. The other side of the model is the expected quality level for each operation. The modeled quality is considered as a restriction of the operation. During the assignment of the assembly production units to the assembly operations, it is checked if:

---

Fig.5: Data model of Cost Model  
Fig.6: Data model of Quality Model
o Quality ability of the production unit << expected quality level for the operation
  ➢ The production unit is not assigned to that operation and another one that satisfies the
  quality requirements is chosen.

o Quality ability of the production unit < expected quality level for the operation
  ➢ Rework is taken out to reach the required quality level. This causes extra costs and
  extra man-hours.

o Quality ability of the production unit ≥ expected quality level for the operation
  ➢ The production unit is assigned to the operation.

The quality is defined by four sub qualities:
  o Surface (e.g. buckling, “Hungry Horse”, primer, paint)
  o Accuracy (e.g. shrinking, positioning, cutting)
  o Weld (e.g. bubbles, polished, other special treatment for high quality steel)
  o Corrosion (e.g. of steel plates, of weld, structure)

The quality scale has to be defined beforehand and used consistently for the whole quality definition.
The object data model of the Quality Model is presented in Fig.6.

5.2. Procedures and its functioning in the Quality Model

The Quality Model concept is realized in the system in the following way:

1) Definition of quality ability at the Production Units
   Firstly, the quality ability at each assembly production unit has to be defined by modeling an
   average quality level for each sub quality as mentioned above. Further, the number of quality
   levels that can be recovered, if rework is done, is defined. Therefore, the expected extra man-hour
   and cost are modeled for recovering one quality level. Moreover, information has to be given at
   which assembly hierarchy level the rework is taken out.

2) Definition of required quality for the operations
   Hereafter the required quality for the assembly operation has to be modeled in the module
   hierarchy for each module. The quality level that is modeled for each module determines not the
   overall quality level of the module, but the quality level of the operation to assemble this module.
   The overall quality of a module is not easy to define and it depends on how much attention is put
   to the quality of some parts of the module. Here only the quality of the operation is lighted.

3) Checking the quality level of operations and its influence on cost and makespan
   When the Simulation System is scheduling the operations to the production unit, it checks if the
   expected quality level for the operation can be reached, including rework. If this is not possible,
   the operation is not scheduled to that production unit and another one that satisfied this criterion is
   chosen.

During the time calculation, the rework time is determined and included in the earliest finishing date
of each operation using the rework definition in 1). Afterwards the costs are calculated. Therefore, a
special cost type for rework is created at each cost center, if rework occurred in one operation. This
connects the Cost Model and the Quality Model. Hence, quality is a restriction for the objectives
“Cost” and “Makespan”.

6. The Optimization System

6.1. The objectives

The optimization includes two objectives, “Makespan” and “Cost”. The makespan is calculated by
end date minus beginning date of the project. The second objective cost can include many things. At
first, it includes the sum of variable costs for the project. Further, this objective can be enlarged by
some punishment terms, e.g. lack of storage capacity, late delivery (Eq.2). These punishment terms
are some how related to the objective “Makespan” but they can be transformed easily in the manner of
costs, which is more important if the profit is optimized. Both objectives are minimized during the
optimization.
Objective(Makespan) = |Project End Date − Project Begin Date| in Seconds

Objective(Cost) = \[ \sum_{i} [PC(m_i) + CC(m_i) + EC(m_i) + MC(m_i) + SI(m_i) + RWC(m_i)] + cf_1 \sum CL + cf_2 \cdot AT \]

m_i = Module, PC = Personal Cost, CC = Consumable Cost, EC = Energie Cost, MC = Material Cost
RWC = Rework Cost, SI = Storage Cost (Interest), CL = Capacity Lack of Stockyard and Erection Spance
AT = Late delivery, cf_1, cf_2 = Cost Factor

6.2. Advantages of multi objective optimization with Pareto concept

Frequently real world applications and their objective functions show us multi attributes like in this optimization. Often objectives are transformed into constrains or penalty functions, or they are combined to one fitness function by weight factors. Penalties and weights are considered problematic. They rely on a priori knowledge of appropriated weights or constraint values. Furthermore, these techniques are only capable of finding individual points on the tradeoff curve (or surface). Finally, the solution is usually very sensitive to small changes in the penalty function and weight factors. True multi objective methods have the potential to generate simultaneously all possible optimal combination of objectives, with less afford than other approaches, e.g. Horn et al. (1994).

Therefore, the concept of Pareto optimum is used in this study. The Pareto concept bases on the classification of dominated and non-dominated points. One candidate dominates another only if it is at least equal in all objectives and superior in at least one. The degree of domination for a candidate is proportional to the number of candidate it is dominated, e.g. Horn et al. (1994). In the following example presented in Fig.7, where the aim is to minimize both objectives, the candidates 1 and 2 dominate the candidate 3. This idea is used in the developed multi objective optimization system.

6.3. Implementation of the Pareto concept in the Optimization System

The main flow of optimization is described in Fig.8. The chromosomes and the controllers that manage the design variables and their genetic representations are not changed from the previous Optimization System as well as the crossover and mutation functionalities of the Genetic Algorithm. Only the way of reproduction (selection and mating) is changed to follow the idea of multi objective optimization. Hence, for selection, the candidates of each generation are analyzed according their stage of Pareto domination. The candidates are put into layers that represent the same stage of domination. These layers are ranked according their generation sequence, e.g. Goldberg (1999).

In order to achieve a big amount of diversity in the Pareto front, the concept of Niche Pareto Genetic Algorithm (NPGA) is used for selection and mating, e.g. Horn et al. (1994).

The NPGA is based on tournament selection. Firstly, a set of individuals is chosen randomly from the current population and a Pareto domination tournament is carried out (Fig.9). After that, sharing is implemented by niche count to determine the winner if a tie occurs. The niche count is the sum of individuals in the neighborhood of the individual. The neighborhood is defined by a circle around the individual (Fig.9). The winner of the niche count is the individual with fewer neighbors and is taken for mating. In the presented examples, tournament size is set to 10% of the population size, and the niche radius is set to 5% of the standard radius.
7. Multi project optimization

7.1. Concept of Order Book and Order List

It is important for multi project optimization how the operations are managed. Therefore, the operations are managed by referring to their project and assigned production units (Fig.10). Order Book and Order List are prepared as managing mechanisms at each production unit. The Order Book is used for managing the assignment of production units to each operation. The Order List is used for managing the order of priority of all the operations (two or more projects) that are processed at the production unit. The flow of this managing process is shown below (Fig.11):

1) A Scheduling Chromosome is generated for each group of production units to schedule the operations of several projects to the production unit and to define the order of priority of the operations at each assigned production unit.

2) An Order Book of each production unit is generated and filled with the assigned operations, based on the Scheduling Chromosome.

3) At each production unit, the man-hours for the newly assigned operations are calculated and time-phased. Hence, in the first level of calculation the already scheduled operations of former projects are neglected. This means, the order of priority of the former scheduled operations are already planed and cannot be changed. They are no longer available for the optimization process.

4) In the second calculation level, the Order List is updated with the new assigned operations by scheduling them around the already scheduled operations based on the starting-and ending date of each operation and the order of priority defined by the scheduling chromosome for the newly assigned operations.

The concept of Order Book and Order List gives the possibility to track the status of the operations and to compare this with the real status of assembling progress given by each control center of the production units. It supports planning with real world boundary conditions.
7.2. Processing the multi project problem in the new system

There are two ways how to interpret multi project management. The first one is planning with resources that are already occupied by former scheduled projects as boundary condition. The other one is that several projects are planned together as one big project.

1) Already scheduled projects as boundary condition
Each operation can be set as “scheduled” so that they are no longer available for the optimization process. However, they are boundaries, where the new operations have to be scheduled around. It influences the overall order of priority at the production unit so that the “Scheduling Chromosome” is now giving a relative order of priority for the new operations that have to be scheduled. This means that the operation with the number 1 in the chromosome is not automatically the first operation that is processed at the assigned production unit because of beforehand scheduled operations. The sense is that it is the first operation of the new project scheduled to that production unit. Hence, the new operations are put in their planned sequence in front of the first found already scheduled operation. During the time calculation, it is checked if the calculated end date of the new operation is lower than the first starting date of the already scheduled operation. If false, the operations are shifted behind that scheduled operation until a time hole is found where it can be processed.

2) Several projects as one big project
Several projects are gathered as one project. Therefore, the optimization system generates a genetic representation like for the single project problem. The chromosomes are now only longer but no chromosome is generated in addition. It is important that projects are not losing their identity. Hence, the projects are identified by their project names not by the name of the ship that represents the Product Model. This means the same ship can be used for several projects, which makes it easy to define a standard ship type for analyzing some strategically questions like product mix. Each project can have its own starting and ending date but common dates can be given, too.

8. Outline and applications of the Prototype System

8.1. Workflow of using the proposed system

Product Design

The first level of the optimization process is the modeling of the ship structure by a Product Design System. At this level the geometry, material properties, components of the structure like plates and stiffeners and connecting relations between these components are defined. Then the whole structure is divided into modules and a module hierarchy for assembling is modeled. Afterwards the quality for each assembly operation is defined at each module. In Fig.12, the used structure for the example is shown with its hierarchy and expected quality of the assembling operations.

Fig.12: Module hierarchy and expected quality of the big and small ship structure
Production Unit Design

For simulating the assembling, the production units and their operation capabilities have to be defined by using the Production Unit Design System. For the examples described later, three shipyards and their production units, are modeled as well as the transportation units for the inter shipyard transportation. Further, the workers with their needed min/max number and their functional relations between the Number of Workers and operation speed for each production unit are modeled. Then the cost definitions are described for each production unit by setting the cost factors. Afterwards the quality ability has to be defined. In Table II and Fig. 13, the modeled production units for the examples are displayed. The costs differ only in personnel cost. The inter-shipyard transportation costs are assumed with 10000 Yen/tons. All assumed values are fictive.

Operation Modeling

The Operation Information Generation System is used to model a scheme of assembly operations into the extracted module hierarchy and to assign a production unit group to each assembly operation. Hereafter, stockyards are assigned to each assembly production unit and a routing table for the transportation is modeled. Using this information, the Operation Controller is able to generate automatically the needed Transportation- and Storage Operations.

Optimization initialization and GA search

In the next step of optimization, the controllers initialize the first generation for the NPGA by generating Employment- and Scheduling Chromosomes at random that are forming together an individual. After the initialization, the Simulation System evaluates each individual. Therefore, the controllers model their respective design variable according to the decoded chromosomes. The Simulation System respects limited storage and erection capacity. Therefore, the objective “Cost” is added with punishment cost if a capacity lack occurs. These make the solution unattractive. During the time calculation, the daily working hours and holidays are considered. Further, the distribution of workers is fixed during the simulation of one project. The inter shipyard transportation is limited in number of ships (three ships for each route) and capacity. The sequence and gathering of transportation operations is decided by a heuristic rule, Bentin et al. (2002). These restrictions are boundary conditions of the optimization. Then the Simulation System is calculating the “Makespan” and “Cost”. Consecutively the new population is classified due to the stage of Pareto domination. Hereafter, selection with tournament and niche count is processed followed by crossover, mutation and again evaluation of the new generated population. This is repeated until a certain number of generations are calculated. In the end, the best solutions are given to the user.

<table>
<thead>
<tr>
<th>Factory</th>
<th>PU</th>
<th>Speed Level</th>
<th>Personnel Cost</th>
<th>Quality</th>
<th>Recover Level</th>
<th>Extra Time %</th>
<th>Extra Cost %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dock</td>
<td>10</td>
<td>2000</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Philipp</td>
<td>GrandA</td>
<td>30</td>
<td>2000</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Assemb</td>
<td>20</td>
<td>2000</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Dock</td>
<td>10</td>
<td>1000</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Popones</td>
<td>GrandA</td>
<td>20</td>
<td>1000</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Assemb</td>
<td>30</td>
<td>1000</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Dock</td>
<td>30</td>
<td>3000</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Kanagaw</td>
<td>GrandA</td>
<td>20</td>
<td>3000</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Assemb</td>
<td>10</td>
<td>3000</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
8.2. Example 1: Multi Objective Optimization

This example is optimized under the following conditions:
- Used ship: One big ship shown in Fig.12 with its quality restrictions.
- Production Units: Production Units shown in Fig.13 and Table II with their restrictions.
- Precondition caused by former projects: There is no project scheduled beforehand.

Optimization results are presented in Fig.14. The Gantt charts of the “Fastest” and “Cheapest” solution are shown in Fig.15. Following points are observed in Fig.15:
- The assignment of the production unit is influenced by the quality expectation and ability of each production unit. The two assembly operations, which require the quality level 4, are not scheduled to the assembly line in Indonesia because it is not able to produce this quality even with rework.
- The difference between the cheapest and fastest solution is given by the two assembly operations with quality level 4 scheduled to Kanagawa as well as the assembly operation at the Grand Assembly with quality 4. Therefore, rework is not necessary in the fastest solution that leads to timesaving. Further grand assembling can start earlier and with this dock assembling starts earlier. This is due to better scheduling.
- On the other hand, the cost savings of the cheapest solution compared with the fastest is given in the fact that the personnel cost of the Kanagawa shipyard are higher and it is slower in assembly production. Hence, assembly production is getting very expensive in Kanagawa.

Example 1: Cheapest Multi Objective

Inter Shipyards Transportation: Ship Chris Indonesia<>Philippines  
Ship Erika Indonesia<>Kanagawa  
Ship Paula Philippines<>Kanagawa

Example 1: Fastest Multi Objective

Fig.15: Gantt charts of Fastest and Cheapest solutions of Example 1
In Table III, the distribution of the workers in this example is shown. The GA supports the production units with high-speed factor so that they are getting even more faster. Therefore, it allocates more workers to the fast production units so that they reach their maximum level of workers. The distribution of workers for the cheapest solution is the same than for the fastest solution. Therefore, timesaving is only caused by the assignments of production units and order of priority of operations. However, there might be still potential to optimize the plan because the GA is only capable to give near optimum solutions. In these cases, the user has to modify the solution taking the GA solution as basis.

8.3. Example 2: Multi Project, Multi Objective Optimization

In this example, two kinds of multi projects are compared under the following condition:
- Used ship (Case 1): One big ship shown in Fig.12 without quality restrictions.
- Used ship (Case 2): Two smaller ships shown in Fig.12 without quality restrictions. Hence, the structure has the size of 40% of the big one, therefore less and smaller modules are used.
- Production Units: Production Units shown in Fig.13 and Table II with their restrictions.
- Precondition caused by former projects: A project is already scheduled as boundary condition.

The fastest solutions of above two cases are shown in Fig.16 and 17. The new additional projects are scheduled around the already scheduled project, and the two new projects of the Case 2 are scheduled in parallel. Therefore, the concept explained in Chapter 7 is verified by these examples.

In this case, the “Big Project” is not attractive. It causes too much idle time at the Dock Kanagawa and the finishing of this project is much too late compared with the other one. On the other hand, the Case 2 finishes 16 days after the already scheduled project, and the caused idle time is less than in the other case. Moreover, cost of the Case 1 is much higher than the cost of Case 2 (Fig. 18).

9. Conclusion

In this paper, an integrated discrete Simulation- and Optimization System for optimizing master plans for cooperative assembling is introduced, as well as a new representation model and modeling philosophy for the simulation model. The conclusions of the new developed Master Plan Optimization System can be summarized as follows:

1) The design variables of the optimization system are not changed for the multi objective/ multi
project optimization. They are: “number of workers”, “assigned production unit” and “order of priority”.

2) The Multi Objective optimization is realized using the already developed Optimization System by implementing the concept of Pareto optimality and NPGA. Hence, a new Cost Model to determine the objective cost is developed so that “Costs” and “Makespan” are optimized together. The costs are calculated based on cost factors and generated operation

Fig.16: Gantt chart of fastest solution of Case 1 “One Big Ship”

Fig.17: Gantt chart of fastest solution of Case 2 “Two Small Ships”
information during the simulation.
3) Quality is implemented as additional restriction. Therefore, a new Quality Model that influences “Makespan” by extra rework time and “Cost” by extra rework costs is developed. Hence, the Product Model is enhanced and the definition of the production units.
4) The management and optimization of several projects is supported by the Simulation- and Optimization System. The same Product Model can be taken for several projects. The multi-project management has on the one side the effect of boundary conditions of already scheduled projects and on the other side; the product mix can be optimized. These functions needed some changes in the order management at the production units.
5) The Multi Objective optimization and the Multi Project optimization are verified by the presented examples.
However, the value of a plan is not only determined by time and cost savings but also the risk of failing the delivering date plays a certain role. This risk can be taken as a further objective for optimization and will be developed in the near future.

Acknowledgements

The authors would like to express their gratitude to Prof. Toshihara NOMOTO and Prof. Kazuhiro AOYAMA of the University of Tokyo, for the permission to use the Product Model SODAS in this research.

References

DÄUMLER, K.-D.; GRABE, J. (2002), Kostenrechnung 2. Deckungsbeitragsrechnung, Neue Wirtschaftsbriefe

132
An XML-Based Data Interface for Simulation of Passenger Evacuation

David Jaramillo, Germanischer Lloyd AG, Hamburg/Germany, jmo@gl-group.com
Tim Meyer-König, TraffGo GmbH, Duisburg/Germany, m-k@traffgo.com

Abstract

In view of the recently released IMO regulations concerning evacuation analysis by means of numerical simulation for Ro-Ro Vessels, Germanischer Lloyd in co-operation with TraffGo developed the software tool AENEAS, which is based on a microscopic, multi-agent modelling for performing simulations of pedestrian flow. AENEAS is tailored for the simulation of egress scenarios on board of passenger ships. It has been certified by the German maritime authority to fully comply with the requirements of MSC/Circ.1033.

AENEAS utilises the geometry information contained in general arrangement drawings to set up a discretised model suitable for the simulation. The import of deck plan geometry is realised through DXF (Drawing Exchange Format), which is a well established approach in the CAD-arena.

However, the use of DXF induces some limitations concerning the export of non-geometric information contained in the drawings and which can be used for the simulation. In order to overcome these limitations a new data interface approach has been developed, based on the Extensible Mark-Up Language (XML). Over the last years XML has become a well accepted and proven standard in the areas of web development, e-commerce, as well as document/data management and exchange.

This paper describes the concept and development of evacML, which is intended to become an open standard for the exchange of evacuation data. The implementations of evacML interfaces in both, AutoCAD and AENEAS are discussed as example of how developers could utilise the approach in conjunction with other CAD Systems.

1 Introduction

According to Greek mythology, Aeneas, a Trojan prince and leader, was the son of Anchises and the goddess Aphrodite. He led his family to refuge from the flames of the destroyed city of Troy, while carrying his crippled father on his back.

This legend inspired the name of a new software tool for ship passenger evacuation simulation and analysis, developed jointly by Germanischer Lloyd and TraffGo. The AENEAS tool was designed taking into consideration recently released IMO Regulations (MSC Circ. 1033), which define guidelines for performing numerical simulations and analysis of evacuation processes on passenger vessels.

AENEAS utilises a microscopic approach based on cellular automata algorithms, Meyer-König (1999), Klüpfel et al. (2001) and Meyer-König (2002). Thus the two-dimensional geometrical information found in General Arrangement Drawings requires to be transferred to an appropriate representation for the simulation. This transfer process comprises the conversion of the drawing representation, usually defined as vector graphics, into a discretised (or "rasterised") format; i.e. a mapping of the geometrical data in to a cellular grid. For this purpose, conventions must be agreed upon how entities (lines, polylines, arcs, etc.) are represented by which kind of cells.

Nowadays engineering drawings are mostly created electronically by means of Computer Aided Design software (CAD). Most CAD-systems are able to exchange drawing data with other programs through standardized exchange formats like IGES (Initial Graphics Exchange Specification) or DXF (Drawing Exchange Format). AENEAS supports import of 2D-Geometry of deck plans through a DXF interface. Following rules applies for the DXF interface of AENEAS:

- Each deck plan must be contained in a separate DXF-file.
- The relationship between a drawing entity and a cell type is given through the entity color.
- Text entities containing only ciphers are recognized as a "pedestrian distribution" in the corresponding area.

DXF provides a straightforward approach for the transfer of geometry data to AENEAS. Though, due to lack of standards for the creation of General Arrangement Plans and the diversity of drafting approaches utilized by the ship yards and subcontracted firms, a time expensive and awkward manual procedure of preparation and "cleaning" of the drawing is required prior to importing the data.

Moreover, the DXF interface is limited to geometry information. The only attribute, which is used to establish a relationship between the drawing entity and its "meaning" in the context of a deck arrangement, is given by its color. An exception of this limitation is the interpretation of text entities containing ciphers to generate a randomized distribution of persons (occupants) assigned to an area in the drawing (e.g. cabin or pantry room).

Looking for an alternative format to transfer all data required for evacuation simulation in a well structured and comprehensive way, an XML-based approach was chosen. XML (Extensible Markup Language) is a standard developed by the World Wide Web Consortium (W3C). Over the last years XML has gained great importance in the areas of document/content management, web-technology and e-commerce, as well as in emerging technologies concerning data exchange, management and integration.

2 Technical Implementation

The structure of an XML file can be driven through two different approaches. Document Type Definitions (DTD) represent the traditional method, while XML-schemas constitute the newer, syntactically and semantically "enhanced" possibility of specifying structural requirements for XML-documents, Mertz (2001) and Fallside (2001).

The XML data interface of AENEAS is based on an XML-schema definition called evacML, which complies with the final recommendation of the W3C XML-Schema Definition (XSD) of May 2nd 2001. EvacML provides data constructs for geometrical primitives and functional objects like Wall, Stair, Door and Person. In addition to these, some abstract types are as so-called content models for a better structural organization of the data. All these special data constructs are achieved through a wide utilization of XML-Schema semantics such as type constraints, cardinality, simple and complex types as well as type derivation and redefinition. The basic hierarchical structure of evacML is depicted in Fig 2-1.

Fig 2-1 Basic Structure of evacML
2.1 Geometrical Data

The geometrical information in evacML is represented through a generic and flexible data construct. The *Geometry*-construct is used to represent geometrical information as combination of most common 2D-primitives; lines, polylines, circles and arcs. It is implemented as a *sequence group* containing one *choice group* of the geometrical elements (*PolyLine, LineSegment, Circle and Arc*), see Fig 2-2.

Using a combination of sequence and choice groups, the geometrical elements are syntactically allowed to occur in any order within a `<Geometry>`-tag in the XML file. This is important for the optimization of the translation process within a CAD-System.

![Fig 2-2 A generic Geometry-construct](image)

2.2 EvacML Objects

As matter of nature, non-geometrical information implicitly contained in the drawings, like the "meaning of symbols" (doors, stairs, egress routes, muster stations, persons etc.) and their characteristics (e.g. a person can be either a passenger or a crew member) are for the evacuation simulation just as important as the geometrical data defining boundaries of rooms and spaces.

EvacML objects contain geometrical and non-geometrical information to represent the data required for evacuation simulation at a high level of abstraction. A staircase for instance, is represented by its boundary walls, beginning steps (up steps), ending steps (down-steps) and normal steps.

![Fig 2-3 Abstract representation of a staircase and corresponding evacML object](image)

2.3 Interface from AutoCAD

AutoCAD is a popular CAD-System used in most areas of engineering design. In shipyards it is mostly employed for 2D-Drafting, i.e. elaboration of all kind of naval architectural drawings. For this reason AutoCAD was considered a good platform for implementing an evacML interface.

The preparation of the General Arrangement drawings to suit the needs of the AENEAS editor is a task, which must be carried out in any case. This tedious and time-consuming work is usually performed manually by means of repetitive tasks using standard functionality of the CAD-system. Thus, a need for some kind of automation to support the preparation work was identified.
A software-supported solution should provide mechanisms, ideally within a CAD-System, to "prepare" General Arrangement drawings for use in the AENEAS editor. Typical preparation work comprises following tasks:

- Remove unnecessary elements like texts, dimensions, title block etc.
- Associate drawing entities with cell types by means a specific color schema.
- Break/Interrupt lines representing walls in door areas and "mark" the door lines as such.
- Prepare text containing information about occupant numbers to be interpreted as such.

Several techniques have been combined to achieve an AutoCAD-based environment for processing the drawings. To begin with, custom objects were defined in AutoCAD by means of so-called Extended Entity Data (EED), Gerlach (1997). This is a mechanism which permits to programmatically attach additional data of different types (strings, real or integer values, binary data, etc.) to drawing entities.

Fig 2-4 A GUI-driven function to create stair objects in AutoCAD

Furthermore, special routines were implemented in Visual Lisp providing automatic and semi-automatic support for the preparation work. These routines include functionality to create and manipulate special objects such as walls, doors, stairs and persons. Fig 2-4 shows a GUI-driven function to create stair objects in AutoCAD.

Fig 2-5 Preparation of General Arrangement Drawings in AutoCAD
Once the preparation work is finished and all evacML objects have been defined, XML output can be generated according to the evacML Schema. The generated XML file can be imported into the AENEAS editor as described in the following section. Alternatively, the "model" can be transferred as DXF file. In this case, a separate DXF file for each deck is automatically generated.

2.4  Interface to AENEAS Editor

The AENEAS software package consists of two software programs. The AENEAS editor, which is used to model the floor plan and the AENEAS simulation which imports the generated project file and performs the evacuation simulation. This approach guarantees, that only the information needed for the simulation is used, making AENEAS spare the resources of the computer used and allowing for the evaluation of very big structures. This also is a reason for AENEAS extreme computational rate which is needed to perform many simulation runs, the basis for statistical analyses.

The AENEAS editor basically represents a simplistic CAD-program, allowing for a revision of the imported CAD drawings. Drawing elements can be edited, moved and new elements can be drawn. Thus, principally a "model" can be created in the editor from scratch, but the more reasonable approach consists in importing existing geometry information as DXF- as well as evacML-formatted files.

After having modeled the ship, the project is saved in a so called project-file, which is used for the simulation. While the element attributes (wall, door, stair, etc.) of DXF files are assigned according to their color, this data is already available in evacML formatted files.

Fig 2-6 Screenshot of the AENEAS Editor

After importing all geometrical data, the continuous elements are automatically projected on the discrete grid and the appropriate cell information is assigned to the cells. The user can alter the geometry, in order to test various layouts, especially the course of routes and the distribution of persons.

For the import of the evacML files into the AENEAS editor, several possibilities were considered. The classic approach consists in using either DOM (Document Object Model) or SAX (Simple API for XML) parser technologies, which are available in several programming languages (e.g. Java, C++, Python, FORTRAN).

SAX implements an event driven serial access protocol, in which the parser invokes callback methods defined by a registered handler whenever a new xml tag is encountered. DOM, on the other hand, utilizes an internal tree structure representation of the xml file, where elements are represented as nodes. Additionally, DOM provides mechanisms to navigate through the tree structure, read and
manipulate the data (i.e. modify or even remove or insert new data). Both technologies can be combined. In fact, some DOM implementations use a SAX parser internally.

Another possibility was a so-called "early binding" approach. Software systems which are used to develop XML schemas often allow for automatic generation of program code (e.g. Java or C++ classes) to import/create the appropriate XML files and at the same time to check the files conformity with the corresponding XML Schema. The names of the generated classes are based on the names of elements and types in the XML Schema, facilitating, thus, the development of data import/export routines. However, since the code is tightly constrained to the corresponding XML Schema, new code needs to be generated each time the Schema is modified.

Nevertheless, due to the fact, that currently the AENEAS editor only needs to import the evacML formatted files, without having to generate new ones, a very simple approach based on so called pointer lists was used. Similar to the DOM approach, the complete evacML file is internally saved in a complex tree structure of pointers. This approach is generic and consequently requires less adaptation to possible future changes of the evacML schema definition.

3 Conclusions and Outlook

An XML-based data interface for the transfer of CAD information to the AENEAS editor has been implemented. The main advantage with respect to the DXF interface is that the evacML data structure is especially tailored for evacuation simulation purposes. The evacML files contain only the required data without any additional and unnecessary "ballast", allowing for a more precise and correct mapping of geometry and attributes to the corresponding cell types in the target two-dimensional grid.

Furthermore, the evacML file format allows for a better overview of the elements contained in the files. In addition, problems concerning the scale of drawings and the alignment of various decks and stairs can be avoided, while more information about the analyzed ship can be added to the file.

The implementations in both AutoCAD and AENEAS editor were achieved using unconventional approaches for creating/reading XML data. This shows the flexibility of the chosen technology. The development of CAD processing routines to be used within AutoCAD obeys the natural requirement of potential users to prepare the General Arrangement drawings in the same environment they are familiar with. A similar approach can be implemented in any other CAD-system providing similar programming capabilities.

Further steps include the definitions of new evacML objects like cabins, evacuation routes, and zones, as well as the enhancement of existing elements with more attributes. Currently an implementation of evacML for the naval architectural package NAPA is under development. This will allow evacuation analysis and checking for compliance with existing IMO requirements at a very early stage of the design.

An important aspect of the new developed data exchange concept is its openness character. The specification of the defined data structure is intended to be publicly available through the internet.

Software vendors and developers are welcome to incorporate it in their software products and to participate in the enhancements and refinements in future releases.

References


MEYER-KÖNIG T. et al. (2002), Assessment and Analysis of Evacuation Processses on Passenger Ships by Microscopic Simulation, Pedestrian Evacuation Dynamics, Springer


Exploring the H-rep Ship Hull Modelling Concept

Herbert J. Koelman, SARC, Bussum/The Netherlands, Herbert@sarc.nl

Abstract

A novel ship hull modelling method, called hybrid representation (H-rep) was proposed some years ago. The aim of the H-rep method is to make ship hull design more intuitive, and therefore faster, more accurate and more predictable. The H-rep method has been implemented in a commercial software package, so in the course of time a lot of practical experience has been gained, while also new supporting functions have been designed and implemented. This paper serves two purposes. The first one is to present and discuss the additional functions and possibilities. The second purpose is to discuss practical projects performed with the current implementation of the H-rep method. After a brief overview on hull shape modelling methods in general, a short introduction of H-rep is given, and its merits and drawbacks are discussed. Special attention is paid to shaping curved surfaces by manipulating user-defined curves, and to Boolean operations. Subsequently general aspects of importing hull shapes and surface re-engineering are discussed, as well as the role H-rep can play in these fields. One particular application of re-engineering, photogrammetric reconstruction of a ship hull, is highlighted. The projects which are presented as practical examples are a simple yacht hull, a commercial ship hull and a complex superstructure of a mega yacht. In order to demonstrate the way of operation with a H-rep implementation, not only the final results of these projects are shown, but also crucial modelling steps. Finally conclusions are drawn, and some possible future developments are proposed.

1. Modelling methods for ship hulls

For the purpose of modelling the shape of a ship hull, many modelling methods are potentially applicable, and have been used in the course of the past decades. These models can be classified as follows:

Point-set models, where an object is formed by an unordered set of points in space, see a.o. Rusák (2003) and Gerritsen (2001).

Curve models. With these models unconnected spatial curves are used. In naval architectural applications curve models are mainly used for engineering analysis, not for design.

Surface models, which use a single surface or multiple unconnected surfaces. Currently the vast majority of commercial and academic ship design software suits use this method, where the NURBS surface is the most common geometrical modelling method, see a.o. Ventura et al. (1998) and Nowacki et al. (1995) for an introduction.

Wireframe models, where connected spatial curves are used, see a.o. Michelsen (1994) and Eida et al. (1999).

Solid models. For general CAD application, the most commonly used solid modelling methods are Constructive Solid Geometry (CSG), where an object is described in terms of Boolean combinations of simple shapes, and the Boundary Representation (B-rep), where an object is described in terms of the separation between ‘inner’ and ‘outer’. For naval architectural purposes Koelman (1999) proposed the Hybrid Representation (H-rep), which is essentially a B-rep extended with curved curves and surfaces.

This classification is given in the order of increasing topological structure. In information-theoretical sense only the last category is topological complete, the others are topologically incomplete (see Requicha (1980) and Koelman et al. (2001) for more details on the issue of completeness).
As far as the NURBS-based surface modelling method is concerned, achievements and results have been published in great numbers, both in commercial brochures as well as in technical and scientific papers, while some papers also make critical notes on this popular method (e.g. Hollister (199Xb) and Koelman (2002)). Because alternative methods are hardly considered in the present technical literature, it is the purpose of this paper to report on the merits and potential of the H-rep method. Currently, the only available H-rep implementation is the ‘Fairway™’ module from the commercial software package ‘PIAS™’, so all illustrations and examples in this paper are created with the aid of this program.

2. Properties of the H-rep concept

The data structure of H-rep was presented in Koelman et al. (2001). Essentially the following elements are involved:

- The topological elements of the conventional B-rep, which are the vertex, the edge and the face;
- the (curved) curve, which consists of a topologically ordered sequence of edges, and which is geometrically shaped by means of a NURBS;
- the polycurve, which is an ordered sequence of curves;
- the (curved) surface, which is topologically a collection of faces, bounded and intersected by curves. The surface derives its shape by means of transfinite interpolation of neighbouring curves.

From a practical point of view it is interesting to note that the B-rep elements are only used inside the system, in order to maintain topological integrity. The user only sees and works with curves and surfaces, so the look-and-feel of the User Interface (UI) is rather conventional, Fig.1. For the user the curve is the main modelling entity, while the surface automatically derives its curved shape from the curves in the vicinity. Shaping and manipulating the hull form is done by manipulating the curves, either by means of moving the curve vertices, or by shifting curve points and a subsequent fairing step. This is shown in Fig.2, which concerns a sailing yacht with a canoe body which is completely defined by eight curves: centerline contour, deck line, transom, the construction water line and four ordinates. The rightmost figure shows a distorted shape, by means of inboard bulging the foremost frame, which is translated into a dent in the entire fore ship region.

There is one potential danger in this free manipulation of curves and points, which is an inconsistency between topology and geometry. As mentioned, the system maintains the topological validity of the ship model, but a user might create some freak geometrical curve whose character
is not matched by the topological structure, in which case the result can be undesirable. So it is the responsibility of the user that the geometrical and topological structures are compatible. Fortunately this requires less effort; the use of sufficient and well-chosen curves suffices.

2.1. Merits of the H-rep method

Compared with conventional (surface-) modelling methods, the H-rep approach offers the following advantages:

- The user can work with an irregular network of connected curves, so orientation and location of curves are completely free, while also partial curves, which are not running over the complete hull surface, can be used.

- The user can work with curve vertices, or with points on the curve itself. Because the curves coincide with the surface this implies that the user can work with points on the surface. Furthermore, because each curve has its own independent geometric representation, the number of (NURBS) vertices can be different for each curve.

- At any time a curved surface is available, so at any time arbitrary additional curves can be generated. It is obvious that the most frequently generated curves will be those in orthogonal planes (transverse ordinates, waterlines and buttocks), but curves in oblique planes can also be generated. Another possibility is to define a loose curve somewhere in space, and project it upon the ship hull. After this action the resulting projection line is included in the H-rep data structure.

- If a ship designer chooses to use a couple of ordinates in the initial design phase, the Sectional Area Curve (SAC) can be utilized to work goal-directed towards a desired volume or Longitudinal Center of Buoyancy (LCB).

- Curves can geometrically be dependant on other curves. An example is a sheer strake at some fixed distance below the deck edge. If the sheer strake is predefined at that particular
distance from the deck edge at an early design stage, its shape will always follow the shape of the deck edge at the defined distance. This geometrical dependency is called ‘master-slave relation’.

- Boolean operations can be implemented, which allow the generation of an object as the Boolean sum, difference or intersection of two other objects. Application can e.g. be the addition of a bulbous bow to a hull, creation of bow thruster tunnels, adding tunnels, flaps and other appendages to a hull, and the composition of a semi-sub from basic elements.

The availability of a solid model is an advantage in this context, because Boolean operations can be performed with a single user action. If on the contrary only a surface model would be available, the intersection lines between surfaces may be determined automatically, but it would still require involvement from the user to trim the appropriate parts of the surface. and to combine resulting surfaces into the desired constellation.

Based on methods from Mäntylä (1986) and Hoffmann (1989) we recently implemented an experimental Boolean functionality into Fairway, from which an example is shown in Fig.3. It is a subject of future work to extend this functionality to efficient curved-surface processing.

![Image](image_url)

Fig.3: Boolean difference between a hull and a transverse rectangular cylinder

2.2. Drawbacks of the H-rep method

The ‘Fairway’ implementation of H-rep is now in use for some years with some tenths of shipyards, design offices, consultants and educational institutes. Based on the collectively gained experience also a list of drawbacks of the H-rep approach can be formulated:

- The development of a software program is an order of magnitude more complex than in case of a surface-based program. After all, for surface based software only the surface equations have to be implemented, which is rather a trivial task. On the other hand, a proper H-rep program requires implementation of topology (in the form of a genuine B-rep), curve geometry, 4-sided transfinite patches or surfaces, N-sided patches, curve fairing algorithms, plus all their interaction.

- In very early design stages the topological coherence was occasionally considered to be too rigid. The reason is that the proper topological structure is automatically maintained by the software by means of links between topological entities, but that implies that it is not possible to sketch some loose curves. And sketching of loose spatial curves gives a designer more the feeling of freedom.

Practically, this drawback can be resolved by leaving the H-rep structure out in early design stages, so the designer is completely free to sketch any curve he likes. At some stage.
when more coherence is desired, a method can be employed which converts a collection of unconnected curves into a solid H-rep model. Such methods exist, and will be discussed in subsection 3.2. However, when using this procedure it is the designer responsibility that the spatial curves indeed intersect each other (within some tolerance) where needed.

- It is not really a drawback, but in some cases the H-rep method does not offer a real advantage. Particularly hull shapes which are ‘simple’ in a mathematical sense, such as the one of Fig.2, can just as easily be made with a H-rep program as with a program based on a surface method. In this context the notion ‘simple’ means a surface with (a) equally distributed curvature, (b) which fits properly in a regular mesh and (c) without chines, or with chines which extent over the complete surface.

3. Surface re-engineering and import of hull shapes

Until now we have discussed the ship design aspects of the H-rep. However, in many occasions some form of shape information is already available, which can be used as the basis for further design or engineering activities. The capturing and subsequent processing of that information, which may exist in the form of drawings, computer files, scale models or actual size models, is the subject of this section.

3.1. Surface re-engineering in general

With software which is based on a surface model, surface re-engineering is in general a tedious job, as experienced by the author, and as reported a.o. by Hays et al. (1997) and Hollister (199Xa). One commonly applied approach is to force the user to choose only curves which coincide with the lines of some regular surface mesh. In this case the set of curves can, possibly after re-arrangement of some points, be converted into a NURBS surface. This method is only applicable for the simplest of shapes (i.e. those whose nature of shape fits a regular mesh).

Another approach is to show the available points or curves on screen, and let the user wrap one or more surfaces around them. In this case the hull is actually re-designed, which is time-consuming if some accuracy is required.

A third approach is to use mathematical optimisation techniques in order to automatically create a surface which fits a given set of points or curves as much as possible. A Genetic Algorithm-based version of such a technique was presented in Birmingham et al. (1998). However, up to the authors present knowledge, this approach is still limited to hull forms which are ‘simple’ in a mathematical sense.

The H-rep, on the other hand, is potentially more suitable for re-engineering purposes, because an important constituent is the curve, which can more easily match the curve or point data which are available as input in the re-engineering process. However, a prerequisite for the deployment of a H-rep method in re-engineering is the availability of a method which converts from a curve model to a solid model. This aspect will be discussed in the next subsection.

3.2. Solid model generation and data acquisition

The approach to generate a solid model from the lowest modelling entity, a point-set, is rather straightforward. If higher modelling entities than a point-set are available as re-engineering subject, initial steps can be skipped:

1. We assume that a point-set is available which is ordered along some lines. If only a point-cloud is available, it must be ordered, for example by manual intervention.
2. With standard techniques curves can be generated through each set of ordered points.
3. Based on a geometric search, all intersection points between all curves are determined, and maintained in a list, so effectively a wireframe model is created. In order to compensate for round-off errors in the source data set, the curves are assumed to intersect if they pass
each other within some user-supplied tolerance.
4. With a method as presented in e.g. Inoue et al. (2001) or Koelman et al. (2002) the wireframe can be converted into a B-rep solid model. As discussed in the referred papers the conversion methods have some limitations, but in the majority of practical naval architectural cases those limitations play no role or can be avoided.
5. In combination with the ordering information of the point-set the B-rep solid model can be converted into a H-rep model.

From the viewpoint of software design the conversion method is implemented in a pre-processor. Communication between the pre-processor and the actual ship design program is performed by means of a couple of ASCII files, which are called Solid eXchange File (SXF) and Curve eXchange File (CXF), whose formats have been made public. This allows the pre-processor to be fed with modelling entities which are ‘higher’ than an ordered point-set, such as curves, a wireframe or a solid. The system design of the pre-processor is sketched in Fig.4

3.2.1. File import

Based on the methodology as discussed in the previous paragraph, import facilities have been implemented for DXF 3D curves (subtypes line, polyline and NURBS), IGES 3D curves (subtype NURBS) and IGES surfaces (subtypes NURBS and parametric surface). Fig.5 shows an example of import from a DXF file which contains polylines, and conversion to a solid model.
3.2.2. Photogrammetric measurement of a ship hull

The possibility to convert an ordered point-set into a H-rep, and to process it further with H-rep software opens the door to capture data from scale models or real vessels by means of a data-collection method like photogrammetry. Basically, with photogrammetry multiple photos or stereo photos from an object are taken, and based on classical projection theory the three-dimensional co-ordinates of those points which occur on more than one photo are determined. Photogrammetry was first applied in the field of aerial and terrestrial surveillance, but later on it has been found useful also in close-range applications for the measurement of medical, archeological, architectural and industrial objects, see e.g. Kraus (2000) for more details.

In order to be able to measure real ships, we have equipped our software with a module for photogrammetry, which in short works as follows:

1. Markers are placed on the hull. A variety of markers can be used, such as stickers, magnets and paint dots. Naturally distinguishable points, such as corners or joints between welds, can also be used as markers.
2. For a limited number of markers (e.g. 4 - 10) the world co-ordinates must be established by some conventional method.
3. Photos are taken from different positions, where each marker must be visible on at least two photos. For taking the photos a regular camera can be used, but in order to achieve a high accuracy, we used a special, so-called ‘metric’, camera which possesses no or only a minor lens distortion.
4. Each photo is shown on screen, and each marker on each photo must be pointed to with the cursor, so the processing software is able to determine the co-ordinates (in the co-ordinate system of the photo) of that point. Should a sticker with a unique pattern be applied as a marker, a pattern recognition technique may be used to identify the markers automatically.
5. With appropriate techniques, see e.g. Kraus (2000) and Atkinson (2001) for details, the 3D positions of the camera and the markers are calculated.

Fig.6: Photos, markers and connections of the ‘7 Provinciën’ reconstruction
6. Besides identifying the markers, in the fourth step the sequences of markers along (arbitrary) lines have been indicated by the operator. This implies that an ordered point-set is available, which can be converted into a H-rep model with the algorithms as described in subsection 3.2.

7. If desired some post-processing, such as smoothing or generating additional curves, can be applied within the standard design environment of the H-rep implementation.

As a practical example we present the measurement of a the ‘7 Provinciën’. This is an actual-size reconstruction of a Ship of the Line, which was originally built in 1665, and measures (LxBxH) 46.14x12.14x4.74 m. The ship can be visited in Lelystad, The Netherlands, and more information can be found at website www.bataviawerf.nl.

Fig.6 shows a constellation of photos, plus the already identified markers and the connections between the markers, while Fig.7 shows the resulting H-rep model.

![Fig.7: CAD model of the ‘7 Provinciën’, as obtained by photogrammetry](image)

### 3.3. Shape re-use

Until now we have discussed re-engineering and data acquisition, but also another use of the collected shape data can be made, which is re-use or re-design. Of course all available design actions can be performed once the ship model is available in a H-rep environment, but it can also be appropriate to convert the data into an other model, where manipulation actions can be applied. An example is explored in a current research project, where the goal is to extend the available geometrical and topological information with some kind of meaningful semantic information. The idea is that with the aid of the semantic layer design modifications can be applied very effectively. More details are discussed in Koelman et al. (2003) from which Fig.8 was extracted, which shows an experimental point-set model.

![Fig.8: Point-set representation of a frigate-like hull form](image)
4. Example projects

In order to demonstrate the practical potential of the H-rep approach, in this section we present two actual design projects which have recently been performed with H-rep-based software. The first example concerns a high-speed monohull, which is presented to demonstrate steps in the design process of a conventional hull. The second example concerns the superstructure of a mega yacht, which is shown to demonstrate that application of the H-rep model is not necessarily limited to ship hulls, but also extends towards unconventional shapes.

4.1. High-speed monohull

The first example we present is a high-speed 20.72 x 4.40 x 2.80 m monohull, equipped with a propeller tunnel. This vessel is currently under construction at 'Engelaer Scheepsbouw' of Beneden-Leeuwen, The Netherlands. This example is shown to demonstrate the flexibility of the H-rep model, because each design step is purely driven by notions and desires of the ship designer; not a single action had to be executed because the software insisted to do so. The design stages are presented in Fig.9.

![Design stages of high-speed monohull](image)

The initial design information consisted of a hand-made sketch of a lines plan with a couple of ordinates (with an idea of the propeller tunnel) and the required volume and LCB. The first step (a) was to measure the centerline contour and the deckline from the sketch, define them in the software and fair those curves. It is important to fair all curves up to full production accuracy; because so few curves are available at this design stage it is very easy to do, while in a later design stage one will benefit from the fact that these curves are already completed. The
straight line at abt. $L_{pp}/2$ is a dummy ordinate, it will be shaped at a later stage. Subsequently a few points on ordinates of the design sketches have been measured or digitized and defined in the computer model. Knuckle points have been marked as such, and the sprayrail chines have been defined, just by pointing at the knuckle points in the subsequent ordinates. The same applies to the chine just below the deck edge. This stage is shown in Fig.9(b), which also shows that at this early stage the deck has already been included in the model. The reason is that a requirement for this vessel was a deck camber of $1/50^{th}$ of the vessel’s breadth, combined with a pre-defined deck shape at CL. To model this, the deck at side and at CL have been defined to have a ‘master-slave’ relation (which was discussed in subsection 2.1), with the deck at CL being the master and the deck at side being the slave. So the geometrical relation between deck at side and at CL was integrally taken into account, and the height of the deck at side is permanently derived from the deck at CL. Fig.9(c) shows a more mature stage, already with the cone-segment as stem-plate. However, it does still not have a propeller tunnel.

It was a design requirement that the propeller tunnel should be circularly shaped at cross section, with a specified but non-constant radius. In order to model this, the topline of the tunnel was projected on the tunnelless hull, and was given the desired shape by means of interactive manipulation. Subsequently, circular arcs have been generated and intersected with the existing bottom. Finally, the operator connected all intersection points between bottom and arcs, thus forming the tunnel chine. The result is shown in Fig.9(d), and the resulting lines plan in Fig.10. where it is interesting to note the curves resulting from intersection with angled planes (e.g. at a longitudinal location of abt. 18 m). Those curves have not been included for design purposes, but as aids for the engineering process, which had already commenced when the finishing touches were put to the design. Fig.11 shows the hull surface.

Fig.10: Lines plan of high-speed monohull
4.2. Superstructure of a mega yacht

Our H-rep application is more or less designed to work with curves, because they match the naval architectural notions of ordinates, waterlines and buttocks so well. However, also other complex sculptured objects than ship hulls may be handled well with an H-rep application, especially if many local constraints apply, such as prescribed diameters, steps, knuckles, round-offs or notches. As an example we present the superstructure of a 76 x 12.80 m motor yacht, which was recently engineered by the company ‘Ankerbeer’ of Veendam, Groningen, The Netherlands. Because the superstructure is asymmetrical, first the SB part was modelled, and copied to PS. For the symmetrical parts the PS curves inherited their shape from the SB curves by means of master-slave relations. Only the actual asymmetrical parts have been modelled by distinct curves on SB and PS.

The purpose of modelling this superstructure was twofold. In the first place it was required to serve as a shape database for subsequent engineering application, and secondly shell plate expansions have been derived on the basis of this model. In order to fulfill these requirements a tolerance of less than 0.5 mm was necessary.

The wireframe representation, which shows all curves which have actually been fixed or manipulated in the design process, is shown in Fig.12, while the resulting surface is shown in Fig.13.
An interesting aspect of this project is the data exchange method between the shape repository (which is the H-rep application) and the engineering application, where all construction elements such as girders, stiffeners and pillars are modelled. Conventionally, the shape data are transferred to the engineering application by means of some kind of data file, but that approach might have two disadvantages. In the first place the transferred data are static, so modified shape data are not automatically processed in the engineering application. Secondly, in case the engineering application is curve-based, it is almost impossible to anticipate on the amount and location of all curves that might be necessary in the engineering process. A practical example is a longitudinal engine girder, whose location can not be foreseen in an early design stage. At the time the location is finally known, additional work is required to transfer the shape at that particular location from the shape repository, with a data file as a carrier, to the engineering application.

In order to improve this procedure, we have developed the so-called ‘hull-server’. Essentially, the hull-server is an instance of the PIAS/Fairway program, without the standard UI. Instead it is equipped with a (Windows-based) communication protocol which allows any external program to request the shape at a particular section or location. In no time the hull-server replies with that requested shape.

Of course this approach is only useful if the engineering application is equipped with a ‘hull-client’, which can interact with our hull-server. Fortunately, quite recently the ‘NUPAS™’ engineering program, by ‘Numeriek Centrum’ of Groningen, The Netherlands, has been equipped with such a hull-client, and because the superstructure under consideration was both defined in PIAS/Fairway and engineered with NUPAS, this mechanism could be applied successfully for this project. Fig.14 shows NUPAS’ construction model, based on Fairway’s shape, as communicated over the hull-server ↔ hull-client interface.

5. Conclusions and future developments

In this paper we have summarized the potential of the H-rep approach, in the field of modelling of objects ranging from conventional ship hulls to complex sculptured compositions. It was demonstrated that for every activity in the design and engineering phase the H-rep structure can offer benefits. Of course each method can be enhanced with additional functions, and currently the following extensions are under consideration:

- As motivated in subsection 2.1, at this moment Boolean operations are implemented, but it would speed up the ship design process if those operations would be more efficient in a curved-surface environment. Implementing of fast Boolean operations is on our list.
- The region of modification is implicitly fixed with the present H-rep model. Modification of one curve does change the surfaces in the neighbourhood of that curve, so the curve
Fig.14: Construction model of the mega yacht superstructure

 spacings ultimately determine the extent of the modification. It could be desirable to explicitly specify the modification region, and currently we are preliminary investigating methods which may provide such a function.
- Extend the software with a pre-processor which generates an initial hullform, either automatically, or by means of user commands in some sort. The method as discussed in subchapter 3.3 is a candidate for such a function.

References


EIDA, Y.; NISHIKIDO, Y.; SASAKI, Y. (1999), Development of Practical 3-D Lines Fairing System for Ship Hull Form (MELFAS) and its Application to CIMS, 10th Int. Conf. Computer Applications in Shipbuilding ICCAS, MIT, Cambridge

GERRITSEN, B. (2001), Using Weighted Alpha Complexes in Subsurface Modelling, Ph.D. Thesis, Delft University of Technology


KOELMAN, H.J.; HORVÁTH, I.; AALBERS, A. (2001), Hybrid Representation of the Shape


KRAUS, K. (2000), Photogrammetry Vol. 1 and 2, Dümmler, Köln


Shape Optimization Strategies for Complex Applications in Computational Fluid Dynamics

Regis Duvigneau, Michel Visonneau, Laboratoire de Mecanique des Fluides, Ecole Centrale de Nantes, Regis.Duvigneau@ec-nantes.fr, Michel.Visonneau@ec-nantes.fr

Abstract

A shape optimization tool is developed for complex applications in Computational Fluid Dynamics. Reynolds-averaged Navier-Stokes equations are solved on unstructured grids using near-wall low-Reynolds number turbulence models. The implementation of such a complicated flow solver in a shape optimization procedure is tedious, because all steps in the design loop should be adapted to the complex numerical framework, for instance the optimization strategy and the mesh update. Moreover, some noisy errors are arising, related to the partial convergence, the discretization and the turbulence modeling. These points are analyzed in this paper, in the particular framework of hull shape optimization at full scale.

1 Introduction

Thanks to the progress in Computational Fluid Dynamics (CFD) during the last years, flow solvers are now integrated in fully automated design optimization procedures. In order to use these shape optimization tools in an industrial framework to solve realistic problems, some conditions are required.

The physical and geometrical configurations encountered in these industrial applications makes necessary the recourse to viscous flow solvers based on sophisticated turbulence modeling, dealing with realistic geometries. In this framework, the naive use of a standardized toolbox optimization software connected to a flow solver and an automated grid generator cannot be a sensible strategy, since the peculiarities of the flow solver should be taken into account. Otherwise, some limitations may be quickly encountered. For instance, the constraints on the grid \( y^+ = O(1) \) near the wall linked to the use of near-wall turbulence models will have serious consequences on the mesh update procedure. If a particular strategy taking into account the high stretching of the volumes near the wall is not included, the mesh update will fail. This observation explains why the recourse to an automatic grid generation software may not be wise in such a context. An other reason is related to the mandatory use of parallelization strategies, such as domain decomposition, as soon as three-dimensional problems are considered. The mesh update, as well as the parameterization of the shape, should be adapted to this multi-block partition to work within each block independently and to send information for updating the overall domain. Otherwise, the parallelization becomes useless. Concerning the optimization methods, the high computational costs implied by three-dimensional calculations, as well as the possible occurrence of numerical noises of various origins during the evaluations, should be taken into account for the choice of an optimization strategy. Finally, if different connected softwares are employed in the design loop, some practical difficulties may arise, when distant computers are involved in the optimization process. All these remarks justify the development of an optimization procedure, in which all numerical tools are adapted to the flow solver and integrated into a single code.

This paper is devoted to the study of the optimization strategies which could be used to fulfill the conditions described previously. The difficulties arising from the use of a complex flow solver in the design procedure are particularly explored.

2 The ISIS flow solver

The flow solver ISIS, developed in our laboratory is included in the design procedure. It solves the
incompressible Reynolds-Averaged Navier-Stokes Equations (RANSE) on unstructured grids, with a strongly conservative formulation. The discretization scheme uses a finite-volume method, generalized to unstructured meshes composed of arbitrary volume shapes. Thus, calculations involving complex grids may be taken into account. The flow variables are stored at the center of the control volumes, surface and volume integrals being evaluated using second order accurate approximations. The pressure-velocity coupling is performed by a SIMPLE-like algorithm. Several near-wall low-Reynolds number turbulence models, ranging from one-equation Spalart-Allmaras model, two-equation $k – \omega$ closures, to a full Reynolds stress transport $R_{ij} – \omega$ model are implemented in the flow solver. Free-surface calculations are also possible through surface-fitting or surface-capturing methods.

3 Optimization strategies

Several strategies may be considered to drive the search of the optimal shape. In order to reduce the computational costs, one may propose to use gradient-based methods, characterized by a super-linear convergence and an independence from the number of design parameters. However, the implementation of this strategy in a complex numerical framework is particularly tedious. Indeed, the evaluation of the derivatives of the cost function with respect to the design parameters relies on the differentiation of the flow solver, through the resolution of an adjoint equation (Anderson and Venkataraman (1999)). Thus, this approach is still limited to mildly complex problems, involving typically unviscid flows. Two alternative approaches may be considered to evaluate the derivatives: the automatic differentiation softwares and an incomplete evaluation of the gradient, neglecting the modifications of the flow. But these methods are still limited by the complexity of flow solver in the first case and the range of application in the second case. Moreover, gradient-based methods have several drawbacks. They can only consider one optimization criterion and are particularly sensitive to the numerical noise arising from the use of complex numerical methods and generating spurious local optima. Considering all these remarks, the gradient-based strategies are rejected in this framework. This choice has crucial consequences, since all other approaches are submitted to the “curse of dimensionality”, the number of evaluations required being at least proportional to the dimension of the problem. Thus, the parameterization of the shapes should involve as few design variables as possible.

Derivative-free deterministic algorithms, such as the simplex method (Duvigneau and Visonneau (2001a)) based only on function values comparisons, may be easily implemented in the design procedure, since the flow solver is considered as a “black box”, yielding both flexibility and robustness to solve complex optimization problems. Moreover, this kind of methods is less sensitive to the noisy errors. Nevertheless, they are limited to perform local optimizations including one criterion.

Therefore, stochastic methods, such as Genetic Algorithms (GAs), are particularly appealing. Indeed, they have the capability to perform global optimizations and solve multi-objective problems. Furthermore, their robustness to solve multi-modal problems has been shown several times in the past. As derivative-free methods, their implementation in the particular framework of complex CFD is quite easy (Duvigneau and Visonneau (2001b)). However, this strategy is particularly expensive, since the number of evaluations is about ten times higher than the number required by deterministic methods. These costs make the use of genetic algorithms unrealistic for three-dimensional problems.

These observations look quite pessimistic. Although the ultimate optimization strategy does not exist yet, one may propose a realistic way, taking the best parts of the different strategies and using approximation methods to reduce the computational time. Genetic algorithms have two main drawbacks: the number of evaluations and the low local convergence rate. To reduce the number of evaluations through the flow solver, one may use the knowledge already collected
on the problem to build local approximations replacing some expensive evaluations (Giotis and Giannakoglou (1999)). More precisely, all exact evaluations through the flow solver are stored in a database, whose entries are used to train Artificial Neural Networks (ANNs). They are employed to perform inexact pre-evaluations of the shapes at each generation, only the most promising shapes according to the ANNs pre-evaluations being exactly evaluated through the flow solver. In that way, the number of expensive evaluations may be significantly reduced, without loosing the interesting capabilities of genetic algorithms. Then, genetic algorithms are only expected to determine the overall characteristics of the optimal shape and the final local search is performed by a deterministic algorithm, which can quickly precisely define the shape.

This strategy, using hybrid genetic algorithms and neural networks, has been successfully applied to optimize the shape of a three-dimensional wing at the incidence 8°, at a location close to the salmon, with respect to the lift (Duvigneau and Visonnneau (2002a)). For this case, an unstructured mesh of about 800 000 cells was used, for a Reynolds number \( Re = 2 \cdot 10^6 \), the near-wall low-Reynolds number SST \( k-\omega \) model being employed. The shape is described by 12 design parameters. Fig.1 representing the evolution of the cost function during the design procedure illustrates the strategy.

![Fig.1: Evolution of the cost function](image)

The calculations are performed using a multi-block parallelization technique, on sixty R14000 processors. As seen, very complex problems may be solved using this strategy. Even if the costs remain quite large, the CPU time has been reduced more than five times with respect to a simple genetic algorithm. Although this approach has still some limitations, such as the number of design variables involved, it seems to be a realistic and promising way.

4 Shape parameterization and mesh update

When three-dimensional unstructured meshes are employed, the parameterization as well as the mesh update during the design process are difficult tasks. Indeed, a curvilinear representation of the shape is not available. Thus, the use of B-spline net representations for instance is tedious. The nodes coordinates on the shape and connectivities are the only knowledge which could be used. Therefore, a Free Form Deformation (FFD) technique (Samareh (1999)) is used to control the shape perturbations during the design process. It consists in first embedding the object to be deformed in a box, and then modifying the metrics of the space in the box and the object inside by deforming the box, rather than modifying the object itself. In that manner, the shape of the
object can be modified without even identifying its nature. This powerful approach provides an easy-to-use solution, requiring only a few control points to control the deformation of the shape. However, this general technique may be inadequate to deal with a particular and specialized problem.

The automated mesh update is problematic, when unstructured grids using a near-wall turbulence modeling formulation is employed, since the grid volumes are highly stretched near the wall to fulfill the condition $y^+ \leq 1$. The spring analogy (Farhat et al. (1998)) previously used to deform two-dimensional meshes (Duvigneau and Visonneau (2001a)), may hardly be employed for three-dimensional problems. Therefore, one may recommend to smoothly deform the shape and the mesh by the FFD method at the same time, since the FFD approach relies on a spatial deformation. This strategy may be carried out for two-dimensional as well as three-dimensional grids and is obviously adequate for multi-block calculations. However, the quality of the resulting grid is not ensured. An example of deformation of a hybrid grid around the NACA 0012 airfoil is provided by the Figs.2 to 4, using the spring analogy and the FFD method. As seen, the FFD method provides a smoother deformation, but the orthogonality of the grid near the wall is not maintained, contrary to the spring analogy. Nevertheless, the FFD method gives satisfactory results, when small perturbations are taken into account.

Figure 2: Initial mesh

Fig.3: Spring analogy

Fig.4: Free Form Deformation
5 Influence of turbulence modeling

The previous sections provide some solutions to solve optimization problems involving a complex flow solver. However, one should validate the results obtained and quantify the influence of the numerical and physical parameters on the design procedure. Indeed, some quantities have an influence on the resolution of the flow and thus on the optimal search. For instance the grid size, the stopping criteria, the discretization schemes, turbulence modeling, generate noisy errors during the evaluations (Madsen (2000)). A precise study concerning the role played by turbulence closures for hull shape optimization was performed in (Dubrul and Visonneau (2002b)).

The flow around the KVLCC2 tanker is computed at full scale ($Re = 10^6$), using a structured mesh of about 400 000 volumes. Two turbulence models are tested: The eddy-viscosity SST $k – \omega$ model of Menter and a second-order closure $R_{ij} – \omega$ model. The goal of the optimization is to homogenize the longitudinal velocity field at the location of the propeller, modifying the stern of the ship described by 6 design variables. It was shown that the two models give different predictions of the wake for the initial shape, the longitudinal vortex generated being intensified by the second order closure (Deng and Visonneau (1999)). However, these differences are less important at full scale than at model scale. We intend to quantify the influence of the turbulence modeling during the optimization process.

Figs.5 and 6 show the isowakes for the initial and final shapes. As seen, the optimal shapes are characterized by a far more intense longitudinal vortex homogenizing the flow and close to the vortices observed at model scale. This evolution during the design process enforces the influence of the turbulence modeling. Figs.7 and 8 represent the streamlines close to the wall. One may observe that the topology of the flow has changed and is finally similar to model scale flows. Therefore, some differences between the models appear, even at full scale. Looking at the optimized shapes, Fig.9, similar trends are obtained, although some differences are noticed due to the fact that the second order closure provides a more intense prediction of the flow. Therefore, lower modifications may generate a more intense vortex. One may notice that the final shapes found correspond to U-shaped hull. The evolution of the cost function, given in Fig.10, shows a reduction of 70% of the velocity mean deviation at the propeller location. The reduction is lower for the second order closure. This phenomena is maybe due to the presence of more complex turbulence structures in the wake, which are not described by the linear eddy-viscosity model.

Finally, this example underlines the influence of turbulence modeling in the framework of hull shape optimization. Although its role was a priori less important at full scale, it was shown that the choice of the turbulence model may have a crucial influence, when a sharp optimization process is performed.

Fig.5: Isowakes for the SST $k – \omega$ model  Fig.6: Isowakes for the $R_{ij} – \omega$ model
Fig. 7: Streamlines for the SST $k$-$\omega$ model; left: Initial; right: Final

Fig. 8: Streamlines for the $R_{ij}$-$\omega$ model, left: Initial; right: Final

Fig. 9: Initial and final shapes
6 Conclusion

This study explored different strategies to solve shape optimization problems involving complex flow solvers. As seen, the choice of the optimizer strongly depends on the framework and has crucial consequences. Genetic algorithms have several advantages over deterministic approaches. It was shown that their use is now realistic, providing that acceleration techniques are employed, such as inexact pre-evaluations through artificial neural networks. Some practical difficulties still exist, for instance the mesh update for three-dimensional problems. However, the main limitation may be the influence of turbulence modeling on the design process for some problems, since this modeling error cannot be controled, contrary to other numerical parameters, such as grid size or partial convergence.

References


Multidisciplinary Design Optimisation and Robust Design Approaches
Applied to the Concurrent Design

Alessandro Giassi, Sirehna R&D company, Nantes/France, alessandro.giassi@irccyn.ec-nantes.fr
Jean-Jacques Maisonneuve, Sirehna R&D company, Nantes/France
Fouad Bennis, IRCCYN Laboratory ECN, Nantes/France

Abstract

The paper presents the implementation of MDO approaches in a concurrent engineering context. These subjects in the naval domain are still weakly developed, but the interest in this field is growing. Besides, the paper describes a quite innovative procedure based on robust design technique: MORDACE (Multidisciplinary Optimisation and Robust Design Approaches Applied to Concurrent Engineering). A roll stabiliser fin optimisation is presented as an example of this methodology.

1. Introduction

The paper will describe an innovative multidisciplinary optimisation method applied to a ship design problem. The subject enters into the wider framework of concurrent engineering, Winner et al. (1988), which is an essential element in the modern product design process. The competitiveness of the market and the required special knowledge and tools, push companies to distribute the design work generating fragmentation. The multiplicity of actors may arise within the same company, e.g. when a company is composed of several offices and work is organised on several sites (extended enterprises). Sometimes, multiplicity arises from collaboration with other companies (virtual enterprises) or subcontractors.

Currently the key to the collaborative and distant design process lies in the use of digital mock-up and product data management software (PDM), Meinecke and Férus (2002). These tools progressively lead to a complete integration of product design. In this context, the paper proposes a multidisciplinary optimisation method, which allows to manage collaborative design without requiring high level of integration. The hydrodynamic and structural optimisation of a roll stabiliser fin offered us the opportunity to test this new design approach. In order to compare our method, we also performed the optimisation process with two classical multidisciplinary methods.

2. Case study: optimisation of a roll stabiliser fin

The proposed methodology proposed in this paper was initially developed using a mathematical test function. These mathematical tests allowed us to define our algorithm and to compare it with classical approaches. Then, we applied our method to a realistic case of multidisciplinary optimisation to validate results.

Roll stabilising fins are usually mounted on rotating stocks at the turn of the bilge, near the middle of the ship, Fig.1. The angle of incidence of the fins is continually adjusted by a control system, which is sensitive to the rolling motion of the ship. The fins develop lift forces, which exert roll moments about the centre of gravity of the ship opposed to the wave excited moment, and the roll motion is reduced, Lloyd (1989). The roll stabiliser fin to optimise was based on a real application. The fin had NACA0015 symmetrical sections, Crane et al. (1989). Fig.2 shows the main dimensions, i.e. the root and tip chords c, and c, and the span b and aspect ratio a:

\[ a = \frac{2b}{c} = \frac{4b}{(c+\epsilon)} = 2.5 \]

where \( c \) was the mean chord

\( (1) \)

The optimisation problem was set as multidisciplinary: we progressively modified the fin external shape and its internal structure in order to enhance both hydrodynamic and structural performances. Thus, we had three objective functions concerned hydrodynamic discipline and two objective
functions defined the structural optimisation. In addition, the optimisation had to respect some hydrodynamic and structural constraints.

Fig. 1: Roll stabiliser fin

Fig. 2: Original fin shape

Fig. 3: Two-discipline system

Table 1: Roll stabiliser fin problem: variables and design functions

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>Design variables of fin geometry outside of structural box</td>
</tr>
<tr>
<td>$x_2$</td>
<td>Design variables of structural box</td>
</tr>
<tr>
<td>$x$</td>
<td>Design variables common to both disciplines</td>
</tr>
<tr>
<td>$y_{12}$</td>
<td>Hydrodynamic loads</td>
</tr>
<tr>
<td>$y_{12}$</td>
<td>Displacements that alter the hydrodynamics shape</td>
</tr>
<tr>
<td>$g_1$</td>
<td>Angle of incidence limit; cavitation limit</td>
</tr>
<tr>
<td>$g_2$</td>
<td>Stress limit</td>
</tr>
<tr>
<td>$f_1$</td>
<td>Maximum lift; minimum drag; minimum sensitivity to angle of incidence</td>
</tr>
<tr>
<td>$f_2$</td>
<td>Minimum structural weight, minimum displacement</td>
</tr>
</tbody>
</table>

Fig. 3 shows this two-discipline system. Table 1 identifies what the variables and functions in Fig. 3 represent for the roll stabiliser fin optimisation. The vectors $x_1$, $x_2$ and $x$ were sets of design variables. Vector $x_1$ defined the external shape of roll stabiliser fin, $x_2$ allowed to draw the internal structure and $x$ contained design variables needed by both disciplines. Subsection 3.3 gives details of the fin parameterisation. The vectors $y_{ij}$ were the coupling functions. The function $y_{ij}$ was computed in discipline $i$ and it was used as input for discipline $j$. The vectors $g_i$ were design constraint functions. We used them to avoid unfeasible and unacceptable designs. The vector $f_i$ contained the objective functions, i.e. performances that we aimed to enhance.

3. The parametric model and the evaluation of performances

3.1. Hydrodynamic evaluations

Each design was defined by a new set of design variables and parametrically drawn using a CAD application, CATIA V5. Concerning discipline 1 calculations, we created a script that launched CATIA and automatically generated the fin drawing using an ASCII file containing design variables as input. When a fin drawing was completed, the same script generated the points that allowed the hydrodynamic mesh to be created. The mesh was made-up of 1260 structured panels. We used the potential flow code REVA to estimate the fin hydrodynamic performances. This application calculated hydrodynamic loads, i.e. pressure distribution on fin's surface, and evaluated lift and drag. Fig. 4 describes the calculation loop. The process ran entirely in batch mode starting from the ASCII file containing design variable values.
The hydrodynamic optimisation process aimed to minimise drag and to maximise lift. The third objective function was to minimise the effect of variation in angle of incidence. In fact, we considered ship pitch motion to be an external noise, which modifies the angle of incidence of roll stabiliser fin. Control system takes into account the pitch motion and corrects the angle of incidence. Thus, we tried to find a robust design in order to reduce as much as possible the control system interventions. REVA evaluated hydrodynamic performances of each design varying the angle of incidence $\alpha_i$. Thus, the three objective functions of hydrodynamic optimisation were formalised as follows.

Maximise:

$$f_{11} (x_1, x_2) = \text{Lift} (x_1, x_2) = \frac{1}{2} \sum_{i=1}^{5} \text{Lift} (x_1, x_2)_{a_i} [N] \quad \text{where} \quad a_i = 2.5 + 1.25 (i-1) [\text{deg}], \ i=1, ..., (2)$$

Minimise:

$$f_{12} (x_1, x_2) = \text{Drag} (x_1, x_2) = \frac{1}{5} \sum_{i=1}^{5} \text{Drag} (x_1, x_2)_{a_i} [N] \quad (3)$$

Minimise:

$$f_{13} (x_1, x_2) = \sigma_j (x_1, x_2) = \frac{1}{2} \sum_{j=1}^{5} \sigma_j (x_1, x_2) \quad (4)$$

for $j=1$: $\sigma_j (x_1, x_2) = [\frac{1}{2} \sum_{j=1}^{5} (\text{Lift} (x_1, x_2)_{a_i} - \text{Lift} (x_1, x_2) \}^2 ]^{0.5} / \text{Lift} \text{ of original design} \quad (5)$

Eqs.(2) to (4) show that the first two objective functions were the mean of performances values and that the third objective function was a sum of normalised standard deviations. Eq.(6) shows that we normalised the standard deviations of performance values by dividing them by the performance values of the original design. In this manner, we were able to sum coherent\(^1\) values in Eq.(4).

The hydrodynamic optimisation was subject to constraint functions for the cavitation limit and the angle of incidence. For the cavitation constraint, the minimum permissible pressure on fin's surface was set at 17000 Pa. For the angle of attack limit, we evaluated, Whicker and Fehlnier (1958):

$$\sigma_{\text{stall}} = (180/\pi) (1.05 - 0.445a + 0.075a^2) \ [\text{deg}] \quad \text{for} \ a < 3 \quad (6)$$

$a$ follows Eq.(1). Designs respected this constraint for $\alpha_{\text{stall}} > 7.5^\circ$, the upper limit of imposed incidence angle variation.

3.2. Structural evaluations

We draw the fin structure using the CAD application CATIA V5. As before, we used the CATIA scripting technology. We created a script that launched CATIA and automatically generated the drawing of the fin structure using an ASCII file containing design variables as input. When the drawing was completed, the same script launched finite-element calculations to estimate structural performance values. The finite-element code was the Generative Part Structural Analysis, an internal CATIA V5 tool. This application automatically generated a non-structured mesh made-up of 29592

\(^1\) We did not use relative standard deviation because it penalised low drag designs.
linear tetrahedrons. Thus, the calculation process was entirely performed by CATIA in batch mode starting from the ASCII file containing design variable values, Fig.5.

![Diagram of structural calculation loop](image)

Fig.5: Structural calculation loop

The optimisation process aimed to minimise the structural weight and to maximise the structural rigidity, i.e. to minimise the maximum node displacement. We imposed a stress constraint. For the chosen material steel, the maximum accepted stress was 200 MPa. The structural optimisation problem was formalised as:

<table>
<thead>
<tr>
<th>Find</th>
<th>Design Variables</th>
<th>$x, x_2 = (x_1, ..., x_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfy</td>
<td>Constraint</td>
<td>$g_{22}(x, x_2) = \text{Maximum Stress} = \sigma_{\text{max}}(x, x_2) &lt; 200 \text{ MPa}$</td>
</tr>
<tr>
<td>Minimise</td>
<td>Objective Functions</td>
<td>$f_j(x, x_2) = \text{Structural Weight}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_{22}(x, x_2) = \text{Maximum Node Displacement}$</td>
</tr>
</tbody>
</table>

Concerning the coupling functions, we only took into account $y_{12}$, i.e. the hydrodynamic loads calculated by the flow solver. We considered $y_{12}$ to be too small to alter the hydrodynamic shape. In fact, the maximum displacement of the original fin shape was 13.5 mm (0.5% of fin span) besides, we tried to minimise it. Thus, there were no iterative calculations of performance values.

### 3.3. Parameterisation

A geometric definition of the problem must be made to begin implementing the optimisation process. The optimisation algorithm must be able to find a relationship between the design variable variations and the evolution of performance values. Thus, a controlled modification of the original fin design was required. The two disciplines involved in this optimisation process demanded two different parametric models, as they analysed different behaviours of roll stabiliser fins. The hydrodynamic discipline aimed to streamline the surface, the structural discipline to modify parameters that defined the internal geometry (thickness values, distribution of stiffener, section areas, etc.).

The choice of parameters is vital since it defines the mathematical model of the optimisation problem. The generated solutions will largely depend on the chosen parameters, since they define the research space.

Another important aspect of parameterisation step is the definition of parameter boundaries. In order to extend the design space around the original fin design and cover the feasible domain, parameters should have wide boundaries. Unfortunately, a parameterisation with very wide boundaries could produce many unfeasible configurations, i.e. fin designs for which performances cannot be calculated by the applications. The flow and structural calculations required a mesh to be adapted in function of the parametric evolution of the fin design and thus, large variations of parameters could produce twisted meshes. The mesh generation had to be as robust as possible. Otherwise, the parameterisation had to progress through rather short variations around the initial values that described the original stabiliser fin.

Our hydrodynamic surface mesh was easily generated, thus the boundaries were large (20% around the initial values). The complexity of fin structure geometry required some careful choices. Fig.6 shows the parameters of fin surface and the parameters of internal structure. Table II describes
parameters and their initial values.

![Fig.6: a) hydrodynamic parameters and b) structural parameters](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOut</td>
<td>Outreach length</td>
<td>2240 mm</td>
</tr>
<tr>
<td>Li</td>
<td>Profile spline length of section i</td>
<td>1950 / 1630 mm</td>
</tr>
<tr>
<td>Thici</td>
<td>Half thickness of section I</td>
<td>0.15 Li / 2</td>
</tr>
<tr>
<td>LThici</td>
<td>Position of max thickness point</td>
<td>0.25 Li</td>
</tr>
<tr>
<td>Nozi</td>
<td>Position of leading edge from pivot axe</td>
<td>0.3 Li</td>
</tr>
<tr>
<td>t1i</td>
<td>Tension of section i spline at the leading edge</td>
<td>0.95</td>
</tr>
<tr>
<td>t2i</td>
<td>Tension of section i spline at the trailing edge</td>
<td>0.55</td>
</tr>
<tr>
<td>t3i</td>
<td>Tension of section i spline at the max thic. Point</td>
<td>1.30</td>
</tr>
<tr>
<td>R1</td>
<td>Root radius</td>
<td>90 mm</td>
</tr>
<tr>
<td>R2</td>
<td>Tip radius</td>
<td>50 mm</td>
</tr>
<tr>
<td>DisH</td>
<td>Distribution of horizontal stiffeners</td>
<td>0.3 Lout</td>
</tr>
<tr>
<td>DisV</td>
<td>Distribution of vertical stiffeners</td>
<td>0.09 Li</td>
</tr>
<tr>
<td>ThicH</td>
<td>Thickness of horizontal stiffeners</td>
<td>10 mm</td>
</tr>
<tr>
<td>ThicV</td>
<td>Thickness of vertical stiffeners</td>
<td>8 mm</td>
</tr>
<tr>
<td>ThicS</td>
<td>Thickness of external surface</td>
<td>2 mm</td>
</tr>
</tbody>
</table>

4. Methods description

We chose as optimisation algorithm the multi-objective genetic algorithm (MOGA) available in modeFrontier, Quagliarella et al. (1997), Poloni et al. (1999), Giassi et al. (2003), user manual at http://www.sirehna.com, http://www;esteo.it. This algorithm allows a multi-objective optimisation where the Pareto frontier is defined in the end. The multidisciplinary nature of our optimisation problem required an effective multi-objective approach.

The next subsections describe the employed methods for the management of interaction between disciplines. Three methods were tested, the first two approaches are usually defined as sequential and single-level (SAND), respectively, Balling and Sobieszczanski-Sobieski (1996), Khatib and Fleming (1998). Finally, we tested our MORDACE method. Comparison between three different methods allowed us to identify real advantages of using our innovative design procedure.

4.1. Sequential Approach

The optimisation was carried out sequentially, Fig.7. First, the hydrodynamic optimisation was performed varying the parameters of the fin external surface. Then, the best found solution was optimised with regards to structural performance values varying the remaining structural parameters.
This optimisation method is the simplest to implement. Besides, the sequential model realistically represents the industrial design process, especially in the ship design domain. However, in a sequential approach, decisions made by the designers at the beginning severely influence and limit the decisions of downstream designers. Possibly, the downstream designers could have difficulty in finding satisfactory or feasible solutions. The optimisation results, that we found using the sequential approach, confirmed the advantages and the disadvantages of this approach.

Hydrodynamic optimisation: The design variables were the 17 parameters defining the external fin surface (Fig.6a). Every set of parameters defined a design point. The genetic algorithm progressed in optimum search, evaluating successive populations of design points. Every new population contained individuals that resulted from the evolution of preceding generations and aimed to improve hydrodynamic performance values. The first population was generated using a random design of experiment (DOE) of 60 individuals. We evaluated 20 generations, i.e. about 1200 individuals. Then, the Pareto frontier was defined, Fig.8. Information about the design space was complete at this step of the optimisation process. The Pareto frontier was made up of 142 designs. The designer had a large number of possible choices.

Structural optimisation: We chose the design ID 1007 from the Pareto designs set. Then, we carried out the structural optimisation varying the remaining parameters and keeping fixed the hydrodynamic parameters and hydrodynamic loads. The parameters were the 7 structural parameters in Fig.6b, the dimension of populations was set to 27 individuals. We evaluated 20 generations, i.e. about 540 individuals. The best found solution was the design ID 226. Table III lists the statistics of both

---

2 The recommended dimension was greater than twice the objective number (3) times the variable number (17).
optimisation processes. Table III shows that the hydrodynamic discipline found designs with very high performance values. On the other hand, structural discipline was strongly penalised. The best found solution was worse than the original design in structural performance. We believe that, if we had chosen another hydrodynamic design, we could have found better solutions. That is exactly the limit of this approach: in choosing ID 1007, we limited the freedom of structural optimisation and we lost information about other hydrodynamic solutions. Besides, the design ID 1007 shape was too streamlined and the structural mesh was not feasible. Thus, we were forced to modify fin shape to create a structural mesh. The sequential approach led to iterative corrections which caused time and performance losses.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>57186</td>
<td>2878</td>
<td>0.433</td>
<td>25872</td>
<td>13.51</td>
<td>808.1</td>
<td>179.6</td>
</tr>
<tr>
<td>Id 1007</td>
<td>62710 (+9.7%)</td>
<td>2873 (-0.2%)</td>
<td>0.444 (+2.5%)</td>
<td>35367 (+36%)</td>
<td>14.68 (+8.6%)</td>
<td>826.2 (+2.2%)</td>
<td>151.5 (-15.6%)</td>
</tr>
<tr>
<td>Id 226</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Time</td>
<td>Evaluation number · CPUs/Evaluation = 1200 · 4.5 = 5400 min = 90h</td>
<td>Evaluation number · CPUs/Evaluation = 540 · 6 = 3240 min = 54h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.2. Single-Level Approach

The single level optimisation method is less simple to implement because of the high level of integration. The system optimiser has to manage directly all analysis applications. Another difficulty of the implementation is in the large problem size as all the variables are combined together. On the other hand, the problem of coupling disciplines is not addressed directly by this approach, Khatib and Fleming (1998). The method implementation used is also called Simultaneous Analysis and Design (SAND), Balling and Sobieszczanski-Sobieski (1996), Khatib and Fleming (1998), Fig.9.

The modeFRONTIER framework easily managed the integration of analysis applications and the large number of parameters (24), objectives (5) constraints (2) and disciplines (2), Fig.10.

Fig.9: Single SAND approach

The population dimension was set to 180 individuals and we evaluated 10 generations, i.e. about 1800 individuals. The population dimension was very large because of large number of parameters and objective functions. Thus, because each calculation was CPU time expensive (it required 10.5 minutes), we were forced to limit the generations number. In order to further reduce the calculation time, we also used response surfaces method (RSM) to extrapolate 40% of the individuals of every new generation. As response surfaces we used the Kringing\(^3\) method and we built them using data of

\(^3\) The Kringing method is best suited to store a large number of designs: 180 designs of the first generation.
the first generation, which was completely calculated. Thus, 1152 designs were evaluated using flow solver and structural analysis, the remaining 648 designs were extrapolated using RSM. The CPU time was reduced by 36% without penalising optimum search. The best found solution was the design ID 1496. Table IV lists the statistics of this optimisation process.

![Fig.10: modeFRONTIER Process Flow](image)

**Table IV: Statistics of the optimisation processes**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>57186</td>
<td>2878</td>
<td>0.433</td>
<td>25872</td>
<td>13.51</td>
<td>808.1</td>
<td>179.6</td>
</tr>
<tr>
<td>Id 1496</td>
<td>57568 (-0.6%)</td>
<td>2779 (-3.4%)</td>
<td>0.429 (-0.1%)</td>
<td>27695 (+7.1%)</td>
<td>12.41 (-8.1%)</td>
<td>544.6 (-32.6%)</td>
<td>117.0 (-34.9%)</td>
</tr>
</tbody>
</table>

We chose ID 1496 because it dominated the original design, i.e. it was better than the original in all objective and constraint functions. This optimisation method allows to take decisions based on a large solution set. The Pareto frontier was made up of 209 designs which optimised the original one. Thus, a designer can choose the best design being aware of design space nature. E.g., ID 1496 gives priority to structural performance values, but, if the designer prefers hydrodynamic gains, there are many designs with high hydrodynamic performance values (e.g. ID 1138 increased lift by 5% improving other objectives more carefully).

The high structural performance values of ID 1496 demonstrate that this approach does not penalise downstream disciplines. Besides, the all-at-once evolution of designs allows designers to immediately detect errors and to avoid iterative corrections. All generated solutions are feasible with regard of two disciplines. However, the approach is very time expensive. In comparison with sequential approach, the single SAND optimisation calculated fewer generations (-50%) and demanded more hours (+40%). However, the sequential approach could demand iterative corrections that are very time expensive because they are not automatic calculations. Besides, the designer may not be able to find feasible or satisfactory solutions. Another disadvantage concerns the industrial implementation of the single SAND approach. This architecture needs a high level of centralisation and integration. Different designers have to concurrently define parameterisations, parameter boundaries and calculation scripts, then a project designer has to link disciplines and to manage whole the optimisation process.
4.3. Multidisciplinary and Robust Design Approaches Applied to Concurrent Engineering (MORDACE) method

This innovative method aims to combine advantages of sequential approach (reduced design time and implementation ease) and single-level method (guarantee of satisfactory and feasible solutions). The MORDACE method is based on a robust design approach and aims to find solutions that are robust with respect to changes in variable values due to discipline interactions, Kalsi et al. (2001). Robust design techniques allow to reduce the effect of uncertainty or variation in design parameters. Variations could be caused by uncontrollable noise factors (robust design Type I) or by deviations in the design variables due to manufacturing tolerance limitations, material variation or evolving design choice (robust design Type II).

The MORDACE method allows to perform the different discipline optimisations concurrently, Fig.11. Every discipline tries to find optimal solution varying design parameters and does not take care of other discipline decisions concerning common variables. In addition to design objectives, disciplines try to minimise the effect of variation in values of common parameters (robust design Type II). The final solution of the optimisation process is a compromise between disciplines on common variables. The variable values have to change in order to reach a compromise and this modification could worsen performance values. Nevertheless, the disciplines propose robust solutions. Thus, negative effects of the modifications are limited.

Fig.11: MORDACE Method

![Diagram of MORDACE Method]

Fig.12: Evaluation of sensitivity

The sensitivity of performance values to variations of common variable values was evaluated using Neural Networks. Fig.12 and Eq.(7) show how we calculated the sensitivity values. For every design solution, we calculated the standard deviation $\sigma_j$ with regard to the common variable $j=1,\ldots, n\text{var}$. Then we summed them. The $\sigma_{\text{Type I}}$ needed $n\text{var}$ times $(2n+1)$ performance evaluations. In order to
reduce CPU time, all these points, including the central point, were extrapolated using Response Surface Method. Thus, none of the applications was called for $\sigma_{\text{TypeII}}$ evaluation. As response surface method, we used multi-layer neural networks with quasi-sigmoid function (QSF), Poloni et al. (1999), Rojas (1996). We built neural networks using initial DOE data, thus the $\sigma_{\text{TypeII}}$ evaluation did not increase the number of application calls.

$$\sigma_{\text{TypeII}}(x_1, x_{1/2}) = \left( \sum_{j=1}^{n_{\text{var}}} \sigma_j(x_1, x_{1/2}) \right) / n_{\text{var}} \quad \text{nvar was the number of common variables} \quad (7)$$

$$\sigma_{\text{TypeII}}(x_1, x_{1/2}) = [\left( \sum_{i=-n}^{n} (f_i(x_1, \ldots, x_j + \Delta i, \ldots, x_{n_{\text{var}}}, x_{1/2} - f_{\text{avg}}(x_1, x_{1/2}))^2 \right) / (2n+1)]^{0.5} \quad (8)$$

If $\sigma_j$ is small, like for design A in Fig.12, the design variable values could be changed without dramatic changes in performance values. On the other hand, design B is a sharp point and changes in variable values strongly deteriorate performance values. Thus, if modifications are necessary due to compromise search, the design B is not the best suited.

Hydrodynamic Optimisation: With regard of robust design type definitions, the roll stabiliser fin was optimised using robust design Type I. As section 3.1. expounded, we minimised $\sigma$ value, i.e. the sensitivity to angle of incidence variations. The MORDACE method is based on robust design Type II approach thus, the fin optimisation with this method contained both robust design types. The hydrodynamic optimisation problem was set as the hydrodynamic sequential optimisation. The only difference was that we defined a new objective function $\sigma_2$ and we transformed $\sigma_1$ into a constraint function. The $\sigma_2$ objective was the sum of Lift and Drag $\sigma_{\text{TypeII}}$ values. Concerning $\sigma_1$ constraint, it had to be less than the original value, i.e. 0.433. We evaluated 20 generations made up of 60 individuals and every design evaluation demanded 4.5 minutes. The Pareto frontier was made up of 281 designs.

Structural Optimisation: The structural optimisation problem was carried out completely differently than the structural sequential optimisation:

1. We defined a new objective function $\sigma_2$ as sum of Mass and Displacement $\sigma_{\text{TypeII}}$ values.
2. The sequential structural optimisation aimed to find the best structural solution for the best hydrodynamic design. The MORDACE method guaranteed disciplines more freedom. The structural optimisation was independent of the hydrodynamic one. Thus, the reference design was the original one and hydrodynamic loads were calculated for every new design. We evaluated them only in the worst condition (i.e. incidence angle $\alpha = 7.5^\circ$) and this single evaluation took only a few seconds. If the flow calculation was more time expensive, we could have charged structure with a generic realistic load. In fact, the two disciplines were independent and the hydrodynamic loads of the compromise solution were anyway different.
3. We added to the 7 specific structural variables $x_2$ (in Fig.6b) 9 variables common to both disciplines. Thus, LOut, L1, L2, t13, t23, LThic1, LThic2, Thic1 and Thic2 were common to hydrodynamic and structural MORDACE optimisations. We imagined that structural designers would have chosen these variables because they defined the main dimensions of stiffeners. We evaluated 20 generations made up of 60 individuals and every design evaluation demanded 7 minutes. The Pareto frontier was made up of 23 designs.

Finding the compromise: The novelty of this method lay in the procedure that we used to find compromise solutions. The data sets of two discipline optimisations were very large. Among available designs, we chose Pareto designs plus all individuals that dominated the original one with regard of two disciplines. Then we evaluated all possible couples made up of hydrodynamic and structural solutions. In every couple there was a difference between the common variable values of two solutions. Thus, we defined this difference as multidimensional distance $d_{\text{IDH}, \text{IDS}}$ between the two solutions $\text{IDH}$ and $\text{IDS}$ (H for hydrodynamic, S for structural):
\[ d_{\text{JDIJS}} = \left( \sum_{j=1}^{n_{\text{var}}} W_j (x_{\text{JDI}} - x_{\text{JDS}})^2 / n_{\text{var}} \right)^{0.5} \]
and
\[ W_j = \left( \sum_{i=1}^{n_{\text{obj}}} |w_{ij}| / n_{\text{obj}} \right) \]

where nobj was the number of objective functions. The weight coefficients \( w_{ij} \) they defined the intensity of relationship between the variable \( j \) and the performance \( i \). High absolute value indicated that the relationship was strong. In this manner, we gave priority to differences between important variables. We evaluated coefficients \( w_{ij} \) with a T-Student analysis.

We chose ten couples with low values of distance and low values of sensitivity, i.e. couples that had small difference between hydrodynamic and structural common variable values and that were robust with regard of change in common variable values. The compromise was found with a weighted mean of common variables:

\[ x_j = \left( W_{ij}/(W_{ij} + W_{kj}) \right) x_{\text{JDI}} + \left( W_{kj}/(W_{ij} + W_{kj}) \right) x_{\text{JDS}} \]
and
\[ W_{kj} = \left( \sum_{i=1}^{n_{\text{obj}}} |w_{ij}| / n_{\text{obj}} \right) \]

nobj was the number of objective functions of discipline \( k \). Thus, between the two common variable sets proposed by the two disciplines, we gave priority to variable values that had more effect on performances.

This compromise method allowed us to define the common variable values of the ten chosen couples, transforming them into ten individuals. Then we evaluated the hydrodynamic and structural performances. All ten individuals were very interesting and improved performances in comparison with the original fin stabiliser fin. The individual coupling ID 185 of hydrodynamic optimisation and ID 364 of structural optimisation was the most interesting, because it dominated the original design. Table V describes the MORDACE optimisation statistics.

<table>
<thead>
<tr>
<th>Results</th>
<th>Lift [N]</th>
<th>Drag [N]</th>
<th>( \sigma_2 )</th>
<th>( \sigma_1 )</th>
<th>Cavitation [Pa]</th>
<th>Max disp [mm]</th>
<th>Mass [Kg]</th>
<th>( \sigma_2 )</th>
<th>Max stress [MPa]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>57186</td>
<td>2878</td>
<td>0.634</td>
<td>0.433</td>
<td>25872</td>
<td>13.51</td>
<td>808.1</td>
<td>0.312</td>
<td>179.6</td>
</tr>
<tr>
<td>Id 185</td>
<td>57566</td>
<td>2776</td>
<td>0.450 (-29%)</td>
<td>0.429</td>
<td>21959</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Id 364</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compromise C10</td>
<td>57460 (+0.4%)</td>
<td>2835 (-1.5%)</td>
<td>0.432 (-0.2%)</td>
<td>27460 (+6.1%)</td>
<td>8.67 (-36%)</td>
<td>615.0 (-24%)</td>
<td>88.07 (-51%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Time</td>
<td>Evaluation number \cdot \text{CPUs/Evaluation} = 1200 \cdot 4.5 = 5400 \text{ min} = 90h</td>
<td>Evaluation number \cdot \text{CPUs/Evaluation} = 1200 \cdot 7 = 8400 \text{ min} = 140h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This approach allows to take decisions based on a large solution set. There were about 60000 possible couples, all potentially improving original fin performances. Nevertheless, performance values of these couples could change after the definition of compromise and it was impossible to verify all couples. Thus, designers had to select and verify a small number of designs. This selection limited the real dimension of possible solutions. However, among the 10 verified designs, we found interesting and varied solutions. The compromise C10 improved all objective functions, but, if the designer preferred hydrodynamic gains, the compromise C3 improved lift value (+7.9\%) improving also other functions but penalising drag (+0.8\%).

In addition, the approach was not time expensive. In comparison with sequential and single-level approach, the MORDACE optimisation calculated 20 generations and demanded fewer hours. In fact, the two disciplines ran independently thus, the total optimisation time was lowest with 140 hours. Another advantage concerns the industrial implementation of MORDACE method. This architecture
does not need centralisation and integration. Different designers can independently define parameterisations, parameter boundaries and calculation scripts, then independently perform optimisations. Finally, a project designer has to verify possible compromises and to choose the more interesting solution.

5. Conclusion

Table VI compares between the three tested methods. The optimisation rate is a mean of performance gains. The negative optimisation rate of the sequential method means that negative structural optimisation results were more important than hydrodynamic gains.

<table>
<thead>
<tr>
<th>Method</th>
<th>Optimisation rate</th>
<th>Generations number</th>
<th>Time [hours]</th>
<th>Possible final choice</th>
<th>Integration level</th>
<th>Risk of corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>-0.7%</td>
<td>20</td>
<td>145</td>
<td>little</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Single Level</td>
<td>9.0%</td>
<td>10</td>
<td>202</td>
<td>large</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>MORDACE</td>
<td>12.4%</td>
<td>20</td>
<td>140</td>
<td>large</td>
<td>low</td>
<td>low</td>
</tr>
</tbody>
</table>

The multidisciplinary optimisation of a roll stabiliser fin with an overview on different methods intrinsically contains elements of interest, especially if we take into account the very positive results that were obtained. The most innovative aspect of this study however, arises from the implementation of the MORDACE method and the comparison with classical methods. In these respects we are very pleased with the experiment, as we were able to identify real advantages of using such a design procedure:

- the required integration level is very low, designers can independently work with their usual application software and specific knowledge.
- the optimisation results are very satisfactory,
- the possibility of final choice is very large,
- the method is not time expensive.

In the future, we must better formalise the compromise choice procedure and verify the method performances with higher number of disciplines.

References


GIASSI, A.; BENNIS, F.; MAISONNEUVE, J.J. (2003), Ship optimisation with distant application tools, Int. CIRP Design Seminar, Grenoble


ROJAS, R. (1996), *Neural Networks*, Springer


Search Safe Ship Trajectories in Collision Situation at Sea by Evolutionary Computation

Roman Smierzchalski, Gdynia Maritime University, Dept. Ship Automation, roms@am.gdynia.pl
Andrzej Lebkowski, Gdansk Univ. of Technology, Control Eng. Dept, andrzejl@ely.pg.gda.pl

Abstract

In a collision situation at sea, the decision support system should help the operator to choose a proper manoeuvre in a given navigational situation, teach him good habits, and enhance his general intuition on how to behave in similar situations in the future. An accepted approach here is the use of a multiple criterion decision problem based system. A new version of the vEP/N++ (Evolutionary Planner /Navigator) algorithm equipped with an on – line mode has been used, as a major component of such a system, for computing the near optimum trajectory of a ship in given environment. Taking into account existing boundaries of the manoeuvring region, navigational obstacles, and moving strange ships, the problem of avoiding collisions at sea was reduced to a dynamic optimisation task with static and dynamic constrains. The introduction of the time parameter, moving constrains representing approaching ships, and the on-line mode are the main distinctions of the newly introduced system. The paper presents selected results in the form of ship trajectories obtained for typical navigation situations.

1. Introduction

When determining a safe trajectory for the own-ship, we look for a trajectory that is a compromise between the cost of necessary deviation from a given route, or from the optimum route leading to a target point, and the safety of passing round all static and dynamic obstacles, here bearing the name of strange ships – targets (own-ship – the ship, for which the trajectory is calculated, strange ship, or target - another ship in the environment which must be passed round at a safe distance). All trajectories, which meet the condition of avoiding the risk of collision to a satisfactory degree make a set of permissible trajectories. Thus the task of determining the safe trajectory is reduced to the choice of an optimum solution, or a subset of optimum solutions from the set of permissible trajectories, taking into account certain cost criteria and the need to satisfy, in any circumstances, the safety conditions. These conditions are, as a rule, defined by the operator, basing on the speed ratio between the ships involved in the passing manoeuvre, as well as the actual visibility, weather conditions, area of navigation, manoeuvrability of the ship, etc. The simplest way of determining the safe trajectory seems to use an additional device - a decision supporting system, which would make an extension of the conventional Automatic Radar Plotting Aids (ARPA) system.

Davis et al. (1982) present a guidance concept for a ship entering the harbour, in which two autonomous systems (VTS and the ship on-board system) were applied for evaluating the trajectory along a given seaway, with simultaneous evaluation of the time correlation of subsequent positions of the ship on the trajectory, taking into account dynamic characteristics of the ship. The concept of Davis et al. (1982) was developed where extended the guidance problem to a set of ships moving along a given voyage route. An autonomous ship guidance system has been presented by Zha et al. (1993). In order to assess the collision situation, and then make a decision and give manoeuvring orders, the authors developed a computer-aided expert system. Certain possibilities of avoiding collisions in the areas of extensive traffic were discussed by Colley et al. (1984) on the basis of statistical analysis. Smierzchalski (1995, 1996a,b) formulated the problem of avoiding collisions as the multi-criterion optimisation task. In the paper, especial attention was paid to the problem of avoiding collisions in the presence of fixed constraints of the navigational area. Furuhashi et al. (1996) presented the attempt to estimate the safe trajectory using genetic algorithms. In the overwhelming majority of the reviewed publications on automatic ship guidance, the navigational process in the areas of intensive traffic was supported using expert systems. On the basis of the evolutionary path guidance concept Michalewicz (1996), Michalewicz and Xiao (1995), Xiao et al.
(1996), Lin et al. (1994), Trojanowski and Michalewicz (1996), a modified version was developed which took into consideration the specifics of the ship guidance in difficult navigational passages by including the motion of other ships, using the evolutionary technique, Smierzchalski (1997a,b,1998,1999a,b,c), Smierzchalski and Michalewicz (1998,2000). Here we present a new version of the vEP/N++ (Evolutionary Planner /Navigator) algorithm, in which especial attention is paid to the trajectory planing in on-line mode in the presence of fixed and moved constraints of the navigational area.

2. Selecting trajectory

One of interesting attempts to solve the task of trajectory planning is an evolutionary programme EP/N (Evolutionary Planner / Navigator) used for determining an optimal path of a moving robot, presented by Michalewicz and Xiao (1995). The solution was searched in the 2D environment using genetic algorithms. Having found the best path, the system controlled the motion of the robot, taking into account new obstacles which appeared on its way. The problem which the authors of the programme EP/N had to solve was similar to that of avoiding collision at sea, as in the both cases a compromising trajectory is to be found.

The compromise has to be reached between the costs of the diversion from an assumed, or the shortest route, and the acceptable safety of passing round static and dynamic constraints. Due to those contrasting criteria the problem can be reduced to a multi-criterion task for optimum trajectory planning. In a sea variant, all trajectories whose vector of echo motion in a relative visualisation obtained from ARPA does not cross the area of danger defined by a safe distance $D_b$ around the actual position of the own ship are permissible trajectories. The optimisation problem consists in selecting the optimum trajectory from the permissible trajectories with respect to an assumed fitness function. After adapting the programme to sea environment, a new programme was developed bearing the name of vEP/N++. The introduced changes include different types of constraints resulting from the new environment in which the ship moves, especially dynamic constraints representing other moving ships. They also include new genetic operators, and time parameter for covering the assumed trajectory. The most recent change made in the programme is the option of on-line trajectory planing.

To obtain full flexibility of a decision support system in situations which may develop in an unforeseeable way, the ship control in a collision situation must have the hierarchical structure. Ship control in a collision situation can be divided into the following stages:
- Introducing input data to procedures calculating the ship trajectory (for vEP/N++ this data is obtained from ARPA),
- Calculating an optimum trajectory for the ship moving in given navigational environment,
- Steering the ship along the calculated trajectory (off-line mode),
- Monitoring parameters of moving obstacles, with possible updating of the introduced input data and
- re-calculation of the optimum trajectory (on-line mode).
- The last, stage, i.e. the on-line control of the ship motion in the hierarchical structure of, ship steering is the main subject discussed in the paper.

3. Environment

Kinetic modelling of the ship motion was used in the programme. The environment in the system EP/N++ consists of static and dynamic constraints. The static constraints represent land, channels, shallows, straits, and/or other navigational constraints like water lines and the areas of traffic separation, for instance. Those areas are modelled by fixed polygons, similar to those used in the process of generating electronic vector maps. Dynamic constraints represent the areas of danger around approaching ships. These areas are called domains and are modelled by hexagons. Their areas depend on the current navigational situation and assumed safety conditions. In modelling the environment the dimension of the own ship is neglected as very small compared to the dimensions of
the domains of strange ships. The dynamics of the ship is taken into consideration in the stage of steering along the already calculated safe trajectory.

All strange ships are attributed moving domains (at course and speed determined by the system ARPA), although only some of them represent real collision threat. It was assumed in the programme that the object representing a real collision threat is the object which has entered the area of observation of the own ship (the distance of 5-8 nautical miles ahead and 2-4 nautical miles astern, depending on the assumed time horizon) an can cross its course at a dangerous distance. The distance of danger is selected by the navigator depending on weather conditions, visibility, sailing area, and own ship’s speed.

In the first version of the programme the own ship was assumed to move at a constant speed V along the entire trajectory, and at the initial time instant \( t_0 \) the motion of strange objects was assumed steady. The present version makes it possible to introduce changes to the parameters of motion of the dynamic constraints. It also includes the mutation operator for the own ship speed. The speed parameters of the dynamic objects, as determined by the system ARPA, include: bearing, distance, course and speed. In the evolutionary algorithm each trajectory is attributed an instantaneous dynamic area connected with each individual moving strange object. The instantaneous position of the dynamic area with respect to the passing trajectory is evaluated by time \( t \), determined from the crossing point \( (x_{int,y_{int}}) \) of the trajectory of the own ship \( S \) with the trajectory of the strange object. This crossing point \( (x_{int,y_{int}}) \) is considered the point of collision threat. The time needed for covering the distance from the starting point \( (x_0,y_0) \) to the crossing point \( (x_{int,y_{int}}) \) by the own ship moving at speed \( V \) along trajectory \( S \) is \( t \). For the same time \( t \) the position of the domain of the strange object is calculated in a similar way and taken into account via its trajectory. The static constraints are seen in the same way from all trajectories and do not have to be re-calculated. In the programme vEP/N++ a new position of the domain is only calculated for the best individual in each population. The level of collision danger is determined using the depth of penetration of the constraint by the trajectory.

4. Simulation studies

In order to test operational correctness of the on-line version of programme vEP/N++, certain trajectories were calculated using the both versions and then compared. The test were divided into three phases. In the first phase, passing paths obtained in the off-line mode were tested. Then the real motion was studied of the own ship travelling along the assumed trajectory and passing other objects.
The level of identity was assessed with which the positions of passed ships – domains were calculated with respect to the own ship. For the on-line version, a quality assessment was made for the calculated trajectory on the basis of on-line calculations performed after changing parameters of motion of one or more dynamic constraints. The comparison was made for two sample environments characterised by a relatively high level of complexity.

The first case, Fig.1(a), presents the situation in which the own ship passes round four islands and four objects moving from different directions and at different speeds. Input parameters for the simulation, in which the population number was equal to 40, are shown in Fig.1(a).

Ship trajectory adaptations, made in the off-line mode after the own ship has met four moving objects in a collision situation, are shown in Fig.1(b),(c), after 500, 1000, 1500, and 2000 generations, respectively. The solution obtained after 2000 generations made it possible to pass round the islands, at the same time leaving aside the moving objects. Current speeds of the own ship were individually calculated for each trajectory segment which allowed the ship to reach the target point in a safe way and the shortest possible time. The realisation of the proposed trajectory optimised the ship passage cost with respect both to safety and economic conditions, eliminating the risk of excessive approach to the passed objects. The results are presented in a relative form, in which the domains of the dynamic constraints are drawn up in positions which they will reach when the own ship is in front of their bows or behind their sterns.
The next phase studies the real motion of the own ship travelling along the assumed trajectory and passing strange objects. This phase makes it possible to assess more accurately the correctness of positions of the passed ships – domains with respect to the own ship. Fig. 2 presents the navigational situation in the real motion after $T_i = 10, 20, 30$ and $40$ min ($i = 1,...,4$). In contrast to the relative presentation (Fig.1(b), (c)), here the positions of the domains representing dynamic constraints and the position of the own ship on the trajectory are determined after the time which elapsed after the simulation has started (Option „Run” pressed down) to the time instant when the option „real time situation” was started. The option „real time situation” was implemented in order to stop the motion of all objects for a given time instant and to simulate possible changes in parameters of motion of the domains representing passed objects interpreted as dynamic constraints. Changing parameters of passed objects creates a new navigational situation – a new environment which provokes the adaptation of the ship trajectory calculated in the off-line mode. Switching to the on-line mode, the system vEP/N++ adapts the off-line trajectory version to the new environment, treating the time when the objects were stopped as the starting point of the modified trajectory.

For the test environment No. 1, the algorithm switched to the on-line mode at $T = 20$ min. Then selected parameters were changed, namely: the course of the object seen in the right part of the screen, which was changed from 45 degrees to 210 degrees, and the speed of motion of the object seen in the upper left part of the screen, changed from 5.2 to 17.6 knots. These changes were dictated by the requirement to observe the operation of the algorithm in extremely complex environment.
Fig. 3 (b) shows a newly calculated safe trajectory (generation number = 2000) which has taken into account the above changes. After comparing Fig.1 (b) (c) and Fig. 3 (b) one can see that the path calculated in the on-line mode is similar to the trajectory obtained in the off-line mode, with certain differences resulting from trajectory corrections made due to environment changes introduced. This proves the operational correctness of the on-line version. Fig. 4 shows the navigational situation in the on-line mode for $T_5 = 50$ min after trajectory correction.

![Fig. 4: Test environment No. 1, $T_5 = 50$ min. Option “real time situation”](image)

The second case (No. 2) (Fig. 5 (a)) presents the situation when the own ship leaves the channel and meets three objects moving from different directions and at different speeds. Input parameters for the simulation made for this environment are shown in Fig. 5 (a). The population number is equal to 40. The speed of the own ship is equal to $V = 18.7$ kn. The adaptation of the own ship trajectory calculated in the off-line mode for collision meeting of three moving objects is shown in Fig. 5 (b), after 1000 generations.

![Fig. 5: Test environment No. 2 (a). Generation=1000 (b). Mode: off-line](image)

Fig.6 refers to the second phase of the algorithm operation. It presents the navigational situation for the tested environment for $T_1 = 10, 20$ min ($i=1,2$) after start. Positions of the domains representing dynamic constraints and the position of the own ship on the trajectory are determined after the time which elapsed from the beginning of the simulation (Option “Run” pressed) to the time instant when the option “real time situation” was started.

![Fig. 6: Test environment No. 2, $T = 20$ min (a), 30 min (b). Option “real time situation”](image)

In order to examine the operation of the on-line mode for test environment No. 2, at $T_3 = 30$ min selected parameters were changed in the motion of the object seen in the lower left part of the screen.
The speed was changed from 12.7 knots to 25 knots, and its course from 45 degrees to 100 degrees. The time of switching the system to the on-line mode is shown in Fig. 7 (a).

![Fig.7: Test environment No. 2, T3 = 30 min. Changing parameters of motion in domain (a). New optimum trajectory (b). Mode: on-line](image)

The newly calculated safe trajectory taking into account changes introduced to the motion of the strange object is shown in Fig.7 (b).

For the test environment No. 2, the comparison of the solutions shown in Figs.5 and 7 leads to the conclusion that the path calculated in the on-line mode is safe and close to optimum. This proves the correctness of operation of the new programme version. It should be stressed here that two most complex environments were selected for the present tests, out of all environments earlier studied by the author, see Colley et al. (1984).

![Fig.8: Test environment No. 2, T = 50 min. Option “real time situation”](image)

Fig.8 presents the navigational situation for T3 = 50 min after trajectory correction was made in the on-line mode.

5. Conclusions

The evolutionary method of evaluating the safe and optimum trajectory of the own ship in the environment with static and dynamic constraints is a new approach to the problem of avoiding collisions at sea. The performed tests lead to the following conclusions:

- evolutionary algorithms can be used for solving problems of avoiding collisions at sea, with the environment modelled in the form of polygons representing navigational constraints and strange object moving at changing speed and course,

- the evolutionary task of evaluating the ship trajectory in a collision situation is substituted by an adaptive search for a set of safe trajectories in the permissible space X and choosing from them the best possible trajectory with respect to the adaptation function.

The evolutionary algorithm operating in nearly real time makes it possible to solve the problem in the widest possible range, where the optimum trajectory is calculated taking into account changes in course and speed of the own ship along particular trajectory segments, and strategy changes of the passed objects, both for the situation of meeting objects on open water regions and restricted offshore regions. The present version of the evolutionary algorithm based system still reveals some weak points, the most significant of which, perhaps, is the inability to recognise the scale of the passed objects. As a result, the system attributes the same level of danger, for instance, to a big tanker and a
small yacht. Besides, the system does not identify water marks, such as buoys and floating beacons, which may lead to a situation in which the calculated trajectory goes beyond the water region.

It is interesting to compare various algorithms for planning safe trajectories in collision situations. In the following study for each method we considered (1) required computational effort (2) the scope of the problem being solved, (3) feasibility of using the algorithm in real navigational situations, (4) type of allowed manoeuvres (e.g., course or/and speed), (5) assumed safe distance between ships, (6) number of targets, and (7) type of constraints.

The following algorithms were included in the study:
- indispensable manoeuvre, the algorithm returns a single change of the course and/or the speed of the own-ship for avoiding collision in a multi-target encounter,
- utilisation of potential collision threat area, due to simple graphic of the potential collision threat area (PCTA), it is possible to display it on a monitor of the anti-collision system in a real time,
- indispensable trajectory, the algorithm returns a safe trajectory as a series of manoeuvres (changes of the course and/or the speed of the own-ship),
- optimal trajectory, the algorithm returns a safe trajectory as a series of manoeuvres (changes only of the course of the own-ship). As opposed to the previous three algorithms, it can be applied not only to "open sea" scenarios, but can include static navigational constraints (e.g., shore line),
- evolutionary trajectory, the algorithm returns a safe trajectory as a series of manoeuvres and it can be applied to scenarios with static and dynamic constraints.

The evolutionary method is a clear winner because of its generality and a reasonable computational time (average CPU time = 35s). Recently this evolutionary method was applied in real scenarios. An interesting comparison was made between the recommendations of evolutionary algorithm and actions of an experienced captain. This comparison test indicated that the solutions were almost identical. However, it appeared also that evolutionary system was more flexible: it controlled the movement parameters of targets.

References


SMIERZCHALSKI, R (1996a), The Decision Support System to Design the Safe Maneuver Avoiding Collision at Sea. ISAS’96, Orlando, pp. 95-103

SMIERZCHALSKI, R. (1997), *Trajectory planning for ship in collision situations at sea by evolutionary computation*. 4th IFAC Conf. on Manoeuvring and Control of Marine, pp.693-698, Brijuni


SMIERZCHALSKI, R. (1999a), *Static and Dynamic Constrains in Evolutionary Problem of Ship’s Trajectory Planning at Sea*, 3rd World Multiconference on Systemics, Cybernetics and Informatics (SCF99), Orlando, pp. 1124-1130

SMIERZCHALSKI, R. (1999b); *Evolutionary trajectory planning of ship in navigation traffic areas*. J. Marine Sciences and Technology, Vol. 4, No 1, Springer, pp. 1-6

SMIERZCHALSKI, R. (1999c), *An Evolutionary Method of Ship’s Trajectory Planning at Sea*, Int. Conf. IEEE Intelligent Transportation System, Tokyo, pp. 907-912


Training Recurrent Neural Networks with Noisy Data for Manoeuvring Simulation

Lúcia Moreira, Carlos Guedes Soares, Instituto Superior Técnico, Lisbon/Portugal
lucia@mar.ist.utl.pt; guedess@mar.ist.utl.pt

Abstract

A Recursive Neural Network (RNN) manoeuvring simulation model for surface ships is presented. Inputs to the simulation are the orders of rudder angle and ship’s speed and also the recursive outputs velocities of sway and yaw. The model is used to test the capabilities of artificial neural networks (ANNs) in manoeuvring simulation of ships when the data used for training is noisy. The simulations will be performed using as basis the ship Mariner. The data generated to train the network is obtained through a manoeuvrability model implemented in a block diagram, performing the simulation of different manoeuvring tests. The noisy data is generated introducing white noise into two inputs of the system: the rudder angle and advance speed of the ship. The results achieved using RNNs and the conventional mathematical model are compared in order to analyse the accuracy of the RNN and its generalization ability.

1. Introduction

Full-scale measurements are one of the means for us to characterize and understand the ship as a system and to predict its future behaviour. If there are underlying deterministic equations, the ship’s behaviour can be determined by solving the equations with the given initial condition. The manoeuvring simulation of ships requires that the hydrodynamic coefficients of the equations are known. If they are unknown, so one needs to find out the rules that govern the system evolution. The governing rules may be inferred from the regularities in past time histories measurements.

Weigend and Gershenfeld (1993) defined three goals for time series analysis: forecasting, modelling, and characterization. Forecasting is predicting the short-term evolution of the system. Modelling involves finding a description that accurately captures the features of the long-term behaviour. The goal of characterization is to determine the fundamental properties of the system, such as the degrees of freedom or the amount of randomness. The three goals are related but may not be identical. Models that produce good short-term forecasts may perform poorly on long-term behaviour and vice versa. The complexity of model useful for short-term prediction may not be related to the actual complexity of the system, because it may contain the model of short-term noise. Modelling and characterization are the goals of system identification in control and signal processing.

The traditional methods for time series prediction use the moving average (MA), autoregressive (AR), or the combination of the two, the ARMA model. These models are linear and are inadequate for even very simple non-linear systems. Neural network approaches have produced very good short-term predictions. Conventional methods for non-linear system identification have limited success and neural network approaches are a very good promise in this field and can be seen in detail in Bose and Liang (1996).

After obtaining some results and comparisons between the data simulated using a RNN and a mathematical model described in Moreira and Guedes Soares (2002a), the study of how the neural network behaves when trained with noisy data seems necessary to estimate how accurate the prediction can be. This is a very important feature in the prediction of trajectories using real data obtained from full-scale trials in the training of the network.
2. Manoeuvrability Model Simulation

A mathematical model described in Moreira and Guedes Soares (2002a), was slightly simplified and implemented using the software MatLab and its toolbox Simulink. This software was chosen due to its interface capability with the user and due to the easy visualisation and comprehension of the system. The Simulink has several algorithms to solve differential equations, and for this particular case was chosen the Runge-Kutta method. The complete block diagram of the model is illustrated in Figure 1.

![Complete blocks diagram of the manoeuvrability model](image)

Although this model is a variation of a more complex model that allows obtaining the ship’s trajectory with good accuracy explained in Sutulo et al. (2002), this has the advantage to allow a rapid visualisation of the manoeuvres because it allows a reasonable interface with the user.

The noisy data was generated introducing white noise into the inputs of the system, i.e., in the rudder angle and advance speed of the ship. For this purpose a band-limited white noise block that generates normally distributed random numbers was used. This block produces the output at a specific sample rate, which is related to the correlation time of the noise. Theoretically, continuous white noise has a correlation time of 0, a flat power spectral density (PSD), and a covariance of infinity. In practice, physical systems are never disturbed by white noise, although white noise is a useful theoretical approximation when the noise disturbance has a correlation time that is very small relative to the bandwidth of the system. To produce the correct intensity of this noise, the covariance of the noise is scaled to reflect the implicit conversion from a continuous PSD to a discrete noise covariance. The appropriate scale factor is 1/τc, where τc is the correlation time of the noise. This scaling ensures that the response of a continuous system to the approximate white noise has the same covariance as the system would have if true white noise has been used. Because of this scaling, the covariance of the signal from the block output is the noise power divided by τc. The noise power parameter is actually the height of the PSD of the white noise. The correlation time used for this noise generation was 1 for both rudder angle and speed and the noise power was 40 for the rudder angle (σθ = 6.3°) and 2 for the speed (σv = 1.4’ = 0.7 m/s).

Two cases will be considered separately in order to test the capabilities of the ANNs in manoeuvring simulation of ships when the data used for training is noisy: firstly, noisy rudder angle input and no noise in the velocity; secondly, no noise in the rudder angle and noisy velocity input. These cases will be compared separately and the effect of the noise will be analysed in each input and how it affects the behaviour of the whole system.
3. Structure of the Simulation Model

The learning problem through neural networks described here will consist of simulating the velocities of sway, \( v(t) \), and yaw, \( r(t) \), assuming that input parameters are the rudder angle, \( \delta(t) \), and the ship’s speed, \( V(t) \). The neural network inputs will be the rudder angle, \( \delta(t) \), and the ship’s speed, \( V(t) \) and also the velocities of sway, \( v(t-1) \), and yaw, \( r(t-1) \). All of these data will be obtained from manoeuvrability simulations with the manoeuvrability model implemented in a block diagram in Simulink. The network outputs will be the velocities of sway, \( v(t) \), and yaw, \( r(t) \).

The kinematical equations used for the trajectories calculation are

\[
\begin{align*}
\dot{\xi} &= V \cos \psi - v \sin \psi \\
\dot{\eta} &= V \sin \psi + v \cos \psi \\
\dot{\varphi} &= r
\end{align*}
\]

The RNN will be trained to imitate the parameters of sway and yaw resulting from the simulations for different speeds and rudder angles requested. The Figure 2 illustrates the representation used in this version of the RNN simulator and shows the type of typical representation of many systems of artificial neural networks. Each node (circle) in the network diagram corresponds to the output of a unit and the lines that input the node from left are its inputs.

As can be seen, there are 10 units that receive inputs directly from the achieved data. These units are called “hidden” units because their output is valid just inside the network and are not valid as part of the global network output. Each one of these 10 hidden units computes a single real output based on a weighted combination of their inputs. These outputs of the hidden units are then used as inputs of a second layer of two output units. Each output corresponds to either a velocity of sway or yaw in the instant \( t \) and these outputs will again input the network (cyclic) as being the inputs sway and yaw in the instant \( t-1 \).

![Fig.2: RNN to simulate ship’s manoeuvrability.](image)

This network structure is typical of many artificial neural networks. Here the individual units are interconnected in layers. In general, ANNs can have with many other types of structures – acyclics or cyclics, directs or indirects. In this paper will be used the approximation with ANNs more common and practical, which is based in the Backpropagation algorithm.
4. Validation of the Learning through ANNs to the Case Study

ANN training methods are adequate even for those problems where the training data corresponds to noisy data, as can be the case of the data achieved on board in full-scale manoeuvring trials. One of the potential capabilities of the model described in this paper will be the simulation of the motions using training data obtained from full-scale tests. Work in this field has been developed and an improved RNN manoeuvring simulation tool for surface ships, trained and validated with data acquired from two ships operating in the open ocean, is described in Hess and Faller (2000). In Moreira and Guedes Soares (2002b) the same methodology used before in Moreira and Guedes Soares (2002a) is applied to analyse full-scale manoeuvring data from patrol vessels and validate the RNN model with real data. The description of the testes performed with the Portuguese Navy ships are described in Guedes Soares et al. (2002). RNNs are trained with the full-scale data and afterwards model simulations are compared with the full-scale results.

The Backpropagation algorithm is the most widely used learning technique for ANNs. It can be used in problems with the following characteristics:

- **The training examples can contain errors.** The learning methods through ANNs are quite robust to noise in the training data.

- **Long training timings are acceptable.** Typically the network training algorithms require long training timings. The training timing can vary between some seconds and many hours, depending of factors such as the number of weights in the network, the number of considered training examples and the assumed values for the learning algorithm parameters.

- **A quickly evaluation of the target (desired) function learned can be required.** Although the training timings of the ANN can be long, the evaluation of the learned network, in such a way to apply it to a subsequent example, is typically very fast.

- **The ability of human understands the learned target function is not important.** The learned weights through neural networks are usually hard to interpret by human.

5. Network Topology

Data for training, cross-validation and test the neural networks was acquired from simulations performed with the manoeuvrability model implemented in a block diagram in Simulink. Because the manoeuvring simulations exhibit similar turning characteristics for both right and left turns, the simulation performed were just for positive rudder angles. The architecture of the neural network is illustrated schematically in Figure 3.

![Recursive neural network](image-url)  
**Fig.3:** Recursive neural network.
The network consists of three layers: an input layer, one hidden layer and an output layer. Within each layer are nodes, which contain a non-linear transfer function that operates on the inputs to the node and produces a smoothly varying output. The binary sigmoid function was used for this work; for an input \( x \) it produces the output \( y \), which varies from 0 to 1 and is defined by

\[
y(x) = \frac{1}{1 + e^{-x}}
\]

Note that the nodes in the input layer simply serve as a means to couple the inputs to the network; no computations are performed within these nodes. The nodes in each layer are fully connected to those in the next layer by weighted links. As data travels along a link to a node in the next layer it is multiplied by the weight associated with that link. The weighted data on all links terminating at a given node is then summed and forms the input to the transfer function within that node. The output of the transfer function then travels along multiple links to all the nodes in the next layer, and so on. So, as shown in Figure 3, an input vector at a given time step travels from left to right through the network where it is operated on many times before it finally produces an output vector on the output side of the network.

Not shown in Figure 3 is the fact that most nodes have a bias; this is implemented in the form of an extra weighted link to the node. The input to the bias link is the constant 1, which is multiplied by the weight associated with the link and then summed along with the other inputs to the node. An RNN has feedback; the output vector is used as additional inputs to the network at the next time step. For the first time step, when no outputs are available, these inputs are filled with initial conditions. The network described here has 4 inputs. The hidden layer contains 10 nodes and each of these nodes uses a bias. The output layer consists of 2 nodes, and also uses bias units. The network contains 16 computational nodes and a total of 72 weights and biases: 50 weights (4 inputs \( x \) 10 + 10 bias weights) related with the input data plus 22 (10 x 2 outputs + 2 bias weights) related with the output. The input vector consists of the rudder angle and advance speed of the ship, and the network then predicts at each time step the sway velocity component \( v \) and the yaw velocity component \( r \). These velocity predictions are then used to compute at each time step the heading angle and the trajectory components. Recursed outputs from the prior step are used as two additional contributions to the input vector.

6. Network Training

The collection of input and corresponding target output vectors comprise a training set, and these data are required to prepare the network for further use. Data files containing time histories of tactical circles manoeuvres formed the training sets. After the neural network has been successfully trained using cross-validation, the weights that provide the minimum error to the network are saved. Thus, the network may be presented with an input vector similar to the input vectors in the training set (i.e., drawn from the same parameter space), and it will then produce a predicted output vector. This ability to generalise, i.e., to produce reasonable outputs for inputs not encountered in training is what allows neural networks to be used as simulation tools. To test the ability of the network to generalise, a separated subset of test data files must be used. These test data files then demonstrate the predictive capabilities of the network.

A neural network was trained in this manner to predict tactical circles. In each case about 70% of the data files comprised the training set with 30% set aside as cross-validation files. The networks were trained for 65500 iterations (epochs). In each iteration the time series are presented for all inputs and outputs for all files in the training set. During this training process, training is paused every 10 iterations, and the network is tested for its ability to generalise. To carry this out, all of the files in the training set are combined with the cross-validation files and the entire set is presented to the network. After training has concluded, one examines the error measures as a function of the number of
iterations at which training should have ceased and where minimum absolute errors and maximums in the measures occur. Summarising, a neural network was trained to predict tactical circles using the procedure described in this section. The results of these simulations are detailed further ahead.

The non-linear activation function used in this case is the sigmoid function that has saturation values of (0,1). Presenting a data set whose values are not bounded by the saturation range will force the neurone to its saturation point and it will no longer respond to changes in input. In this case study was chosen to normalise the data between 0.2 and 0.8.

7. Case Study: Analysis of the RNN Performance using Noisy Data for Training in the Simulation of the Manoeuvrability Characteristics of Ships

In the following case study, the results achieved using recursive neural networks and the conventional mathematical model built in a form of block diagram are compared. The learning objective in this case evolves the classification of the sway and yaw velocities of the ship to several rudder angles and advance speed using noisy data for training. Sampling period used was 1 second and 16 runs of simulated data are available, i.e., 16 tactical circles. One target function will be learned through the data obtained in the manoeuvrability simulations. Given as input the rudder angle, the advance speed of the ship, sway and yaw at the instant \( t-1 \), the RNN can be trained to produce as outputs the sway and yaw velocities at the instant \( t \).

7.1. Modelling Options

**Input Encoding.** Given that the input of the RNN will be a representation of the order of manouevring of the ship, a modelling key is how to encode this order. One option could be just to use the rudder angle and the advance speed of the ship as inputs. One difficulty that could happen with this option would be that this leads to a higher variable number of manouevring characteristics (velocities of sway and yaw) for each instant of manouevre. Taking as inputs the rudder angle, the advance speed of the ship and also the velocities of sway and yaw at the instant \( t-1 \) the possible number of variables will be decreased in the learning of the velocities of sway and yaw at the instant \( t \).

The values were normalised between 0.2 and 0.8 in order that the inputs of the network have values in the same range that the activation of the hidden unit and output unit.

**Output Encoding.** The RNN must provide as output values the sway and yaw velocities for each instant \( t \). The output values were also normalised between 0.2 and 0.8. If one tries to train the network to tune the desired values exactly equal to 0 and 1, the gradient descent will force the weights to grow without limit. On another hand, the values 0.2 and 0.8 are obtained using a sigmoid unit with finite weights.

**Network Structure.** For this work a standard structure of a recursive neural network, using two layers of sigmoid units (one hidden layer and one output layer), was selected. Using 16 manoeuvres, the training time was approximately 2 hours and 20 minutes using a Pentium IV (1.7 GHz) for the tactical circles network.

**Other Parameters of the Learning Algorithm.** On this learning experiences the learning rate \( \eta \) was settled equal to 0.1 and the momentum \( \alpha \) was chosen equal to 0.7. The weights of all network units were randomly initialised. 65500 iterations were used because in the software used for training was not possible to establish a stopping criterion. The data available was separated in two different groups: one set for training and another set for validation. After 10 iterations the network performance was evaluated through the validation set. The final network selected was that with better accuracy through the validation set. The final accuracy obtained was measured through a separated test set different from the training and cross-validation sets.
7.2. Results of the RNN Performance using Noisy Data for Training

Beginning with the RNN to simulate the tactical circles, the network was trained using 11 tactical circles with five set aside for cross-validation. A set of 8 tactical circles was also used for test. All these manoeuvres are described in Table 1. Figures 4 up to 7 depict the time histories for sway and yaw and the circle trajectories obtained through the RNN simulation superimposed upon the time histories and the circle trajectories obtained through the simulation with the model using noisy input for the rudder angle and no noisy input for the velocity. These inputs are the time histories for the rudder deflection angle and for the advance speed of the ship generated with noise. In each case the only information provided to the trained network were the time histories for the rudder deflection angle and for the advance speed of the ship and the initial conditions of the vehicle. The training runs that are shown are comprised by three of the 11 manoeuvres used for training, three of the 5 validation runs and 8 separated circles used for test. The three training runs that are shown represent a mixture of three different rudder angles and two different approach speeds. The validation manoeuvres contain two different rudder angles and three different approach speeds. The test manoeuvres comprise a mixture of 6 different rudder angles with 7 different approach speeds. Solid lines represent the simulation using the RNN and the dashed lines are used for the previous predictions. In all the cases the circles are simulated with as input for the rudder angle a step function at 20 seconds.

<table>
<thead>
<tr>
<th>#</th>
<th>Test</th>
<th>Approach Speed (knots)</th>
<th>Rudder Angle (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Circle</td>
<td>Max</td>
<td>Max.</td>
</tr>
<tr>
<td>2</td>
<td>Circle</td>
<td>75% Max</td>
<td>Max.</td>
</tr>
<tr>
<td>3</td>
<td>Circle</td>
<td>50% Max</td>
<td>Max.</td>
</tr>
<tr>
<td>4</td>
<td>Circle</td>
<td>25% Max</td>
<td>Max.</td>
</tr>
<tr>
<td>5</td>
<td>Circle</td>
<td>Max</td>
<td>75% Max.</td>
</tr>
<tr>
<td>6</td>
<td>Circle</td>
<td>75% Max</td>
<td>75% Max.</td>
</tr>
<tr>
<td>7</td>
<td>Circle</td>
<td>50% Max</td>
<td>75% Max.</td>
</tr>
<tr>
<td>8</td>
<td>Circle</td>
<td>25% Max</td>
<td>75% Max.</td>
</tr>
<tr>
<td>9</td>
<td>Circle</td>
<td>Max</td>
<td>50% Max.</td>
</tr>
<tr>
<td>10</td>
<td>Circle</td>
<td>75% Max</td>
<td>50% Max.</td>
</tr>
<tr>
<td>11</td>
<td>Circle</td>
<td>50% Max</td>
<td>50% Max.</td>
</tr>
<tr>
<td>12</td>
<td>Circle (*)</td>
<td>25% Max</td>
<td>50% Max.</td>
</tr>
<tr>
<td>13</td>
<td>Circle (*)</td>
<td>Max</td>
<td>25% Max.</td>
</tr>
<tr>
<td>14</td>
<td>Circle (*)</td>
<td>75% Max</td>
<td>25% Max.</td>
</tr>
<tr>
<td>15</td>
<td>Circle (*)</td>
<td>50% Max</td>
<td>25% Max.</td>
</tr>
<tr>
<td>16</td>
<td>Circle (*)</td>
<td>25% Max</td>
<td>25% Max.</td>
</tr>
<tr>
<td>17</td>
<td>Circle (**)</td>
<td>30% Max</td>
<td>50% Max.</td>
</tr>
<tr>
<td>18</td>
<td>Circle (**)</td>
<td>Max</td>
<td>35% Max.</td>
</tr>
<tr>
<td>19</td>
<td>Circle (**)</td>
<td>95% Max</td>
<td>30% Max.</td>
</tr>
<tr>
<td>20</td>
<td>Circle (**)</td>
<td>40% Max</td>
<td>25% Max.</td>
</tr>
<tr>
<td>21</td>
<td>Circle (**)</td>
<td>25% Max.</td>
<td>55% Max.</td>
</tr>
<tr>
<td>22</td>
<td>Circle (**)</td>
<td>Max.</td>
<td>30% Max.</td>
</tr>
<tr>
<td>23</td>
<td>Circle (**)</td>
<td>90% Max.</td>
<td>25% Max.</td>
</tr>
<tr>
<td>24</td>
<td>Circle (**)</td>
<td>45% Max.</td>
<td>20% Max.</td>
</tr>
</tbody>
</table>

(*) circles used for cross-validation
(**) circles used for test

As basis to compare the convergence of the RNN the results obtained with the neural network model for the tests presented in Table 1 but using data without noise are presented before proceeding with
the simulations with noise. Averaged errors of the tactical circles trajectories have been tallied in Table 2 for two variables: $x$ and $y$ for the case in that the network was trained with filtered data and the simulations are done with no noise in the inputs. The first number in each cell is an error averaged over all 24 manoeuvres, whereas the second number is the error averaged over the 8 test circles only.

Table 2: Tactical circles error measures averaged over all manoeuvres / averaged over test runs only (network trained without noise and no noise in the simulation inputs)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Absolute Error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>131 / 72 m</td>
<td>10.8 / 3.5</td>
</tr>
<tr>
<td>$y$</td>
<td>130 / 71 m</td>
<td>10.8 / 3.5</td>
</tr>
</tbody>
</table>

The following simulations in Figures 4 up to 7 depict some of the time histories for sway and yaw and the circle trajectories obtained through the RNN simulation superimposed upon the time histories and the circle trajectories obtained through the simulation with the model using noisy input for the rudder angle and no noisy input for the velocity. The aim of these simulations is to compare the convergence of the network increasing the noise in each input.

Fig.4: Test #6 - Time histories for sway and yaw 75% Max. Rudder Angle; 75% Max. Speed Rudder Angle Input with noise

Fig.5: Test #6 - Ship's Trajectory 75% Max. Rudder Angle; 75% Max. Speed Rudder Angle Input with noise

Fig.6: Test #17 - Time histories for sway and yaw 50% Max. Rudder Angle; 30% Max. Speed Rudder Angle Input with noise

Fig.7: Test #17 - Ship’s Trajectory 50% Max. Rudder Angle; 30% Max. Speed Rudder Angle Input with noise
The predictions for the training circles trained with noise are quite reasonable. It is visible that the trained network has learned the noisy behaviour for the velocities of sway and yaw. This is evident by the good performance of the network on this type of manoeuvres.

To quantify the convergence of the RNN, averaged errors of the tactical circles trajectories have been tallied in Table 3 for two variables: $x$ and $y$. As in Table 2, the first number in each cell is an error averaged over all 24 manoeuvres, whereas the second number is the error averaged over the 8 test circles only.

Table 3: Tactical circles error measures averaged over all manoeuvres / averaged over test runs only (network trained with noisy data and noisy rudder angle simulation input)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Absolute Error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>224 / 214 m</td>
<td>13.6 / 10.7</td>
</tr>
<tr>
<td>$y$</td>
<td>201 / 179 m</td>
<td>12.7 / 9.3</td>
</tr>
</tbody>
</table>

Now the same tests plotted before in Figures 4 up to 7 are shown in Figures 8 up to 11 but using filtered inputs (without noise).
In the attempt to simulate the tests using a network trained with noisy data but with filtered inputs we can verify that the network lost accuracy. The network still performs well in some cases as seen in Figures 8 up to 11. To quantify the convergence of the RNN, averaged errors of the tactical circles trajectories have been tallied in Table 4 for two variables: $x$ and $y$, and are given an error averaged over all the tactical circle manoeuvres.

Table 4: Average of tactical circles error measures
(network trained with noisy data and no noise simulation inputs)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Absolute Error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>328 m</td>
<td>19.4</td>
</tr>
<tr>
<td>$y$</td>
<td>298 m</td>
<td>18.1</td>
</tr>
</tbody>
</table>

To make some estimates of precision error manoeuvres with the same rudder deflection and approach speeds were compared. For the tactical circles using as inputs noisy data the turning diameter differed by 8.4 - 887 m or 0.4 - 46%. For the tactical circles using as inputs filtered data (without noise) the turning diameter differed by 9 - 931 m or 1 - 47%.

Figures 12 up to 15 depict the time histories for sway and yaw and the circle trajectories obtained through the RNN simulation superimposed upon the time histories and the circle trajectories obtained through the simulation with the model using noisy input for the velocity and no noisy input for the rudder angle.

**Fig.12:** Test #3- Time histories for sway and yaw  
Max. Rudder Angle; 50% Max. Speed  
Velocity Input with noise

**Fig.13:** Test #3 - Ship’s Trajectory  
Max. Rudder Angle; 50% Max. Speed  
Velocity Input with noise

**Fig.14:** Test #9- Time histories for sway and yaw  
50% Max. Rudder Angle; Max. Speed  
Velocity Input with noise

**Fig.15:** Test #9 - Ship’s Trajectory  
50% Max. Rudder Angle; Max. Speed  
Velocity Input with noise
The predictions for the training circles trained are slightly worse than for the case studied before, showing that the noise in the velocity input affects more the behaviour of the neural network model than the noise in the rudder angle.

To quantify the convergence of the RNN, averaged errors of the tactical circles trajectories have been tallied in Table 5 for two variables: \( x \) and \( y \). Again, the first number in each cell is an error averaged over all 24 manoeuvres, whereas the second number is the error averaged over the 8 test circles only.

Table 5: Tactical circles error measures averaged over all manoeuvres / averaged over test runs only (network trained with noisy data and noisy ship’s speed simulation input)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Absolute Error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>287 / 251 m</td>
<td>21.5 / 15.1</td>
</tr>
<tr>
<td>( y )</td>
<td>286 / 253 m</td>
<td>21.1 / 15.1</td>
</tr>
</tbody>
</table>

Now the same tests plotted before in Figures 12 up to 15 are shown in Figures 16 up to 19 but using filtered inputs (without noise).
In the attempt to simulate the tests using a network trained with noisy data but with filtered inputs we can verify that again the network lost accuracy. The network still performs well in some cases as seen in Figures 16 up to 19.

To quantify the convergence of the RNN, averaged errors of the tactical circles trajectories have been tallied in Table 6 for two variables: \( x \) and \( y \), and are given an error averaged over all the tactical circle manoeuvres.

Table 6: Average of tactical circles error measures
(network trained with noisy data and no noise simulation inputs)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Absolute Error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>385 m</td>
<td>27.4</td>
</tr>
<tr>
<td>( y )</td>
<td>385 m</td>
<td>27.2</td>
</tr>
</tbody>
</table>

To make some estimates of precision error manoeuvres with the same rudder deflection and approach speeds were compared. For the tactical circles using as inputs noisy data the turning diameter differed by 13.7 - 1391 m or 0.9 - 107%. For the tactical circles using as inputs filtered data (without noise) the turning diameter differed by 7.1 - 1421 m or 1 - 109%.

Moreover, it can be concluded that it is possible to simulate the tests using a network trained with noisy data using filtered inputs but it is necessary to be careful with the intensity of the noise in the velocity input as the results exhibit higher error in the RNN performance for this type of simulations. This fact leads to conclude that either is not advisable to perform the training of the network with high noise in the velocity input or it is advisable to perform the simulations of circle manoeuvres with velocity input of same type of those used for training.

8. Conclusions

The predictions for the training circles trained with high noise present acceptable results. Although the networks have been trained with noisy data it is visible that they have learned the behaviour for the velocities of sway and yaw in the studied cases. The errors for the trajectories averaged over all the training and cross-validation data are 13% or less using noisy data as inputs (when considering only the test manoeuvres, errors for these variables ranged from 9-11%) and 19% or less using filtered inputs for the network trained using noise in the rudder angle input and no noise in the velocity. For the network trained using noise in the velocity input and no noise in the rudder angle the errors for the trajectories averaged over all the training and cross-validation data are 21% or less using noisy data as inputs (when considering only the test manoeuvres, errors for these variables were around 15%) and 27% or less using filtered inputs.

Moreover, it can be concluded that it is possible to simulate with reasonable accuracy the tests using a network trained with noisy data, i.e. either using noisy inputs or filtered inputs. To get lower errors in the estimation of the trajectory is advisably do not train the neural network with high noise in the velocity input as it was shown that this noisy input affects more the performance of the simulator than the rudder angle input. The results of the study show that the neural network behaves well when trained with noisy data what is very important in order to estimate how accurate can be the prediction of trajectories made with a model trained with real data obtained from full-scale trials. The RNN showed to be adequate for problems where the training data corresponds to noisy data, as can be the case of the data achieved on board of ships.

The values of the errors obtained for the test manoeuvres shown that the network has good generalization ability to data different from training, what is very good for the simulation model.
One improvement to get better accuracy can be to introduce a greater number of simulation runs for training. Recursive neural networks have demonstrated ability as a robust and accurate manoeuvring simulation tool and as a noise-tolerant functional relationship.

Acknowledgements

The first author has been financed by “Fundação para a Ciencia e Tecnologia” under contract number SFRH/BD/6437/2001.

References


Modern Procedures in the Structural Design and Production of Ships

Geert J. Tepper, Numeriek Centrum Groningen B.V., Groningen/The Netherlands, tepper@neg.nl
Theodoor de Jonge, Numeriek Centrum Groningen B.V., Groningen/The Netherlands, jonge@neg.nl

Abstract

Today NUPAS-CADMATIC is one of the major 3D CAD/CAE/CAM systems used in the international shipbuilding industry. The system is developed because our engineers could not find suitable software at the market to suit their needs. From a simple 2D system, NUPAS-CADMATIC developed into a mature, full scope, 3D system.

Our philosophy is to team up with specialists in every field and integrate with the best available software. You cannot be the best in every field yourself. We found this a very successful formula, which is also widely recognised by users of our software.

This has brought us were we are today. Developments that distinguish NUPAS-CADMATIC from other software are: integrated design, cost driven shipbuilding and collaborative engineering. In addition, we made the first start with rule-based engineering that assists the engineers in following the rules of class societies. We can also state that virtual shipbuilding is reality today, and has never been so much within reach of everybody involved in a ship project.

Concluding we state that “state-of-the-art” today, gives no guarantees for tomorrow. Shipbuilding is top sport, and to keep up, continuously innovative software developments are necessary.

1. Introduction

Numeriek Centrum Groningen (NCG) develops and markets NUPAS-CADMATIC software for the international shipbuilding industry, delivering a complete software solution for 3D integrated design, engineering and production. NCG is a member of the Dutch Central Industry Group (CIG), a global group of companies supplying industrial goods and services to the maritime sector.

In recent years, NUPAS-CADMATIC has achieved a position as a major supplier of 3D CAD/CAE/CAM systems. Nearly 100 firms of engineers and shipyards in more than 20 countries have now implemented NUPAS-CADMATIC as their main engineering and production system.

This paper sets out NCG’s vision of the current state of 3D CAD/CAE/CAM software for shipbuilding and the developments that we can expect in the near future. How modern and effective are the software tools currently used by design offices and shipyards?

2. From drafting to Three-Dimensional Structural Design

Before the computer era, the main reason for preparing drawings by hand was to help workers at the shipyard understand the plan of the ship to be built. The workers knew what they had to do to create all the steel parts for assembling the ship. The work was carried out using the common working methods of the time. Of course, this is something of a simplification, but essentially, there was as little as possible on paper, and maximum reliance on the worker’s experience. With the arrival of new production technologies, the growth of the economy and demand for more and better quality ships, a different approach was required. Computer technology entered the shipbuilding industry.

From the earliest days, there were Computer Aided Manufacturing (CAM) systems especially for shipbuilding. The main aim of these systems was to translate existing (hand-produced) drawings (of the steel structure) into production information for NC cutting machines. Alongside these developments, Computer Aided Design (CAD) systems that focused on calculation programs like hydrostatic calculations were also being developed. However they did not cover the entire ship's engineering in detail. This required CAE (Computer Aided Engineering) software. General 2D drafting systems were developed but they were merely a substitute for the drafting table. General 2D drafting packages did have their benefits compared to paper and pencil, especially in terms of time saved when drawings
had to be modified. However, the output from these systems was still a 2D drawing. In the early Eighties, these systems were the ‘state of the art’.

It was one of NCG’s sister companies, the former Engineering Centrum Groningen (ECG), which opted for a complete different approach during the second half of the Eighties. A Computer Aided Engineering (CAE) solution called NUPAS, developed in collaboration with Numeriek Centrum Groningen, provided the missing link between CAD and CAM.

The reason for taking this step was that ECG was dissatisfied with the systems available on the market for its specific engineering activities for the shipyards in the north of the Netherlands. This local shipbuilding industry has a typical infrastructure made up of shipyards and many subcontractors. This required a very flexible, modular system of computer applications that had to be able to work independently and have a positive impact on the separate specialist stages.

The infrastructure consists of a ‘network’ of design offices, engineering offices, shipyards and subcontractors who are responsible for aspects such as engine room installation, outfitting, accommodation, steel cutting and block assembly.

3. The NUPAS-CADMATIC Philosophy

The development of what is now called NUPAS-CADMATIC started with the development of a new approach to CAE software. Right from the start the basic philosophy was that the end result of engineering activities should be information needed to build a ship: production information (NC cutting data, net weights, C.o.G, piece part lists) and of course also production drawings. So the CAE system should aid the engineering process by generating drawings such as classification drawings, construction plans, shop drawings, but should also include the production information required to produce the ship.

The initial development work resulted in an application for generating 2-dimensional construction drawings, staying as close as possible to the proven working methods. The idea was that the application should work in the way the engineer was used to work: working with views just as at the drafting table, but instead of drawing lines, the engineer generated the information in an interactive way. By selecting a command, the system asked for information about the structure (position, limits, measures, thickness etc.) and generated the structure in the view. Because the system was linked to the three-dimensional hull shape database, the shape in the drawing was 100% accurate. This initial CAE application was one of the first ‘intelligent’ drafting tools that was easy to learn and with which 40% savings in man-hours could be achieved.

The next step was the development of the current 3D Hull Engineering module. For this module we chose for an approach which involves working with a completely 3D product model of the ship. Unlike other systems, there is no need to construct the model on the computer first and extract drawings from it: NUPAS-CADMATIC Hull works the other way around!

The advantage of this approach is that the information in the 3D product model is always up-to-date and ready for production i.e. allowing nesting and NC cutting at the same time as the structure is designed.

In addition, the views and drawings, which are the means of communicating the model, are created from the model itself and are therefore always consistent. This results in better quality drawings. It became feasible to use CAE/CAM in a time-critical environment such as shipyards where engineering, work preparation and production are activities which run more in parallel than sequentially. We designed software to solve this critical process.

The philosophy behind today’s NUPAS-CADMATIC is consistent with the philosophy at the start: to concentrate on activities that offer most savings in terms of man-hours, time and which improve the quality of information. This resulted in developments such as the removal of manual part coding, and

197
new developments for fully automatic part nesting, mathematical nominal shell plate development and controlling NC cutting and bending machines.

However, the arrival of NUPAS-CADMATIC 3D CAE for hull structure in the late Eighties gradually extended the NUPAS-CADMATIC philosophy. The available 3D constructional model offered opportunities for the implementation of 3D engine room modelling and pre-outfitting. Since a ship is much more than simply the steel structure, the benefits of combining 3D hull engineering, piping, machinery and outfitting were quite clear. Pre-outfitting of the engine room and the ship’s piping systems clearly contributed to savings in production and the ship's lead-time. To implement this, NCG teamed up with Cadmatic Oy from Finland, who specialised in piping CAD/CAM systems and who had proven their value in the petrochemical and paper industries. The NUPAS-CADMATIC 3D product model was integrated with this system, resulting in a system that completely defines the ship at the engineering stage in terms of construction parts, pipes, components and fittings.

NCG’s philosophy was extended by buying and integrating specialised modules instead of building everything in-house. By teaming up with specialists and specialist products, NUPAS-CADMATIC was able to become a complete and mature 3D CAD/CAE/CAM solution, which resulted in a solid user base worldwide in a relatively short time. Clients recognise that our approach - co-operation between specialists - produces faster and more sophisticated results.

4. Moving towards Virtual Shipbuilding

Today NUPAS-CADMATIC has reached a position where it offers a complete range of high quality software modules for the design, engineering and production of hull structure, piping, machinery, HVAC and outfitting. At this time, it is state of the art, but what will tomorrow bring? Let us have a look into the future (some of which is already happening today) in terms of visualisation, hull engineering, integrations and concurrent engineering or collaborative engineering.

4.1. Visualisation

Since computer hardware has ceased to be a restricting factor, the visualisation of the 3D model has become an important subject for consideration. NUPAS-CADMATIC currently offers full real-time Walk Around functionality that enables you to actually walk through the ‘virtual’ ship, highlight objects, query objects etc. These new developments have opened up many opportunities for shipyards, ship owners and subcontractors and provide a helpful tool for communication during project management.

NUPAS-CADMATIC recently launched an even more advanced visualisation tool called ‘eBrowser’ which makes Virtual Shipbuilding reality. The eBrowser is based on modern Internet technology. A compact file is generated from the 3D product model that can be viewed with the standard Internet Explorer browser. Because the file is relative small, it can be published on a web server or sent easily by e-mail. The file also includes detailed information about the objects in the original NUPAS-CADMATIC 3D model, e.g. system information, plate material, position IDs, spool numbers, weights, supplier etc. (see also figure 1). The most appealing aspect of the eBrowser is that one does not need the full NUPAS-CADMATIC software in order to be able to browse the 3D model.

The eBrowser comes in a light version (which can be downloaded free from our website) and a professional version which includes additional functionality.

NUPAS-CADMATIC also offers a maintenance module in which the eBrowser acts as the graphical interface for a maintenance database. The development of this module is based on every day routines in maintenance. You can keep record of all equipment and service data, schedule the work needed and control the spare part stock flow. With the maintenance history kept by the system, you can pinpoint areas at any time and keep track of all maintenance costs. Ship owners enthusiastically recognise this approach of re-use the existing ship model for maintenance purposes.
The eBrowser can also be used as an interface for third-party maintenance software. Several combinations have been established already. This gives the end user the possibility to integrate his eBrowser model with his existing maintenance database structure.

![eBrowser running in Internet Explorer](image1.png)

**4.2. Topology and rule-based engineering**

The present 3D Product Model of NUPAS-CADMATIC is a full topological model. Topology means that structure parts in the model are related to other (structure) parts in terms of their geometrical boundaries. In addition to topology among structure parts, structure parts also can have a topology with the three-dimensional hull shape. The benefits of a topological model are the flexibility and the time saved when modifying the construction design or hull shape. (See figure 2).

We have built a clear user interface to prevent the user losing the overview of the inter-relationships. The user is able to do his work without being bothered by complex processes in the background.

Rule-based engineering is a method whereby rules that apply to specific structural properties in the ship are automatically taken into account during engineering. For instance, by defining the minimum permissible thickness for specific decks (laid down by the class societies) or defining tank areas where collars have to be used, the software will automatically apply the rules and assist the engineer during his job.

![A view with topological construction](image2.png)

The techniques of topology and rule-based engineering are not completely new, but the combination with external software packages such as NAPA or the E4 system of Flensburger Schiffbau Gesellschaft is a true innovation. This offers a lot more flexibility.
4.3. Integrated Design

As stated before, not all modules of NUPAS-CADMATIC are produced in-house by NCG. We seek partners with proven products and team up to integrate them with our products. This approach has proved to be effective, and will be intensified in the near future. The NAPA hull shape model has recently been integrated in NUPAS-CADMATIC, and another initial design package - Fairway from the SARC company in Holland - has also undergone this integration. The first steps towards integrating the NUPAS-CADMATIC system with the NAPA Steel program have been taken recently.

4.4. Cost-driven Shipbuilding

During the basic design of a ship, many factors with respect to the construction costs are unknown and initially often estimated from experience of comparable projects. Every shipbuilder's wish is to have the right information available at the right moment to minimise the risks on the project (see figure 3) and to shorten the time needed for basic design. Since most of this information is related to the preliminary structural design, the 3D product model would be a perfect place to store and retrieve this information. Not only information relating to computable data such as weight, paint area, welding length etc., but also about purchase parts (e.g. manhole hatches, stairs, railing etc.) and components.

A simplified form of cost-driven shipbuilding is already possible with the help of the Plantek module, our drafting module that supports functionality to create general arrangements, manholes and doors plans, deck layouts etc. The information stored on the drawings and plans contains attributes that can be extracted in the form of lists.

To take advantage of the 3D product model during this stage of the design process, we initiated the development of a new module called 3D-Ship some time ago. 3D-Ship is a constructional modelling tool using the same product model as the 3D Hull Engineering modules, and is used during the early (basic) design of the ship. 3D-Ship offers complete topology for designing and modelling the (approximate) main structure of the ship, whereby the ship can be divided into a number of primary blocks, e.g. fore, midships and aft. In addition to creating the structure, it also allows the user to define compartments (including their properties) of the ship. The program contains smart and user-friendly functionality to assist the designer in storing as much information as required to reduce the time needed for basic design. This information is not limited to the 3D construction only (to determine factors such as weights, stiffener lengths, welding lengths and paint areas) - it also provides the basis for the necessary arrangements and plans.

One of the most interesting advantages of using 3D-Ship is that all information stored in the product model is directly available for the second phase: detailed engineering. This prevents the engineer from having to redo the work when detailed engineering starts. The 3D topological model is cut into blocks or sections from where the detailed engineering is carried out. An additional advantage of using the same product model for both basic design and detailed design is the refinement of the arrangements

![Fig.3: Design Volume - Risk](image-url)
and plans during the process. The General Arrangement automatically becomes more complete and detailed as time passes.

With 3D-Ship, cost-driven shipbuilding becomes a reality and available to the shipbuilder. By extending this module with appropriate functionality in the near future, it will become a valuable tool for supporting the basic design.

4.5. Collaborative Engineering

Today, at the start of the new millennium, NUPAS-CADMATIC is a complete software solution for shipyards that do everything in-house and for shipyards working together within a subcontractor infrastructure. Due to the modular way in which the system is constructed, many design and engineering offices also use (specific modules of) NUPAS-CADMATIC, resulting in a network of NUPAS-CADMATIC users. For instance in Germany we see a growing spin-off effect where many (smaller) engineering companies are starting to use NUPAS-CADMATIC to service the shipyards.

This form of collaborative engineering requires flexible and easy-to-use data exchange functionality in the software. NCG recognised this requirement several years ago, and implemented services in its software for the optimum exchange of (partial) project data. The 3D Hull Product Model is constructed in such a way that 3D block constructions, the hull shape database, all related drawings and the yard standards can be exported / imported between users with a single press of a button. This results in a compact data file that can easily be transferred by e-mail or Internet (see figure 4). The result is that a subcontractor works with the same standards, libraries and procedures as the shipyard.

A new development we expect is the shift from local working to a new form of the server-client principle to support maximum collaboration between shipbuilding partners. The principle behind this server-client method is quite simple: bring all application components (software/data) together in one place and make is accessible by means of easy to use and cheap local client software. In fact, one could say that it is a highly advanced form of the old mainframe using terminals, but now offering the functionality of a powerful PC workstation (see figure 5).

![Diagram](image)

**Fig.4: Data exchange between Nupas-Cadmatic users**

The main advantages are:
1. Accessible by any client independent of his geographical location;
2. Only standard media is required which are getting cheaper by the day;
3. No need for local software/data storage at the client side;
4. Better application performance since the applications run on the server where the data is also stored;
5. Easier project management with respect to application management;
6. No data exchange between parties based on files anymore;
7. Communication from various locations is real-time on up-to-date databases.

The complete project environment (all software applications and data for supporting the shipbuilding process) is situated in one place. This place can, for instance, be a shipyard or an independent com-
mercial data centre providing full services with regard to hardware, (inter)network facilities, maintenance etc. An example of such a server-client solution is Citrix, which is currently in use at Peene-Werft in Wolgast to enable collaboration between the shipyard, engineering offices, subcontractors and the construction site in Berne. Technically speaking, not all applications (especially 3D CAD applications) lend themselves to be used in such a server-client environment. NCG carried out specific software modifications that allow NUPAS-CADMATIC to be used with a Citrix server. The parties involved use existing links such as direct lines, ISDN or ADSL to work on the Citrix server.

The advantage of using a commercial data centre is that they provide the most up-to-date technical facilities. They have the knowledge and are responsible for the performance of the facilities. It relieves the shipyard of the non-shipbuilding activities such system and hardware management so they can concentrate more on their core business, building the ship.

5. Conclusion

“State of the art” shipbuilding software is a relative notion. The development of CAD/CAE/CAM software for the shipbuilding industry is often based on the knowledge and procedures available at the time, and is therefore labelled “state of the art”. Shipyards are constantly searching for tools and solutions that help them to stay ahead of the competition. New developments will therefore continuously follow the demands from the market. However, we think there is also a gap between what a system offers and how it is applied. As a software supplier, we have the mission to constantly optimise the use of the software by our clients.

We are proud of what we have achieved with NUPAS-CADMATIC over the past few years as a relative newcomer to the shipbuilding market. We consider Germany, where shipbuilding has achieved a high level of knowledge and productivity, as a good example of the application of NUPAS-CADMATIC. Because of the high labour costs, efficiency has a high priority. Use of our software spread particularly rapidly in Germany, and it is now widespread. We consider this to be confirmation that NCG's software development is heading in the right direction.
The Virtual Shipyard – Simulation in Production and Logistics at Flensburger

Dirk Steinhauer, Flensburger Schiffbau-GmbH & Co. KG, Flensburg/Germany, steinhauer@fsg-ship.de

1. Introduction

In order to survive in today’s shipbuilding market, it is vital for a shipyard to have optimum utilisation of its resources. Therefore the greatest challenge for a shipyard as a producer of one-of-a-kind products lies in managing the complex relationship between products, production processes and resources. The reasons for the growing complexity are the complex product itself as well as the production characteristics in shipbuilding. Furthermore reduced operating cycles become necessary giving less time for the management of even closer events and increasing the effects of disturbances.

Since 1997 simulation has been used at Flensburger to show and evaluate the complex relationship of products and production processes. This tool allows detecting and evaluating potentials throughout the different levels of planning. The effects of products as well as changed production processes on production flow can be displayed and activities can be initiated in time. Using simulations in shipbuilding allows to achieve and exploit series effects even in one-of-a-kind production.

2. Application Fields of Simulation at Flensburger

For the management of the complex relationships in shipbuilding, the traditional static tools do not obtain satisfying results. Especially the dynamic effects as the result of the high amount of variant parts can only be shown and evaluated in simulations. Simulations, the approved tool in series production had to be extended according to the requirements of a one-of-a-kind production. Especially the focus on the description of the product and the continuous product data flow to simulation had to be developed.

After successful implementation of the simulation in the layout and material flow planning the field of application was extended to the production planning. Today a ship can be produced in the virtual shipyard about one month before the start of the real production, allowing plans to be improved in time. Fig. 1 shows the application fields of production simulations.

![Fig. 1: Application Fields of Production Simulations](image)

The layout planning at Flensburger is pushed by a production development project. Its goal is a great improvement in the productivity keeping the actual work force. Therefore a concept for the future shipyard is being developed which is to be realised by some investment projects. These investment projects are assured by simulations, i.e. before the decision for the investment is made all its function-
ality and influence on the rest of the production have been verified. The application of the simulation in production planning is divided in the phases of strategic and tactical planning and operative control. In strategic planning, a new order is planned in the early design phase. Building methods and sequences have to be defined. The tactical planning tends to the optimisation of the plan for the next weeks in certain production stations. In this phase the parameters are sequences and manning level. In operational planning foremen on the shop floor react on actual changes (e.g. breakdowns). The simulations in the field of logistics are to solve logistical problems in the shipyard or on board of ships. By simulation, an optimum design according to the aboard-logistics with minimum costs can be offered to the customer.

3. The Simulation Toolkit at Flensburger

The simulation technology in production and logistics was especially developed according to the requirements of the automotive industry. For this industry several functionalities are available on the market in terms of simulation toolsets. An automotive production can easily be modelled by a combination of predefined simulation modules. Neither for the shipbuilding industry nor for the related plant construction simulation modules are available. Therefore an own simulation toolkit for shipbuilding industry has been developed at Flensburger. The simulation modules developed at Flensburger are used in several different projects. They only have to be adapted to the special needs of the model by changing their parameters, Fig.2.

Fig.2: Synergy effects by programming simulation modules

The simulation modules are improved continuously by Flensburger’s simulation team. New functionalities can be easily added to the toolkit and are then available in all existing models. Expensive changes of existing simulation models are not necessary anymore and a lot of mistakes can thereby be avoided. It is also advantageous that new simulation modules can be tested easily in different surroundings before they are released for public use. Simulation modules can be created in parallel so that different people can work simultaneously on a simulation problem.

4. The Simulation Model of the Production

While modelling the production of Flensburger, the product with its parameters is strictly separated from the simulation model of the production, which means that different kinds of ships can be produced in the virtual shipyard. Therefore the simulation model must be universal. The product data contains all geometrical and methodical information about the ship while the simulation model includes all parameters describing the production facilities, resources and processes. One basic model can be used for the different applications of simulation. This universal model is updated continuously for the tactical planning and can be easily configured for the different applications.
5. Data for Simulation

The product data for simulations contain all the attributes of the product and its single components which influence the turnaround time. The attributes are on the one hand the geometrical information like dimensions and on the other hand information about the building method like sequences and assembly descriptions. To achieve an appropriate result from the simulation all single parts of a ship including all the pre-outfitting material are described in the product data. The exact identification, the code for material control, all relevant geometrical dimensions, the weight and the material quality are available for each part. In the course of the simulation these data are attached to every part as attributes. An automatic generation of product data is absolutely necessary, especially when using the simulation for production planning. At Flensburger, different interfaces have been programmed to get access to the CAD data on the one hand and to the planning data on the other hand. Afterwards, these data are prepared for the simulation model in the simulation database. Fig.4 shows the data flow to the simulation.

6. Simulations for Layout Planning

Simulation has been used at Flensburger for the layout planning for five years now. Therefore a model of the whole steel production was created and adjusted for new layout alternatives. One typical ship of Flensburger’s product mix was selected as a reference ship and made available for the simulation model with all its detailed data. The changes in layout can be evaluated by means of the different
turnaround performance. Meanwhile the alternative layout is developed by a team of employees from the production, the planning department and the production technology group.

The New Panel Line as one Example of Use:
Within the shipyard development project a concept for a new panel line has been created. This concept contained a completely new combination of plants in restricted space and was assured by the simulation. The new panel line was implemented in the summer of 2001 and fully reached the required performance. Fig. 5 shows the simulation model of the new panel line. On this production line plates are welded together and afterwards stiffened by profiles. Additionally outfitting materials like lashing foundations are assembled and welded here.

Fig. 5: Simulation model of the new panel line

At the time of panel line development the C-Box container vessel was the actual type of ship at Flensburger. But the upcoming RoRo-ship has also been produced on the virtual panel line to be compared to the container ship. Fig. 6 shows the different utilisation while producing the different ships. Explicit differences can be seen in that output data. The plasma cutting machine is the bottleneck producing the container ship. This bottleneck moves to the profile erection and welding in the production of the RoRo-ship.

Fig. 6: Utilisation of the panel line at different types of ships

7. Simulations in Production Planning
Today, the simulation model developed for layout planning is also used for the planning of the future production programme. The latest data about the orders to be produced in the next weeks are continuously generated, allowing the production programme of the next two month to be simulated with realistic data to support the tactical planning. Through the simulation the planner can verify his plan and improve it if necessary. He has got a special user interface at his disposal where he finds the parameters he has influence on like sequences and personnel capacity.
The goal of simulation in production planning is to have an optimum plan at the time of production and to tap the full potential that can be detected before production starts.

Planning Tool for the Panel Line and Section Assembly:
After the end of the investment project for the new panel line, the simulation model was available for the tactical production planning. The model was extended to meet the requirements of this application and implemented into the planning within a project.

The plan is regularly tested for its feasibility by the planner or the foreman. The results are then used to define the required manning level for the panel line and the section assembly. The following steps are necessary to verify the plan:

1. Adjustment of parameters
2. Run of the simulation
3. Interpretation of the results

If the results do not meet the target requirements, another cycle of the three steps has to follow.

1. Adjustment of Parameters
Before the actual simulation run the simulation parameters have to be adjusted. The foreman or the planner enters the planned or the available manning level on the simulation database. He has got a special dialog where he can dispose the workers by qualification and shifts.

2. Simulation Run
The settings of the simulation database, the actual product and planning data and the actual production dates are read into the simulation model. The simulation run is carried out and after reaching the end date the simulation results are saved.

3. Interpretation of the Results
For the evaluation of the simulation run several methods of analysis are available. First of all there can be checked if the schedule is met by the simulation. The influence from one production station to another can also be shown by a graph. If there is not enough buffer to feed the next station properly it can be seen in the graph. If the schedule is not met by the simulation, bottlenecks can be detected. There are utilisation diagrams of the different plants for each week and the station with the biggest part of working time can be called the bottleneck. Furthermore the utilisation of the personnel can be evaluated per week. A low utilisation of workers of a special qualification may indicate the possibility to reduce the manning level.

8. Simulation of Onboard-Logistics
At Flensburger the simulation is also used to evaluate the logistics on a ship, in the harbour or in a fleet. The goal is the optimum design of a ship from the logistic point of view. The loading and unloading of ships sometimes requires a complex logistic especially when there are chains of different transportation means or several vehicles in parallel use blocking each other. In these cases simulation can be used to calculate the time for loading, to show the bottlenecks and to evaluate alternatives. There have been many successful simulation studies of loading or unloading equipment of ships or evacuation studies for passenger vessels at Flensburger. Bulk carriers, RoRo-ships, container ships or paper carriers have been loaded or unloaded virtually in the computer to get the optimum design.

Loading of a Paper Carrier:
A ship owner enquired a paper carrier once at Flensburger with a complex loading equipment. The below deck was to be loaded from the pier by conveyors, lifts and other conveyors. Down in the deck the paper rolls were to be distributed by several fork lifts.
For this concept the required amount of transportation means (conveyors, lifts, fork lifts) had to be defined. The suggestion of the ship owner contained three lifts and three conveyors flush with the below deck. First suggestions of Flensburger's design department showed the possibility to resign one
of the lifts and all of the deck conveyors which would mean less costs for the customer. The logistic
of the paper carrier was modelled in the simulation, Fig.7. The amount of lifts, the deck conveyors
and the amount of forklifts could be varied. The process times and relationships were taken from the
ship owner's experts and from a video. The result was the time for loading the whole deck with paper
rolls. The results for the different alternatives were collected and visualised for two different paper
roll's diameters, Fig.8.

Fig.7: Simulation model of loading a paper carrier

Fig.8: Analysis of the paper carrier's loading simulation

The graphical analysis shows that the third lift has very little impact on the time for loading. The deck
conveyors have almost no influence because the bottleneck of the system is the lift transport. For the
bigger diameter of paper rolls the optimum number of forklifts in the deck is three.

3 elevators
3 forklifts
3 deckconveyors
The simulation proved that one lift and the deck conveyors are not necessary on this paper carrier. This would save the ship owner approximately $1,300,000 per ship.

9. Perspective

In the near future application of simulations will be more and more integrated into the early design phase. In the framework of the programme "Design in Seven Days" - firm offer after seven days - the aim is "simulation on the fifth day". The product data in the early design phase will be generated from the design tool for the planning and the simulation system. The aim is the support not only of planning the building programme and of the scheduling, but also an optimisation in the design according to the requirements of the production.

At the same time the simulation will be used to a greater extent in production planning and control. The aim is a tool for the daily production control on the shop floor by the foreman. This has to be based on a continuous and automatic acquisition of the actual production data on the shop floor. The production area covered by simulation models will be extended from pre-production to the block assembly and the final assembly on the slipway.

Previous activities concerning simulation have been focussing on steel production. The outfitting of the ship has just been mentioned if it took place in the steel production. Since the beginning of the year 2000 the outfitting processes were analysed and visualised analogous to the processes in steel production. In the outfitting process there are much more degrees of freedom due to big variety of possible material flows and sequences. A public funded research project has been started for to develop simulation modules for the outfitting of ships.
Artificial Intelligence for Automatic Container Stowage Planning Optimisation

Samuel E. Kemp, Paul A. Roach, Andrew J. Ware, Ian D. Wilson

School of Computing, Univ. of Glamorgan, Pontypridd/UK, sekemp@glam.ac.uk

Abstract

Container ships visit many ports along a route, and transport large numbers of containers of varying dimensions and contents. The unloading of containers destined for each port, and the loading of containers destined for subsequent ports, is arranged to follow pre-determined stowage arrangements and to make efficient use of cranes. These arrangements are termed stowage plans, and the problem of determining the best arrangement of containers within a container ship, on leaving a given port, is called the container-ship stowage problem.

Most published work on stowage planning has focussed on semi-automation and tools used to assist stowage planners. However, full automation of the production of solutions to the stowage problem has received some attention, with a recent growth of research in the area. Approaches to full automation have used techniques as varied as the comparison of plausible plans by simulations of voyages, meta-heuristic search, and (most commonly) the application of linear programming techniques to a mathematical model of the cargo-space. However, many published approaches have relied on simplifications of the problem domain which have rendered them of no commercial value. This paper explains and compares a range of approaches taken by the authors to the solution of the container-ship stowage problem, which focuses on the reliable production of valid, sub-optimal solutions. These include the strong decomposition of the problem into different conceptual levels of planning to which branch & bound, Tabu search techniques and Genetic algorithms have been applied. A particular focus of the paper is the relationships between the methods of solution and the corresponding models of cargo and stowage spaces, and the consequences that these models have for the accuracy and usefulness of the solutions produced.

1. Introduction

Since the 1970s, containerisation (the packing of cargo into large, dedicated boxes, of different dimensions, enabling multiple units of cargo to be handled simultaneously) has facilitated transportation. The increase in size of container ships has enabled companies to benefit from the economies of scale effect. The increase has typically been from small 350 Twenty Foot Equivalent Units (TEUs) to ships with capacities of more than 4500 TEUs.

Container ships travel on “round-robin” routes. At each port of destination (POD) along the ship’s journey containers may be unloaded whilst additional containers destined for subsequent ports may be loaded. Determining which legal configuration of containers best facilitates this process, in a cost-effective way, comprises the container stowage problem.

Determining an acceptable arrangement of containers is an error-prone process that traditionally relies on the intuitive skills of human planners. The planner must determine a suitable arrangement of containers so that constraints are satisfied and handling costs are minimised. Therefore, the planner utilises several planning documents in order to achieve this. These documents can be separated into two main groups – long-term (generalised) and short-term (specialised) stowage strategies. Planners must consider expected loads at subsequent ports that often include statistical information describing loads in generic terms. Planners use a combination of documents (the General Arrangement, Outline Plan and the Bay Plan) to plan the stowage of cargo.

Therefore, if it is possible to model (by employing a divide and conquer technique) the processes that human-planners use in order to produce these documents then Artificial Intelligent (AI) paradigms
could be reasonably applied to match, or, even better the results gained by the human planners. This paper focuses on the complexity of the container-ship stowage problem before showing how human planners break down the problem into more manageable sub-problems. An introduction and evaluation on the application of Tabu search and Genetic Algorithm techniques to these sub problems is also considered.

2. Problem Size & Complexity

A large container ship can require vast amounts of container movements (i.e. loading, unloading, repositioning) at each POD. It is important that container movements are kept to a minimum in order to maximise profitability, whilst not hindering the discharge and loading process.

Container ship efficiency is determined by the arrangement of containers within the container terminal and on the container ship itself. Any system must be able to determine the optimal arrangement of containers so that all constraints (restrictions placed upon where and how containers can be stored) are satisfied and material handling costs are minimised. Additionally, minimising the number of re-handles will greatly reduce operating costs and improve efficiency. A re-handle is movement of a container, which is only required in order to access another, or to improve a stowage configuration for subsequent ports.

The complexity of stowage planning is increased further by its multi-port nature. Therefore, a plan for a stowage configuration at one port must assimilate the consequences at subsequent ports. Otherwise, the problems associated with constraints being violated and re-handles increase in likelihood later in the ship’s journey.

In summary, the fundamentals of the container ship stowage problem is the determination of a stowage configuration for a container ship, on leaving a port, so that ship stability and stress constraints are not violated and container re-handles are minimised.

Given these requirements and the nature of the problem, the container stowage problem is described as a combinatorial optimisation problem. The size of this problem largely depends upon the ship’s capacity (given by the number of TEU units) and the container supply and demand at each POD. The container stowage problem is combinatorially explosive with the number of possible stowage configurations for a medium-sized container ship being vast. Dillingham and Perakis (1986) found that the number of possible configurations for a 2000 TEU ship is approximately $3.3 \times 10^{5735}$. If it is assumed that a computer can evaluate the cost of each possible configuration in $10^{10}$ seconds (which means that in one second the computer can evaluate $10^{10}$ solutions) then the time taken for the computer to evaluate all configurations would be:

$$\frac{3.3\left(10^{5735}\right)}{10^{10}} > 1 \text{ Trillion Years} \quad (1)$$

As (1) shows, although enumerating all possible configurations will guarantee an optimal solution its processing time renders it an unfeasible method. This has led to the problem being described as NP-Hard (Botter and Brinati (1992), Avriel et al. (1998)) meaning that it is impossible to guarantee an optimal solution in a reasonable amount of processing time. Evolutionary search techniques (such as tabu search and genetic algorithms) have been successfully applied to hard combinatoric search problems like the container stowage problem. Although these techniques do not guarantee an optimal solution, they have been shown to produce good sub-optimal solutions in a reasonable amount of processing time.

The success of any evolutionary search method will be determined by the problem decomposition. An over simplification of the problem will render it of no commercial value; conversely, increasing the complexity of the problem can unnecessarily increase the search space.
3. Problem Decomposition

The first stage of the approach taken by human planners is the strategic planning process. Here generalised containers are allocated to a blocked cargo-space in which slots corresponding to hatch-lids are grouped together. The second of the two stages is the tactical planning process. Here the specific containers are allocated to specific slots within the blocks determined during the strategic planning phase.

3.1 Strategic Planning Phase

Human planners use a General Arrangement Document (GAD) in order to allocate groups of containers according to destination, length and content. The GAD is an abstraction of the vertical longitudinal section through the centre of the vessel, viewed from the starboard side. The planner ‘reserves’ areas of the ship to hold groups of containers destined for the same port. To understand the task an awareness of the cargo-space (indicated by the document) and the containers is required.

The cargo space of a container-ship is made up of cells (20’ long, 8’ wide and 4’3’’ high). Cells are grouped into vertical stacks, which are grouped into bays (collections of stacks across the width of the ship). Bays can either be above-deck or below-deck (enclosed within the ship beneath removable hatch-lids) and are grouped together by an associated hatch number. Planners must consider both the physical dimensions and the contents of the cargo in relation to the cargo-space. Some cargo may have special stowage requirements (e.g. a power supply for either cooling or heating). Other cargo may be defined as hazardous and thus have specific rules applying to them such as segregation from other cargo.

Cargo stowage planners must also ensure that the vessel remains in a stable condition using intact stability calculations (Goldberg, 1980). It is essential that cargo weight is spread evenly across the vessel to avoid heeling (an inclination from the vertical towards port or starboard). Uneven weight distributions also produce forces that can affect the vessel’s physical structure such as bending (acting from bow to stern) and torsion (port to starboard). Ballast (seawater) can be used to stabilise the vessel but is considered an undesirable addition of cargo that requires minimisation.

The positions of containers in the GAD depend on the existence of other cargo for the same destination, permitted length of containers within each hatch, provision of special cargo and the number of cranes at each destination.

Therefore, the strategic planner’s goal is to ensure maximum crane usage at each destination and that any constraints are met. To achieve this, containers should be spread across the vessel in a number of hatches which is a multiple of the number of cranes at each destination. Sufficient space is given between the hatches to allow simultaneous crane usage.

A sub-process known as outline planning is carried out in the strategic planning phase. This is used to allocate containers within hatches on the GAD to above or below deck stacks. It must be stressed that specific containers are not allocated a cargo space but rather a container of a particular general class is allocated to the space. The allocation of a particular container of such a general class to a cargo space is made later in the tactical planning phase. The primary objective of outline planning is to minimise the removal of hatch-lids whilst minimising the amount of unused below-deck cargo space.

3.2 Tactical Planning Phase

This is the second stage of the process that human planners perform to determine the stowage locations for specific individual containers. The general plan outlined in 3.1 is used to guide specific placements of containers into specific slots. During tactical planning a number of containers may still be enroute to the container-terminal. Therefore, individual bay plans are documents that are prepared incrementally by human planners. New bay plans are generated (or refined) as containers become
available for loading.

The result at the end of this phase is a set of detailed bay plans that show the precise stowage configuration of the vessel.

Now the problem has been divided into the manageable sub-problems outlined in 3.1 and 3.2, search techniques can now be applied to finding an optimal configuration. Tabu search and Genetic algorithms (GAs) will be considered in this paper because they have been successfully applied to the container-ship stowage problem.

4. Tabu Search

4.1 The Theory of Tabu Search

Tabu search (Glover et al., 1993) is a well-known local search technique, which uses various rules to guide the choice of neighbours in the search space. A neighbourhood, \( N(s) \) of a given candidate solution, \( s \), can be partitioned into two subsets \( T(s) \) and \( NT(s) \) where \( T(s) \subseteq N(s) \) contains those neighbours which are tabu and \( NT(s) \subseteq N(s) \) contains those neighbours which are not tabu. Tabu rules are a way of classifying whether a neighbour \( q \in N(s) \) is tabu or not. A neighbour \( q \in N(s) \) is tabu if:

1. It involved a move from \( s \), which has occurred recently (e.g. last 5 iterations). This is called recency or short-term memory.
2. The move from \( s \) to \( q \) has occurred frequently in the search so far (e.g. 30% of the iterations executed so far). This is called frequency or long-term memory.
3. \( F(q) \), where \( F \) is the cost function gives a value found over a number of previous iterations.

The use of the recency and frequency rules help prevent an undesirable phenomenon called cycling, which can occur in other heuristic approaches such as simulated annealing. Cycling is where a particular solution is repeatedly selected because of its “good” cost value. However, this can keep the search in local optima as it neglects other areas of the search space, which may lead to the global optima not being found.

Sometimes it may be beneficial to override the tabu rules and select moves from the tabu set, if some aspiration criteria are satisfied. For example, if a tabu neighbour gave the best solution found so far it would be beneficial not to ignore it. Typical examples of aspiration criteria used are:

1. If all neighbours are tabu then select the “least” tabu solution.
2. \( F(q) \) is the best value found so far in the search.
3. \( |F(q) - F(s)| > \beta \), whereby \( \beta \) is a pre-determined value.

The tabu search algorithm is stated below

Set \( j = 1 \)
Generate initial solution \( s_j \) (possibly at random)
WHILE
   - Generate the neighbourhood set \( (N(s_j)) \)
   - Find the tabu neighbours in this set \( (T(s_j)) \)
   - Find the aspiration set using the aspiration criteria \( A(s_j) \subseteq T(s_j) \) and choose the new solution, \( s_j + 1 \) from: \( (N(s_j) - T(s_j)) \cup A(s_j) \) such that \( s_j + 1 \) is locally the best value.
   - \( j = j + 1 \)
END WHILE

4.2 Tabu Search in Practise

Wilson (1997) advocates a two-phase approach (i.e. strategic phase and tactical phase), which attempts to model the process used by human-planners. This approach assumes the following:

(a) At each POD, unloading and loading has occurred, but the latter did not begin until the former had finished.
(b) The user sets ballast conditions.
(c) Two cranes were available for loading and unloading at each POD.

Tabu search is applied to the tactical planning phase (Wilson and Roach, 1999). However, to begin with branch & bound is utilized to generate a general representation so that all containers are allocated to individual blocks, rather than specific cells. Using this framework avoids the combinatorial complexity of attempting to make specific placements within the entire cargo-space.

General containers to be loaded at the current POD are ordered by those having the fewest available legal stowage locations and the furthest POD first. There is need to define a cost function which can quantify ‘goodness’ of block stowage and crane usage. The cost function therefore, is the weighted sums of these functions the exact weighting of which depends on the shipping operator practises, the vessel, the route and the number of cranes at each POD (see Section 5.2.2 for a full cost function definition).

It must be noted that at this stage in the planning process specific containers are not allocated to specific cells since the goal is to select the best overall generalised solution. The generalised solution does not only reduce the combinatorial complexity of the strategic planning phase but also reduces the neighbourhood associated with a given configuration during the tactical phase. Before the general solution can be optimised each container must be allocated (heuristically) a slot. Therefore preparing an initial specific loading configuration gives a starting point from which the optimum solution can be determined.

A container can heuristically be allocated a slot by applying the following packing algorithm which is designed to sequence containers into blocks (Wilson, 1997). For each block:

1. List containers according to size (large first), and then by destination and weight (furthest and heaviest first).
2. The first container is taken from the list.
3. A standard dimension container is loaded into the first available slot and non-standard dimensioned containers are swapped (where possible) with containers for the same POD at the top of a stack. A container so displaced is returned to the list to be placed somewhere else.
4. If the list is empty then the placement procedure is terminated, otherwise the process begins again from 2.

By applying this packing algorithm to each of the blocks results in a stowage configuration that is near optimal (for that block), which gives a good foundation for the tabu search to optimise. This approach excels at producing good weight gradation stacks, low mixing of PODs in stacks and enforcing non-standard dimensioned containers to be located at the top of stacks.

4.3 Computational Results of Tabu Search

Results were obtained on a 166 MHz Pentium with 40 MB of memory using Allegro Lisp to encode the blocking and GFA (a PC-based 3GL with a high degree of functionality and graphic display.
features) to encode the specific placement algorithm. A generalised solution was obtained in approximately 90 minutes, whereas specialised solutions for each block were obtained in less than one hour. The search space for any given problem is dependant on the vessel capacity and the number of ports; however, the blocking of cargo-space is believed to ensure that solutions of acceptable quality can always be generated in a reasonable amount of processing time.

Empirical studies (Wilson & Roach, 1999) on the optimisation of stowage in individual blocks by tabu search has showed that optimum solutions for below-deck blocks can be found in as few as 15 iterations and a recency list (Glover, 1977) of one move. For above-deck blocks the number of iterations increases (in the worst case) to no more than 200 and in the recency lists 7 moves are required. This can be explained by the variations in container length and hazardous cargo segregation requirements.

5. Genetic Algorithms

5.1 Theory of Genetic Algorithms

Search based on evolutionary models, such as evolutionary programming (Fogel, Owens and Walsh, 1966) had been tried before Holland’s (1975) introduction of genetic algorithms (GAs). However, these models were purely based on mutation and were not notably successful. The foremost difference of modern day research is an emphasis on natural selection and the inclusion of a “crossover” operator to mimic the effect of sexual reproduction.

All GAs consists of six main components:

1. **Representation of the problem**: The term chromosome is used to describe a “legal” solution to the problem. It is composed of a string of genes.

2. **Initial population**: Once a representation has been chosen then it is necessary to create an initial population with which to begin a search. This can be created randomly or using some problem specific information.

3. **Fitness Function (or cost/objective/penalty function)**: This is defined so that a test can be applied to all chromosomes for suitability.

4. **Selection**: This is a process whereby chromosomes are selected from the population for reproduction (to create new, different chromosomes). Two chromosomes (parents) are chosen, which are used by crossover and mutation to produce two new offspring for the new population. Selection is based on Darwinian evolution (*i.e.* natural selection), therefore the fitness of an individual is proportional to the probability that it will reproduce effectively. Hence selection is based on fitness, the higher the fitness value the higher the probability of it being selected.

5. **Crossover**: This is where the genes from each parent are being combined to form offspring. Two parents crossover to produce two offspring that will effectively replace them in the new population. A crossover rate is usually applied to restrict the number of selected pairs of chromosomes that have to undergo crossover. A crossover rate of 1.0 means that all the selected chromosomes undergo crossover, *i.e.* none of the present chromosomes are carried through to the next generation.

6. **Mutation**: The purpose of this is to introduce some kind of random element. If crossover is used on its own to produce offspring then sometimes problems can arise, for example, if all chromosomes have the same gene in the same position then all future offspring will have the same gene at this value. The mutation rate is typically about one gene in every thousand chromosomes tested. Each gene in each chromosome is observed independently and checked for possible mutation by generating a random number, *q* in the range 0 to 1 (inclusive). If $q < \frac{1}{1000}$ then the gene is changed.
This completes one generation (cycle) of the genetic algorithm. The fitness of each chromosome in the new population is evaluated and the whole procedure is repeated. This process continues until some pre-determined criteria are achieved by the GA. Examples of such criteria include a set number of generations or the standard deviation of the population’s fitness exceeding a given threshold.

5.2 The Genetic Algorithm in Practise

5.2.1 Representation of the Problem & Initial Population

The chromosome will be a list of all the containers in the load list and their associated TEU value the ordering is such that chromosomes can be mapped to available space. For each hatch there are two associated TEU capacity values (above-deck and below-deck). Therefore, containers are loaded into a given hatch up to and including the TEU capacity of a given hatch but are not allowed to exceed this capacity. The initial population is generated by randomly ordering the chromosomes.

5.2.2 Fitness Function

The stowage objectives of the strategic planning phase are to:

- Minimise the number of bays occupied by each POD, and the number of PODs in each bay.
- Maximise the number of cranes in operation at each POD.
- Minimise the number of re-handles and hatch- lids moved.
- Minimise the number of cargo blocks occupied by containers.

Wilson and Roach (2000) suggest the following fitness function:

\[
    \text{fitness} = [(f_1 \cdot 3) + (f_2 \cdot 1) + (f_3 \cdot 4) + (f_4 \cdot 3) + (f_5 \cdot 10) + (f_6 \cdot 4) + (f_7 \cdot 3)]
\]  

(2)

In (2) \( f_i \) is a measure of one factor of the solution and the weight is the relative importance of that factor (the weights given in (2) have been assigned through empirical analysis). A low value of \( f \) indicates a good solution.

The factors \( f_1 \) and \( f_2 \) of the fitness function are concerned with the production of good block stowage (stowing together containers destined for the same POD) creating efficient hatch-lid movements.

\[
    f_1 = \sum_{i=1}^{nd} \sum_{j=1}^{nh} DH_{ij}
\]  

(3)

\[
    f_2 = \sum_{i=1}^{nh} \sum_{j=1}^{nd} DH_{ji}
\]  

(4)

In (3) \( DH_{ij} \) is 1 if a container exists with destination \( i \) within hatch \( j \) else it equates to 0, \( nd \) is the number of PODs on route and \( nh \) is the number of ship hatches. Thus, \( f_1 \) calculates the number of hatches occupied by containers of each POD and \( f_2 \) calculates the number of POD that exists within each hatch. The factors \( f_3, f_4 \) and \( f_5 \) measure the validity of efficient crane usage. They are defined as follows:
\[ f_3 = \sum_{i=1}^{nd} \left( \sum_{j=1}^{nh} DH_j \right) \cdot c_{ri} \]  

(5)

In (5) \( c_{ri} \) is the number of cranes at destination \( i \). So in (5) we are comparing the number of hatches occupied by containers for each POD with the number of cranes present at the POD.

\[ f_4 = \sum_{i=1}^{nd} |\mu_i - \phi_i| \]  

(6)

In (6) \( \mu_i \) is the highest number of containers with destination \( i \) stowed within any of the hatches and \( \phi_i \) is the total number of the containers with destination \( i \) stowed minus \( \mu_i \). (6) states the spread of containers between hatches. A ‘good’ spread will allow all cranes to be used simultaneously throughout the loading and unloading process.

\[ f_5 = \sum_{i=1}^{nd} \sum_{j=1}^{nh} \psi_{jk} \]  

(7)

In (7) \( \psi_{jk} \) is assigned 1 if there is a container with destination \( i \) within hatch \( j \) and within adjacent hatch \( k \). (7) will penalise stowage configurations where containers of a particular destination are stowed in adjacent hatches, therefore preventing the two cranes from working simultaneously.

\[ f_6 \] and \( f_7 \) measure container re-handles, they are defined as follows:

\[ f_6 = \sum_{i=1}^{nh} \sum_{j=1}^{nh} \sum_{k=1}^{nd} \sum_{l=1}^{nd} \Omega_{ijkl} \]  

(8)

In (8) \( \Omega_{ijkl} \) is the number of containers stowed on hatch-lids, beneath which are containers destined for an earlier POD.

\[ f_7 = \sum_{i=1}^{nh} \lambda_i \]  

(9)

In (9) \( \lambda_i \) is the remaining capacity below-deck for hatch \( i \) where containers are stowed on-deck. (9) calculates the number of empty spaces below a hatch-lid which support containers that are unavailable without first removing both the hatch-lid and any containers stowed on it. A high number of empty spaces indicate poor stowage.

### 5.2.3 Selection, Crossover & Mutation

A linear fitness rescaling (Coley, 1999) technique was employed, which prevents a chromosome with a high fitness score being selected repeatedly. Repeated selection can cause two parents to be the same and so only mutation can produce changes. Under these circumstances, the population would remain relatively unchanged during the course of the search, which would be undesirable. Roulette Wheel (Coley, 1999) was the chosen chromosome selection operator as it favours the fittest individuals without excluding others.

Partially Mapped crossover (PMX) was used as the crossover operator. This is because standard crossover produces illegal configurations (This is where some containers can be left out of a
configuration and/or some containers can be repeated in a configuration). PMX works by swapping a
given sub-string of the two parents and then adding any containers that do not produce conflict. If any
conflicts do exist then they are mapped to a container that does not produce an illegal configuration.

A chromosome with a good fitness score has strongly blocked stowage. Swapping too many pairs of
containers is likely to cause a loss of blocking, therefore creating a chromosome with a worse fitness
score. If a high mutation rate is set it is likely that large numbers of the population with good fitness
scores will transform into ones with a poorer fitness level. However, mutation is the key to a diverse
population. It can be shown from empirical analysis that the best rate is between 1% and 2%.

5.3 Computational Results of the GA

The results were obtained on an 800 MHz PentiumIII with 128 MB of memory using Visual C++
under Windows NT. The stowage objectives outlined in 5.2.2 were used to test the suitability of all
chromosomes. From empirical work, a population size of 4096 and a crossover rate were chosen. The
number of generations produced from an experiment was based on the time taken to produce one new
generation and a realistic time taken to produce a stowage plan configuration. Therefore, it takes 1.8
seconds to produce one generation and 50 hours was chosen as a reasonable processing time to
produce a stowage plan, leading to 10,000 generations.

Empirical analysis showed that even randomly ordered chromosomes making up the initial population
had a quick convergence, with relatively strong blocking becoming evident after approximately 2000
generations. Reasonably good solutions were determined at about 6500 generations, or 3.28 hours
(using the hardware outlined), however, further improvements in population fitness were extremely
slow after this point.

6. Conclusion

Modelling how human planners solve the container-ship stowage problem gives a detailed insight into
the problem itself. Other published work to tackle the problem using search heuristics has neglected
many of the details that underly the problem and therefore has over-simplified the problem (e.g. by
assuming all containers are the same size, by assuming that there is no hazardous cargo). The
methodology presented in Section 3 highlights the inherent complexity of the problem and provides a
good platform on which to use search heuristics to find a good configuration of containers in the
cargo-space. Indeed, Tabu search and Genetic algorithms have been applied effectively using the
methodology outlined and all constraints have been met in the results given.

The most efficient search heuristic proposed in this paper to solve the problem, based on empirical
analysis, is Tabu search. This is because a good result was obtained within 1.5 hours compared to 3.28
hours using a Genetic algorithm approach on a machine with considerably more processing power.

An important question now arises: How can the “goodness” of the solutions gained from Tabu search
and the GA be measured? A quantified measure cannot be given because this would require a
knowledge of the global optimum solution, which given the combinatorial complexity of the
problem is impossible to know. However, a comparison between human-planners and the techniques
outlined would provide a qualitative measure. The heuristics used and the plans output are reported by
industry experts as being comparable with those of human-planners (Wilson and Roach, 1999).
Furthermore, because the approach described is automated it allows the consideration of more stowage
plans than of human-planners in the time available.

Now that a proven methodology is in place, it is hoped that other search heuristics such as simulated
annealing and an enhanced GA (e.g. use of evolution strategies), will be used in order to improve on
these results.
References


Evaluation of the Effectiveness of a Fuzzy Logic Software Agent to Aid Design Team Negotiation and Communication

David J. Singer, University of Michigan, Ann Arbor/USA, aswell@engin.umich.edu
Michael G. Parsons, University of Michigan, Ann Arbor/USA, parsons@engin.umich.edu

Abstract

Ship design, like many engineering problems, involves both objective knowledge such as mathematical models and subjective knowledge such as expert opinion or design requirements. Fuzzy set theory is a method of systems analysis that bridges the gap between traditional methods of analysis and the real world relationships between variables that are ambiguous. This work focuses on a set-based design approach to the cross-functional design team (Integrated Product Team) activity in early ship design. A hybrid agent analogy with human agents aided by intermediate computer agents is utilized. A new implementation of a fuzzy system has been developed to enable these designers to communicate their preferences over a range of parameter values. The system involves a new input sequence compared to classical fuzzy systems and improves scalability through the use of “meta rules”. The new hybrid agent fuzzy system is unique in that the inputs are the membership functions \( \mu_i(x) \) which are changed by the human agents to express their individual preferences for a particular design parameter as a design progresses. The fuzzification, linguistic fuzzy rule bank, fuzzy inference, and defuzzification components are typical. The input membership functions and the output variable span the set-based range currently under consideration from the parameter’s lower limit to its upper limit. The output is, however, a crisp value function of the design variable indicating the joint preference of the team of human design agents over the current set of that variable. A new hybrid agent design team communication and negotiation assistant software agent has been developed in JAVA using the described fuzzy logic approach.

1. Introduction

New advances in design tools as well as the increased complexity of modern early stage design have led to the need for design team communication and negotiation tools. These new communication and negotiation tools need to invoke processes that utilize the software capabilities available to design teams. The analysis of prior agent-based design experiments reported in Parsons et al. (1999) led us authors to develop a fuzzy logic based agent design tool, Parsons and Singer (2000). Since then we have developed, refined and tested this fuzzy logic agent-based software environment.

The underlining problem that drove the fuzzy logic approach is due to the overall complexity of ship design methodology that translates designer intent over a range of variables. The new approach is intended to increase the probability of reaching a global optimum over the initial design space while keeping the human designers in control of the design decision process. This idea goes against traditional models of preliminary ship design. The traditional approach to design was depicted by Evans (1959) by his “design spiral”. The “design spiral” is also known as “point-based” design. The point-based approach consists of choosing initial design variable values and then revising the chosen values to produce a single design that satisfies the required characteristics of the design. While this technique can produce a feasible design, it may not produce a local or global optimum.

Many industries in the 1980’s and 1990’s began to focus on multidisciplinary team-based concurrent engineering approach to advanced design. The marine industry conducted its own research in this area and developed the ideas of Integrated Product Teams (IPT’s) and Simulation-Based Design (SBD) within the marine framework. Bennett and Lamb (1996) as well as Keane and Tibbitts (1996) have advocated changes in this direction. The explosion of the Internet and Internet based tools in the 1990’s gave un-realized hope of solving the co-location issues associated with the multidisciplinary
team-based concurrent engineering approach. To enable these new Internet technologies and relieve the co-location requirement, new processes need to be developed.

The next evolution in design theory involves a concept called “set-based” design. Set-based design conceptualization followed the study of the Toyota Production System and the formulation of Lean Manufacturing, Womack et al. (1990). The Japanese Technology Management Program at the University of Michigan studied the Toyota design approach, Ward et al. (1995a,b), and identified this unique process. The set-based design process produces world-class designs in a significantly shorter time than other automobile manufactures. The main features of this design process include:

- broad sets for design parameters are defined to allow concurrent design to begin,
- these sets are kept open longer than typical to more fully define tradeoff information,
- the sets are gradually narrowed until a more globally optimal design is revealed and refined.

A simple set-based example is the competition between an audio system and a heating system for volume under the dashboard of a car. Rather than specify in advance the envelope into which each supplier’s design must fit, they can each design a range of options within broad sets so that the design team can see the differences in performance and cost that might result in tradeoffs in volume and shape between these two competing items.

The investigation and use of the different design methods has led us to introduce a hybrid agent approach to conceptual ship design. Agents, in an AI sense, are autonomous software components with elements of perception, intelligence, and adaptability, which are capable of taking independent action. In a network, simple agents can each perform their assigned task and an overall result can emerge from the interactions of the group of agents. Agents can observe system activities and act when necessary.

2. Hybrid Agent Design Approach

Parsons et al. (1999) introduced a hybrid agent approach to team-based early ship design. We believe that problems as complex as ship design will continue to require the expertise, perception, and judgment of human designers. This is accented by the increased upstream use of more complex design tools. These tools require individual expertise that has tended to limit their use to detailed design. Using an agent model, however, designers with the requisite unique skills can be organized and task as a network of agents based upon their technical specialty or particular design role. The fuzzy logic software agent is introduced between these designers to facilitate their communication and design parameter negotiation. This concept results in a hybrid network of human and software agents, Fig.1.

![Agent Network Structure](image)

Fig.1: Agent Network Structure
The network structure in Fig.1 differs somewhat different from the one presented in Parsons et al. (1999), Parsons and Singer (2000). In our initial work, software environment limitations necessitated that each agent only communicate with other agents one at a time, Fig.2. From evaluation of past experiments, it was concluded that this one-to-one communication promotes confusion and inconsistencies. The one-to-one communication, as well as the directional flow of information, caused the design to become skewed by several factors. One such factor is information overload. If an agent needs to have one-to-one interactions for each of its negotiated variables, an agent could need to involve itself in a large number of parallel negotiations with some variables requiring at least 4 unique communication paths. The overload affect causes inconsistencies in the value of variables used by different agents and biases to be introduced into the system due to imposed factors. The necessary asynchronous communication between agents introduces additional variability. The new structure allows all interested agents to communicate and negotiate a variable simultaneously eliminating these problems.

The specific application goal of our previous and current work is the design of a preliminary, parametric, bid-response design for a feeder container ship. The vessel needs to satisfy a set of owner’s requirements that can be generalized by the following requirements:

- Carriage of XXX TEU (Twenty foot Equivalent Unit) with an average weight of 13.0 tonnes with a VCG at 45% of the container height. Uniform loading.
- Endurance of 1600 nm at service speed for fuel, and 20 days for provisions and water
- Panamax length and beam; maximum draft of Y.Y m.
- Service speed at 85% Maximum Continuous Rating on trials of ZZ.Z knots.
- Minimum $G_M$ of at least 0.25 m in the uniform load condition.

To establish starting sets for the primary ship dimensions at the beginning of each experiment, regression equations developed from a database of the world feeder containership fleet where used to estimate $L_pp$, $B$, $D$, and $T$ as a function of TEU and ship speed. The mean value produced by the regression equations ±2SE (regression Standard Errors), which is expected to contain 95.5% of the world fleet, was used as a guide for the initial set. The initial sets for the primary size variables allowed the agents to begin their particular evaluations.

To perform the design in the initial research, human designers were organized and tasked as agents and negotiation markets were developed between relevant agents, Fig.2, Parsons et al. (1999). Seven design agents were utilized in this initial investigation. This number has been subsequently reduced to five to ease the experiment mapping requirements. It was determined that the extra two agents used in the original work were not critical to final design and experimentation.

To facilitate the negotiation between the human agents in the initial experiments, the Responsible Agents for Product/Process Integrated Development (RAPPID) product was used. RAPPID was designed to help human designers manage product characteristics across different functions and stages in the product life cycle, Parunak et al. (1998,1999). The human agents in RAPPID participate in a systematic design marketplace where the goods being traded represent the design characteristics of each of the product components. The markets represent the explicit cost and value of these design characteristics in a common currency, dollars, resulting in a self-organizing dynamic that should yield a competitive design. These markets allow individual designers to make tradeoffs and narrow sets of design characteristics in a way that leads to more globally optimal designs. RAPPID provides two means by which the agents communicate and negotiate. The design agents present their utility curves over the active set of the variable and associate specific currency bids with this utility. Fig.3 shows the two methods in the RAPPID market interface.
The development of the markets that the agents would negotiate provided interesting insight. After the
types of agents and their particular roles were established, we were required to determine what design
variables each agent would need to achieve their assigned goal as well as who would be responsible to
establish each variable and who would negotiate concerning the selection of each variable. The final
development of the market variables led to the decomposition of some classical ship design variables.
An example of this is the length of the ship. In our agent model, the Resistance agent provides the
required speed and is required to evaluate the total resistance of the ship. Since the length of the ship
is a critical determinant of resistance, the Resistance agent needs to know the total ship length and
become a player in the length market. The Propulsion agent’s role is to provide the required thrust
needed by the Resistance agent by choosing an engine and designing a propeller. Due to the
Propulsion agent’s restrictive role he/she is only concerned with the length of the machinery room \( L_m \)
which is a portion of the ship total length. The Cargo agent in the model is required to provide a
capability to carry the needed total number of TEUs. The Cargo agent is also concerned with ship
length, but only needs to know about and negotiate in a length of cargo box \( L_c \) market.
The conceptual design of a hatch-covered, cellular, feeder container ship was undertaken by a team of student design agents to assess the effectiveness of this design approach. The design process converged within one seven-hour design session indicating the promise of a hybrid agent approach in future marine design efforts. A subsequent, parallel investigation using nonlinear programming confirmed that this team of human agents achieved the optimal Required Freight Rate design, Skwarek (1999).

The definition of the agent network and markets, which were required by RAPPID, was extremely useful in developing the hybrid agent approach. The RAPPID product interface and negotiation methodology, however, was cumbersome and not robust in this application. Discussions with the participants in the design experiments revealed that the specific bid price aspects of the markets added confusion and work and were not particularly useful. The human agents, however, successfully used the qualitative utility curves to communicate about the design sets and drive the direction of the design. The utility curves were used to help the agents determine where to begin investigating solutions in the design space. If an agent declared a triangular shaped utility curve, the other agents involved in that market would investigate the region of the design space where the maximum utility could be achieved. Based upon this experience, we began to investigate a simpler and more intuitive way in which the agents could communicate and establish the joint preference for all the agents involved in a particular design parameter selection negotiation.

Parsons et al. (1999) revealed that the hybrid agent approach showed promise as a means of achieving effective conceptual ship design by cross-functional design teams. The approach also helped foster a set-based design approach and provided an effective way to organize a cross-functional team. The negotiation process can also improve the reasoning and cross-functional understanding achieved during the design tradeoff process. Some of the other conclusions included:

- The set-based design paradigm replaces point-based design construction with design discovery; it allows more design to proceed concurrently and defers detailed specifications until tradeoffs are more fully understood.
- Set-based design can significantly increase the number of design alternatives considered.
- Agent effectiveness declines rapidly when they are required to participate in more than about seven systematic markets.

The hybrid agent approach can provide a means to address the potentially limiting design communication and negotiation process in advanced cross-functional team design, particularly if it is virtually linked across the Internet.

This paper presents the results of experiments with a new fuzzy logic computer agent to replace the RAPPID systematic market agents utilized in the initial experiments. The software agent utilizes fuzzy system theory to develop joint agent preferences, which can guide the design analysis and development, set reduction, and negotiation concerning design variable selection.

3. Fuzzy Sets Theory

Real world situations are not precise. A person may wish to characterize the cost of dinner at restaurants. Over the range of prices, a person might have three preference values or sets: cheap, moderate, and expensive. If the dinner is cheap the person is happy. If the meal is moderately priced then the person is neutral. Finally, if the meal is expensive the person could be unhappy. Classically, one might program this model with discrete threshold points at which the price transitions from cheap to moderate and moderate to expensive occur. This approach, however, does not correctly model what happens in the real world since each individual has a different, imprecise range over which these preference transitions actually occur. This natural imprecision and vagueness can be effectively handled using fuzzy set logic introduced by Zadeh (1965).
Fuzzy set theory assumes that an element can belong to more than one set at a time and that its membership in a set is a matter of degree. A parameter’s degree of membership (also called its truth value) in a fuzzy set is determined by the membership function of that set. In practice, triangular and trapezoidal membership functions are typically used because of their computational simplicity. The value of the membership function for any crisp parameter value indicates the degree of membership (truth value between 0 and 1) in the related set. Adjacent membership functions overlap to a certain degree to reflect the fuzziness or vagueness of the set classification. The convention that all membership functions total 1 for any parameter value is utilized here.

![Fuzzy membership functions](image)

**Fig.4: Example of Fuzzy Sets and Membership Functions**

Example, Fig.4: The concept of “tallness” for people depends vaguely upon the height of the observer, the context of the discussion, etc. In conventional logic, a person would have to be classified as either a “not tall man” with membership truth value 1 or as “tall man” with membership 1 (solid lines). In the more general fuzzy logic, the dashed membership functions in Fig.4 could be used. Below perhaps 5’7” a person is clearly “not tall”, above perhaps 6’3” a person is clearly “tall”, and at 5’11” a person might be considered half “not tall” and half “tall” with membership 0.5 in each fuzzy set. There is a gradual transition between these situations between perhaps 5’7” and 6’3”. The vagueness will certainly vary between describing jockeys at a racecourse or players in an professional basketball dressing room.

Fuzzy systems provide a nonlinear mapping between crisp input variables and crisp output variables and allow the use of linguistic IF-THEN expressions for the rules that define the input-output relationship. A fuzzy system consists of four steps: fuzzification of the crisp input parameters to membership in activated fuzzy sets, activation of the appropriate fuzzy rules, the use of fuzzy inference, and finally defuzzification to produce a crisp output. Further details of the type of fuzzy system used here can be found in Kosko (1992), Li and Parsons (1996). Correlation-minimum inference and centroid defuzzification are used in this work. The authors’ fuzzy approach is a unique variation to the classic fuzzy system. The details of this new system are not directly relevant to the presentation of the experimental results and will, thus, not be presented here. Parsons and Singer (2000), Singer (2003) gives further details about the alternative fuzzy approach can be found in Parsons and Singer (2000), Singer (2003).

4. **Software Environment**

The fuzzy logic software environment consists of two main parts; the chief engineers’ interface and the agents’ interface. The chief engineer is the agent that controls the timing of the design as well as the range of the negotiated variables. The fuzzy logic software gives the chief engineer suggestions for the possible range of a variable for a new negotiation round, but the ultimate decision is in the hands of the chief engineer. The chief engineer’s interface contains information about all negotiated and non-negotiated variables for a design. This information includes the upper and lower limits of the active set for a variable for the current negotiation round, the variable value at which the design team’s joint preference curve is at a maximum value, the status of the variable’s negotiation round, and the time left for a given variable’s negotiation round. The chief engineer orchestrates the design
process by setting the range of a variable for a negotiation round as well as the time available to complete the analysis for that range. Fig.5 shows the input screen used by the chief engineer.

Fig.5: Chief Engineer's Input Screen

Fig.6: Agent Input Window

Fig.7: Joint Preference Window
The second component of the system is the Agent interface. The agent interface contains all of the information needed by an agent to achieve his/her design tasks. The main interface lists all of the negotiated and auxiliary variables for the given agent. The main interface information includes the high and low values for a variable’s set range for a current negotiation round, the variable value at which the joint preference curve is at a maximum value, the status of the variables negotiation round, and the time left for a given variable’s negotiation round.

When the chief engineer submits a request for a negotiation round, the negotiation process begins. The agent’s interface will post that a negotiation round has been requested and the time left until the required completion is shown. When ready, each agent involved in the negotiation then clicks on the negotiated variable and constructs their input preference membership functions in a new window, Fig.6. There are two main components of the agent window, denoted by A and B on Fig.6. Area A displays the variable’s current joint preference curve as well as the agent’s inputs for that variable from the prior negotiation round. Area B shows the location where the agent inputs the new preference input membership functions for the variable. The membership functions in this example are denoted with an M-P-M. The notation corresponds to a preference rating of Marginal (M), Preferred (P), and Marginal (M) for the variable over the negotiated range.

After all the required agents have submitted their membership functions for a given variable, the fuzzy logic agent then computes a joint preference curve, Fig.7. The joint preference curve provides two functions. The first function is to guide the chief engineer. The chief engineer uses the zero values of the preference curve as a guide for setting the new ranges for the next negotiation round. The second use of the joint preference curves is to provide guidance to the agents on where they should be focusing their design investigation.

5. Experimental Design

Most research in design methodologies and communication/negotiation within design has not yet utilized careful experimental design. Most of the conclusions made within this field have been based on judgment and survey information. A consequence of this can be seen in the lack of success of some web-based solutions. Web whiteboards and Internet conference solutions have not achieved their suggested impact. If the correct experiments had been planned and analyzed, the researchers might have found out what was truly important in the communication and negotiation within parties.

The questions arise: What are the important factors that are involved in the communications within a design team? What are the important factors that are involved in the negotiation within a design team? Do different treatments have affects on the success of the design? To answer these questions an experiment was designed to establish what factors are truly important, Wu and Hanada (2000).

Experimentation is critical to the discovery of what factors really do play an important role in the design process. An example of this can be seen in the use of email. Does email help or hinder design? We can all agree that email should help design, but what part of email provides the benefit. If we knew specifically what makes email valuable within a design team, we could further develop the positives and take out the non-value added components.

The fuzzy logic hybrid agent experimentation followed the standard pair comparison design. A pair comparison design is used when one wants to compare two treatments. In this case, the two treatments were to complete a design task without co-location using the fuzzy logic agent software and complete the same design task using only Internet chat window communication. The objective of these experiments is to see if the new methodology facilitates set-based design, aids in communication, and facilitates negotiation among the human agents thus, in theory, producing a better, more globally optimal result.
To explain pair comparison further, the following example will be used. Suppose a company wishes to run an experiment to determine if a new synthetic material is better than the existing one used for making soles of boys’ shoes. An experiment is to be run to see if the new, cheaper sole wears at the same rate at which the old soles wear out. The experimenters have only enough resources to make a small number of shoes. The question arises; how should one run the experiment? Consider two possible experiments:

**Experiment 1:**
- 10 boys are selected at random
- Each boy is given a pair of shoes
- 5 boys receive a pair of shoes with the old sole (Sole A) and 5 boys receive shoes with the new sole (sole B)
- Each boy wears the shoes for 1 month and the amount of wear is measured

**Experiment 2:**
- 10 boys are selected at random
- Each boy is given a pair of shoes
- Each pair has 1 shoe with the old sole and 1 shoe with the new sole
- For each pair of shoes, the sole type is randomly assigned to the right or left foot
- Each boy wears the shoes for 1 month and the amount of wear is measured

Which experiment is better? If experiment 1 is analyzed one can see that the results of this type of experiment are affected by the introduction of variability. If one boy receives sole A and another boy sole B, then the experimental error (variability among experimental units that receive the same treatment) reflects variability between boys and the variability within each boy. Experiment 1’s design introduces boy-to-boy variability. If the boy with sole A is physically active and boy with sole B is non-active, how do you know that your conclusions about the sole wear is actually due to the new material or due to the fact that one boy is more active than another.

If experiment 2 is analyzed one can see that the variability has been minimized. If each boy receives both soles, then the comparison within each boy eliminates the variability among boys from the reference noise. In this case the experimenters would measure the wear between a boys’ left and right shoe. This eliminates the boy-to-boy variability seen in experiment 1. It no longer matters if one boy is active and one boy is not since wear is measured only within each boy. The randomization of which foot gets the new sole also eliminates variability. This is an example of blocking. In summary:

- **Blocking** — eliminates sources of variability
- **Randomization** — balances possible effects of uncontrolled sources of variability providing a fair estimate of noise variability
- **General Guidance** — Block what you can and randomize what you cannot

The discussion above shows the importance of good experimental design. There are many noise or blocking factors involved in design experimentation. Such factors are the learning curve of the agents, relative skill of each agent, and overall intelligence of each agent. To determine if the fuzzy logic agent software is useful the following experiments were designed and completed:

- Four groups of five students participated in the experiments. Each group completed two experiments giving eight total experiments performed. One experiment was conducted with the fuzzy logic agent software to negotiate and communicate and one experiment was conducted with an Internet chat window as a communication tool. The students were not co-located in either case.
- Each student was randomly given an agent role to perform and performed the same role in both experiments.
- The same design tools were used for each experiment.
- Each group designed a unique containership. The same design requirements were used for the non-software and software experiments within each group. Between each group, the TEU and speed requirements changed. The draft restriction changed between each group as well.
• To minimize variability due to learning curves, the starting software choice was randomized for each group. Two of the groups started with the fuzzy logic software and then completed the same design with the chat tool, while the other two groups started with the chat window and then used the software tool.
• To further minimize learning curve variability, the two experiments for each group where spaced at least a week apart so that the students would be less able to remember specific values and answers.
• To ensure a fair comparison, each group was given 5 hours to work on the task.

This type of setup allows one to make conclusions while minimizing variability. Since the comparisons are done within groups, the group-to-group variability is eliminated.

6. Results

Experimental results must be evaluated and interpreted in terms of the subject matter, not just expressed as statistical results. The goal of the experimentation was to prove or disprove the following hypothesis:

The fuzzy logic agent software will help produce a more globally optimal design by:

1. Keeping the variable sets open longer thus facilitating set-based design. The facilitation of set-based design is assumed to produce a better design.
2. Increasing the number of alternatives an agent looks at for a given variable. If agents look at more alternatives, they are sampling a larger portion of the design space and increasing the chance at finding a global optimum.
3. Increasing the amount of efficient communication among agents.
4. Providing a more systematic approach to design.

The results of the experimentation proved the entire hypothesis to be true. The fuzzy logic based software approach enabled a systematic method for achieving set based design for an advanced product. Due to page limitations, only typical results that best show the conclusions are presented. The results and conclusions made from the data were consistent for all four groups. The trends and results were consistent throughout all of the experimental groups as well. For full results see Singer (2003).

6.1. Range for a Variable versus Time

Figs.8 and 9 show the typical range of negotiated variables over the time of the experiment for the software and non-software experiments. To prove the hypothesis correct, the graph should have a logical narrowing of the range over time. Fig.8 shows the typical comparison between the range of a negotiated variable in the software and non-software cases; here the range (set width) of the beam considered in a particular experiment. The trend for the software case is a gradually narrowing set. For the non-software case there is no trend to the range data and most of the time the team was considering only a single value. In almost all the negotiated variables, the range of a variable was zero in the non-software case.

Fig.9 shows the range (set width) of the cargo vertical center of gravity considered in a particular experiment. The trend of the non-software experiment is typical but in the software case the range opens up at the end of the experiment. This shows one of the hidden benefits to the fuzzy logic system. As the design progressed more flexibility presented itself with regards to the cargo center of gravity and the chief engineer challenged the team to reevaluate a wider set to ensure that the best solution had been formed. Since the system is dynamic, the variable ranges can be expanded to exploit flexibility that might have been infeasible earlier in the design. Figs.8 and 9 show that the
fuzzy logic software agent does facilitate set-based design as well as provide the designers a method of opening up the solution space when feasible.

6.2. Number of Alternatives per Variable for an Agent

Figs. 10 and 11 show the number of alternatives particular agents evaluated for each negotiated variable in the software and non-software cases. To obtain these values in the software case, the total number of membership function elements transmitted by an agent for a variable was counted. For the non-software case, the number of times an agent mentioned a variable and it changed was totaled for that variable. To prove the hypothesis correct, the software should have a larger number of alternatives per variable for each agent. In theory, the larger number of variables and variable combinations an agent evaluates can, in essence, be related back to the total number of designs the agent investigated. It is, however, impossible to directly evaluate and measure the total number of designs an agent investigates. This is due to two facts. First, the human brain can, without direct knowledge, evaluate billions of combinations of possible designs from a discrete number of variables. It is impossible to document all of those decisions. Second, there are an infinite number of combinations of similar variables that make up essentially the same design. One design with a waterline length of 120.2 meters and a beam of 20 meters is essentially the same as a design with a waterline length of 120 meters and a beam of 20.03 meters. In a preliminary design these differences are negligible, thus, they are not really two different design choices. Given these facts, the question arises: how should the number of designs be evaluated? If, for a given agent, the number of variable alternatives is evaluated then conclusions can be inferred regarding the number of designs an agent investigated. Figs. 10 and 11 show the number of alternatives for all the negotiated variables considered by the Hull and Cargo agents in a particular experiment. Both figures show that the fuzzy logic agent software dramatically increased the number of variable alternatives evaluated. Thus, the experiment has shown the hypothesis to be true.
6.3. Number of Total Iterations Completed per Variable

Fig. 12 shows the total number of iterations for a variable for all of the agents in a group. To obtain these values in the software case, the total number of joint preference curves produced for each variable were counted. In the non-software case, the total number of times a variable was mentioned and changed was noted. To prove the hypothesis correct, the software should have more iterations per variable than the non-software case. Fig.12 differs from Figs.10 and 11 in that the latter show the number of alternatives a particular agent evaluated, while Fig.12 shows the number of group preference assessments made for each variable. Fig.12 shows how well the system promotes negotiation and conflict resolution. This data makes no inferences about the quality or feasibility of the design or variable just the number of times a group decision was made. From Fig.12, the conclusion can be made that the fuzzy software in all but two cases had more iterations than the non-software case. Even though the software case performed better in almost all of the variables, the difference between the software and non-software case is not large enough to make any conclusions. When the software case is superior, the difference is large. Thus the software case is slightly better in most cases, but it has the potential to make substantial improvement thus making the characteristic described statistically significant. Fig.12 thus also proves the hypothesis to be correct.
6.4. Total Number of Submissions per Agent

In the software case, the total number of preference submissions per agent for all of their negotiated variables was counted. This assesses the level of engagement of each agent in the design process. In the non-software case, the total number of times an agent mentioned a variable and changed its value was counted. To prove the hypothesis correct, the software case should have a larger number of submissions per agent than the non-software case. Typical results are shown in Fig.13.

Fig.13 also helps to determine if some of the other conclusions made from the previous data are correct. For the software case the total number of submissions made for all five agents far exceeded the non-software case. For the non-software case, the number of submissions for the Cargo,
Propulsion, and the Resistance agents are all consistent in contrast to the activity of the Hull and Stability agents. Because the software environment forces the designers to communicate on the cycle established by the chief engineer there is never an instance where an agent is allowed to remain silent. With the Stability agent, it can be noted that they were not involved in the design process during the non-software case while, in the software case, they were very active. The Stability agent’s lack of involvement could have affected the overall quality of the final design in the non-software case.

The number of submissions data for the Hull agent, Fig.13, has a different interpretation than for the Stability agent. In the non-software case, the hull agent’s number of submissions was a factor of 4 times greater than the average number of submissions of the other agents. This can be explained by the fact that the Hull agent in this experiment is a very talkative, energetic person. Thus, the Hull agent took full advantage of the open forum environment of the chat window. The reason that the number of submissions in the software case is larger than the average is directly related to the number of required communication negotiation paths in the fuzzy logic software environment. A detailed review of the experimental data shows that, in the non-software case, the data in Fig.12 was actually skewed by the Hull agent’s increased level of communication. This explains why in the Beam and $C_{wp}$ data in Fig.12 have more iterations in the non-software case.

Fig.13 also supports the hypothesis. The fuzzy logic agent software increases the amount of communication and negotiation that can take place with in a team-based preliminary ship design environment.

**Conclusions and Directions for Future Work**

The conclusions made from the experimental series are that the fuzzy logic agent software does facilitate set-based design, thus, increasing the probability of reaching a globally optimal ship design. The set-based design paradigm can replace point based design construction with design discovery; it allows more of design to proceed concurrently and defers detailed specifications until tradeoffs are more fully understood.

One of the underlining advantages of the fuzzy logic agent software is its ability to keep the variable sets open longer, which will, in theory, facilitate and enable set-based design. In open form communication there are no controls to assure that all team players are actively participating in the set-based design philosophy. The fuzzy agent software eliminates this problem. Since the fuzzy agent software is constantly evaluating the joint preference curves of a variable over the variable’s current range, the software possesses the ability to dynamically adapt to the changing design. The software environment also demands and balances the active participation of all agents.

The fuzzy agent software communication structure enables an agent to increase the number of alternatives that it evaluates for a given variable. If agents look at more alternatives, they are sampling a larger portion of the design space and thus increasing the chance at finding a globally optimal design solution.

The next evolution for the fuzzy logic agent software development is to evaluate the usefulness of the tool in a commercial real world setting. The system has been tested so far only in an academic environment using inexperienced student designers. We believe that this new method for computing and presenting multi-variable joint preference curves can help cross-functional design teams further understand the design space, achieve a more globally optimal design, and more quickly respond to bid proposals.
References


EVANS, J.H. (1959), Basic Design Concepts, Naval Engineers J.


SINGER, D.J. (2003), A Hybrid Agent Approach for Set-Based Conceptual Ship Design through the Use of a Fuzzy Logic Agent to Facilitate Communications and Negotiation, Ph.D. Dissertation, Univ. of Michigan, Dept. NAME

Virtual Prototyping for Developing South African Ports

Desmond Simpson, National Port Authority, Johannesburg/South Africa, dess@npa.co.za
Ludwig Furstenberg, ITE, Durban/South Africa, ludwig@ite.co.za
Ektor Kalathas, ITE, Durban/South Africa, ektork@ite.co.za
Volker Bertram, ENSIETA, Brest/France, volker.bertram@ensiesta.fr

Abstract

Although Expert System technology is no newcomer to the Artificial Intelligence scene, few applications of virtual prototyping has been reported for the maritime industry. The ability of the expert system to apply human reasoning that can evaluate the state of a system to recognise and act on triggers provides a powerful environment for virtual prototyping of logistic systems. Heuristic based discrete event simulation models have been successfully developed and applied to assist the National Ports Authority of South Africa (NPA) with the development of port infrastructure to cater for future industrial expansions and cargo growth, de-bottlenecking process flows and numerous capacity studies during the past 10 years. Potential problems of proposed expansions can be identified and rectified within a low cost environment. This article will discuss the enabling technologies available through an expert system shell and a typical project life cycle will be discussed to illustrate the practical benefit of this technology.

1. Introduction

Recent advances in Artificial Intelligence technology have seen the emergence of Expert Systems into the industrial marketplace, where they are rapidly gaining ground in the planning, scheduling and modeling of complex business systems and environments. It is estimated that some 25,000 applications of Expert Systems are currently in use worldwide, though few have been reported in the shipping industry, Bertram (1997). Applications to shipbuilding and shipping are reviewed by Bertram (1998).

In broad terms Expert Systems are computer software packages that are able to apply expert reasoning to a particular problem or environment. Through a method of structured reasoning, the Expert System diagnoses a situation to reach a conclusion. The Expert System is programmed with the rules and procedures relating to how decisions should be made: this includes a variety of soft reasoning – “rules of thumb” and a great deal of process information, which is often difficult to express in conventional mathematical or statistical terms. A Knowledge Base is the name given to a set of Expert System reasoning: it is not just data or information, but encapsulates the knowledge of how to reason with the information.

Integrated Technologies Engineering (ITE) and South Africa’s National Port Authority (NPA), have led the way in the International Port fraternity by implementing Gensym’s G2, a state of the art expert system shell, in the development, management and operation of South African ports. All seven of South Africa’s commercial ports are under NPA’s control. In 1992 the first expert system application was build to assist in distributing some 12 million tons of maize donated by foreign aid to 4 drought stricken Southern African states. Consequently ITE has been involved in developing simulation models for each of the South African ports and has conducted numerous capacity studies, planning expansions, assisting NPA to find non-capital solutions for capacity problems (optimising resources). ITE has also built and used expert system based simulation models in various process industries to perform scenario testing, risk analysis and design verification of various logistic systems.

It is the purpose of this article to document the application of simulation in the maritime environment. It also discusses a simulation model based investigation for the Port of Richards Bay. The model was commissioned by NPA, with the specific purpose of predicting storage and berth requirements, given
the expected increase in commodity throughputs. The first couple of paragraphs would be devoted to describing the methodology of building a simulation model, with some practical guidelines given to applying a simulation model to a real life problem. The second part of the paper will discuss the use of simulation by NPA and the article will close by sharing some comments of a specific application of simulation in the port of Richards Bay.

2. Simulation – Art or Science

A simulation study is initiated by a problem in an existing system or while designing a new system. Simulation is but one of several tools available to the engineer for problem solving. There could be other alternatives that provide quicker/cheaper solutions, but might not be suitable for the particular problem. Own experience has shown that there is often great inertia for introducing new technology (simulation, expert system) to management level or that there exists a great deal of hype about what the computer model will be able to do – accepting results produced by some fancy (misunderstood) computer system blindly.

2.1 Defining the problem

It frequently happens that a problem is analysed without really understanding what the real problem (or root cause) is and the side effects or symptoms of the problem are investigated. It is of paramount importance to clarify the purpose of the study, that could be one (or any combination) of the following:

- Evaluation Comparing a system when evaluating against specific criteria
- Comparison Competing systems/proposed alternatives
- Prediction Estimate system performance under different operating conditions
- Sensitivity analysis Determine which and to what extent factors influence system performance
- System improvement/Optimisation

The analyst needs to view a part of the real world (the problem) as a system in order to define boundaries for the study as well as the level of detail. A system is a collection of interrelated objects that function towards one or more stated goals. Two boundaries are of concern:

- The boundary that separates the system from the rest of the universe it belongs to.
- The boundary that separates the system defined from the environment in which it operates.

The purpose of the boundary definition is therefore to simplify the simulation study by reducing the amount of detail included. “Keep it simple, as simple as possible, but no simpler” - Einstein. It is often a good idea to first develop a high level model, with little detail and then add in the required complexities in subsequent iterations. Validation and verification are generally easier and faster for each iteration. However, there are always modules and parts of other models that can be adapted and used in the current model, which then eliminates to a large extent this clean sheet approach.

Boundary definition and model abstraction also include making assumptions. These should be thoroughly and unambiguously documented and referred to during the study. The list of assumptions is dynamic and should be updated during model development. The output results should also be evaluated against the assumptions made, thus they should form part of all reporting.

In the field of knowledge engineering, the knowledge about the process sits with the client and the skill to develop a simulation model sits with the knowledge engineer/analyst. These two resources have to combine forces to effectively attack the problem at hand. In the majority of cases the person(s) with the operational and process knowledge have little or no computer skills/interest, thus some
common ground needs to be found for exchanging knowledge. A conceptual model is very useful at this stage to:

- Teach the analyst/knowledge engineer how the system functions.
- Identify lack of information on system operations, which can be supplied during a second round of interviews with operational personnel.
- Establish the model’s logic.
- Identify input date required.
- Verify assumptions.
- Clarify the various levels of detail that would be required.

Once this first part of the computer model has been completed, can the actual building of the model commence, as discussed in the next section. ITE makes exclusively use of Gensym’s G2 expert system shell for developing simulation models. Some of the discussion will thus specifically apply to G2 and the port environment.

2.2 Model Building and Knowledge Representation

The first step in developing a realistic dynamic simulation model is in capturing information/knowledge that effects the operation of the system to be modeled. The capturing of information takes place on various levels:

![Simulation Model Diagram]

Fig.1: Example of a simulation model

The first level consists of defining the infrastructure, such as berths, conveyors, railway lines, storage facilities, cargo-handling equipment (tipplers, ship loaders, trains, etc). Object-oriented design is the basis of development in G2. Built around the concept of objects, object-oriented design offers highly
reusable code, and an application structure that is much more intuitive to understand than those built with conventional programming tools. The applications are highly graphical, with the schematic of the layout built from scaled drawings. The actual location of the objects upon the schematic has physical significance that the model can reason with (i.e. connectivity in automated route finding, selecting resources closest to current location, etc.). These objects also have attributes that are configured to define their operations and interactions, such as draft restrictions, total length, bollards used (berths), various handling rates, train travelling times, locomotive requirements, etc. These objects can be something physical like a tank, pump, or instrument – or something abstract like an event, task, message, request or logical connection.

The second level of information consists of the details of the various operations that the system needs to perform, i.e. production rates, orders to be filled, information about the delivery schedules for materials, storage facilities available, to name but a few. This information is typically stored in the user defined attributes of the various objects, or in global parameters, lists and distributions (system defined objects).

The third level of information is where the real expert system capabilities of the model are used. The expertise of the operators of the enterprise is captured in rules, procedures and methods that control the interaction (behavior) of the objects and addresses issues such as:

- Where, when, what, by whom and how certain orders/tasks are performed.
- General rules and methods in the plant operations, i.e. effects of weather, dust contamination, reserved infra structure, train schedules, shifts, etc.
- Detailed information on process times, maintenance schedules and breakdown information.
- Special shunting rules and duties in the rail yards.
- Hand over procedures between mainline locomotives and shunting locomotives.
- Priorities in servicing departing/arriving vessels.
- Preferential berthing rules.
- Tidal and daylight restrictions on operations.

G2 provides procedures for carrying out complex sequential actions and calculations. Unlike a rule, a procedure does not simply tell G2 how to respond to a given set of conditions, but provides G2 with a set of instructions to follow. Procedures make it easy to iterate over all objects in a class, spawn parallel processes for concurrent execution, and execute many other actions.

G2's procedural language has all of the control structures and many of the characteristics of other common programming languages. For example, it contains repeat statements, case statements, and so on. It also has a number of features that where developed specifically for the needs of a real-time application and are very useful for developing simulation type applications. For example, it provides “do in parallel” statements for executing several sets of statements at the same time. Furthermore, procedures can make use of powerful wait states (either time driven or event driven) to coordinate tasks happening in parallel. For example, a process could start (servicing of a ship), but as soon as a breakdown occurs (controlled by separate code), the process can be interrupted and the control handed over to a different set of procedures. Once the breakdown is completed, the control can be handed back to the servicing procedures. This ability enables the developer to capture complex knowledge of how to handle various types of breakdowns and interruptions in an enterprise:

- What equipment should be freed up.
- What contingency plans should take effect – accurately reproduce the ripple effect caused by disruptions.

Methods in G2 capture the behavior of objects. Methods can be thought of as procedures that are linked with object classes. Associating methods with object classes captures the object’s behavior. This means that methods can be hierarchical. G2 also allows multiple inheritance, where a class can inherit capabilities from many other classes. For example, a communications computer might inherit properties and (methods) from both a communication-device and computer class or a tank-truck might
inherit from both a tank and truck class. Objects do thus not only inherit properties (attributes) from their parent classes, but also behavior, and usually contain additional attributes and behaviors unique to themselves. This reusability can be deployed to dramatically streamline class hierarchies and reduce the number of rules and procedures required.

The heart of G2's real-time operation is a scheduling engine for concurrently executing rules, procedures, models, and other tasks. Typically, many lines of reasoning, models, and other activities are occurring concurrently. The engine cycles according to a clock “tick” that can be specified at the millisecond level. Every task has a priority, so critical items can take action first. G2 takes care of the scheduling of tasks for the user, based on simple configuration choices.

There is practically no limit to the detail that can be included in a simulation model with the basic model of the client’s operations often extended to include the infrastructure that transport material/products to and from the hinterland, i.e. rail and road networks. In fact, since the dynamics of the system as a whole is influenced by the interactions between supply and delivery operations, it could be essential that the simulation model also include the rail and road operations.

2.3 Data Sources

The relevant information for an enterprise is extracted from plans, drawings, asset registers, performance statistics and numerous interviews with experienced personnel. It is in the “interpretation” of all this information that an experienced expert system developer is required to transform the infrastructure-related information into an object hierarchy representing the layout of the port and convert operational information into rules, procedures and methods that controls how the objects interact.

A large amount of process information is required and can be sourced from one or more of the following, du Preez and Bekker (1996):

- Historical records
  - Data may or may not be up to date.
  - Usually large amount of information is stored.
  - Current data format might require special computer programs for extraction and translation.
  - There may be platform restrictions (VMS, UNIX, Windows).
- Observational data
  - Observe a system in operation and gather data personally.
  - Time consuming – limited sampling space.
  - Observations may reflect a snapshot of a seasonal trend.
- Operator estimates
  - Can be utilised if insufficient time is available to do a complete study.
  - People are often poor estimators of parameters/events when they are highly familiar with them – extreme cases are usually forgotten and most recent ones over emphasised.
- Vendor specification
  - Provide estimates of the operating characteristics of new equipment.
  - Estimates are invariably highly optimistic.
- Designer estimates
  - May be the only source of data (new systems).
  - Estimations can also be far off, as the designer expects that the system will operate as intended.
- Theoretical considerations
  - Mean time between failures for electronic equipment is Weibull distributed.
  - Time between arrivals is usually exponentially distributed.
Once this process information is accumulated, it is usually stored as distributions that get sampled whenever a process time (for example) is required during the simulation run.

2.4 Benchmarking

Benchmarking is the process of validating the model and should not be confused with verification. Verification confirms that the model was built correctly (it does what we expect), it does not guarantee that the correct model was built. In other words, the fact that the model produces results does not necessarily provide confidence as to the relevance of these results to the real world problem. For validation, three questions must be considered:

- Conceptual validity Does the model represent the real world system adequately?
- Operations validity Does the model produce realistic results?
- Credibility Does the end user have confidence in the model/results?

The development of a model is an interactive process where the operators of the system have to play an important role in sharing their expertise and validating the model. The highly graphical nature of G2 enables easy visualisation of system operations that greatly assists in qualitatively verifying the model. Once the client is satisfied that the basic workings of the model is an accurate representation of the reality, a quantitative benchmarking process is commenced where a stable period of system operations is selected for simulation and various performance parameters are compared with historically recorded values.

All these steps are necessary to establish the integrity of the model before it can be applied for a simulation study.

The model is usually run for a one-year duration for Port type applications, which takes about 5 - 60 minutes on a PC, depending on the level of detail simulated and the complexity of the operations. The output of the simulation includes:

- A complete breakdown of operations giving gross production rates as well as simulated delays for breakdowns (electrical, mechanical, etc.) and weather delays.
- Gross and net utilisation for each machine.
- Various queuing times and stock level trending.
- Process turnaround/completion times.
- Throughputs.
- Some economic factors such as ship demurrage.
- Berth occupancy levels

At this stage it might be necessary to refine some parts of the simulation model (simulate processes, rather than sampling a process time from a distribution) and the benchmarking process has to be repeated.

2.5 Optimisation

In broad terms, optimisation involves adjusting a set of input parameters that influence the performance of a system in order to maximise or minimise a certain objective function, e.g. total costs, profit, throughput. This objective function could consider any number of output parameters and could include a number of “rules of thumb”, not easily expressed in mathematical terms. After each simulation run the optimisation algorithm should intelligently adjust the input parameters based on the results of that run (and previous runs) and re-do the run with this new set of input parameters until a satisfactory or mature solution is found. A satisfactory solution is one that meets specified constraints. A solution is considered mature when there is not a significant change in input parameters between consecutive iterations, i.e. the algorithm eventually ends up re-doing the same run. However, a mature solution does not guarantee optimality.
2.6 Scheduling

Scheduling is a specific branch of optimisation, where the variable is the sequence in which certain (interdependent) operations are performed and/or the resources used to perform these operations. In real life problems, there are often complicated constraints that have to be met, which often renders most traditional operational research and mathematical techniques of little or no value. What makes real life problems even more difficult to handle is the fact that there is seldom a clear way of distinguishing between good and better schedules. For example, one can only plan for known events, but random breakdowns and interruptions have somehow to be accounted for. The volatile environment of scheduling might find a region of high average “goodness” more useful than an isolated (vulnerable) optimal solution. In other words, in a real life environment a schedule that is optimised according to a certain objective function might turn out to be very hard to follow in the event of unplanned disruptions. A better option might be to consider a sub-optimal solution that has a number of contingency plans in place that could be followed in case of disruptions. The ability to react intelligently to a deviation from a planned schedule is of great importance to reduce the ripple effects introduced by unforeseen disruptions.

There are thus a number of criteria that a “good scheduling environment” could be measured against:

- The ability to produce good, not necessarily optimal solutions, in reasonable time.
- The ability to handle complicated constraints.
- The ability to perform risk analysis to evaluate the vulnerability of a schedule.
- The ability to monitor the actual operations, comparing it with the planned schedule and automatically trigger re-scheduling / contingency plan activation, should disruptions occur.

ITE has developed several optimisation tools, one of which is discussed in Furstenberg (2000).

3. Simulation by NPA

NPA has commissioned the development of various simulation models over the past decade. The purpose of these simulation models has been to provide NPA with a facility to analyse the capacities of different types of terminals. A simulation model of a port is a cost-effective way to accurately determine the current capacity and constraints of that port. The same model can then be used to predict the future capacity, constraints and maximum throughput, given the cargo projections of the port. These predictions enable NPA to be pro-active in their management and planning of the port, to identify bottlenecks, correct them before they happen and to time capital intensive projects correctly to meet market demands.

3.1 NPA Profile

All the commercial ports within South Africa are owned by Transnet, and are constructed, controlled and operated by NPA under the direction of the Minister of Public Enterprises. The ports are located in the cities of Richards Bay, Durban, East London, Port Elizabeth, Mossel Bay, Cape Town, Saldanha Bay and last in Coega where the port is still under construction.

NPA’s policy has been to provide a complete range of port services and facilities. These comprise pilots, pilot boats, berthing, towage tugs, berthing staff, dredged channels and basins, quays, cranes, rail connections and roadways, dry-dock and slip-ways and shore labour for shipping and discharge. NPA is also responsible for lighthouses, radio beacons and other navigational aids.

3.2 NPA Planning & Development team

Planning and Development resides at NPA’s Head Office in Johannesburg. This is a multi-disciplinary team, resorting under the General Manager of Landlord Services, which investigates projects and accordingly reports to the NPA Planning Board. The functions of the team are to:
• Support and advise the Capital Expenditure Committee.
• Manage and facilitate the capital planning process.
• Participate in the strategic planning process.
• Advise on strategic issues.
• Ensure trade and cargo handling facilitation.
• Short, medium and long-term port planning.
• Terminal planning.
• Ensure effective use of port infrastructure.
• Plan ahead to provide adequate facilities for port activities.

3.3 The role of simulation in the planning and development of ports

Although proper Information Systems based on Information Engineering principles are absolutely mandatory for the efficient management of ports, the focus is only on data modeling of present operations. It does not provide the capability to improve or redesign operations, nor does it provide system insight into why problems occur and “why” operations are performing the way they do”. It mainly provides management with information about “how” operations are performing”, mechanistic forecasting and statistical analysis.

3.3.1 Port simulation

Computer simulation models are proving themselves to be a major new technology in port planning and design in response to the many problems and combinations of parameters in the port environment. Port simulation enables a detailed time based modeling of all relevant operations and does not assume a purely mathematical or statistical representation of processes. In such a simulation model the actual process flows and interactions are modeled as discrete events.

Models of this nature enable operators to re-design the business process and helps to develop and optimise the heuristics of the various operational strategies. Such models can enable managers to exert control over the whole operational system, not only enabling them to evaluate the effects of system changes, but to also optimise system performance by altering operational rules after such changes are introduced.

The models can be tested with different tonnages, equipment and capacity, full range of operating restrictions, variable number of berths, changes in support facilities and operations, alternative inter-modal transport system, and other aspects which bear on its effectiveness. Different scenarios are compared by making changes in the capital investment (% improvement measured related to number of berths, number and capacity of equipment, infrastructure support etc) or operations to see how the costs associated with vessels and terminal are impacted, Simpson.

3.3.2 Benefits of port simulation

Simulation analysis in the above mentioned format enables management:
• To understand where and why bottlenecks occur.
• To perform extended “what if - suppose that” analysis on all system variables.
• To change the way operations are presently performed in a simulation model and therefore evaluate system effects and costs.
• Eradicate uneconomical operations and port activities.
• Evaluate the impact of future investments on total system costs and performance.
• Determine the effect on port and rail costs as well as on capacity utilization and traffic volumes if specific commodities are routed to different ports.
• Establish the effect on port facilities and systems utilization by new patterns of origins and destinations for specific commodities.
• Determine realistic maximum levels of port utilisation before congestion sets in.
• Determine the lowest cost logistical process flows.

3.3.3 NPA and G2

Based on the above mentioned features NPA decided to purchase G2. A strategy was embarked upon to model all the main business areas of NPA. The idea being that once these areas had been modeled, they would be kept “live” on a 6 monthly basis by updating and re-benchmarking the models. Furthermore these models would be connected on a national level (Big Picture Simulation Model) to create a single knowledge base, and in so doing, allow the testing of any scenario and future development to be run on any of the models. This type of approach would enable the Planning Team to quickly assess the needs and proposals suggested by clients and port terminals and in so doing draw conclusions which otherwise would have taken many more man-hours.

4. Specific application of simulation in the port of Richards Bay

An example of applying simulation is given in the following Case Study of the Dry Bulk and the Multipurpose Terminal at the port of Richards Bay. The purpose of the study was to enable NPA to evaluate the current and future capacities of these terminals and ensure that proper infrastructure is in place on time to accommodate the expected cargo growth.

4.1 Port description

The Port of Richards Bay is a commercial cargo handling port located approximately 160km North-East of Durban and 465 km south of Maputo on the eastern seaboard of South Africa. The Port’s main hinterland comprises the Northern KwaZulu Natal, Gauteng and Mpumalanga regions. It is a relatively young port, having opened on 1 April 1976. The port is presently South Africa’s leading port in terms of cargo volumes and handles in excess of 90 million tons per annum, representing approximately 55 % of South Africa’s sea-borne cargo trade.

The Port of Richards Bay has the capacity to develop into one of the larger ports in the world. The present total land and water surface areas of the port are 2157 hectares and 1495 hectares respectively. To date only 40 % of available land area is developed, however, this does not include the future development potential if one includes the areas lying outside port limits that are zoned for port development.

The port of Richards Bay consists of a private terminal (Richards Bay Coal Terminal) and NPA controlled terminals (The Dry Bulk Terminal (DBT) and the Combi Terminal (CT)). Ships which carry break bulk commodities such as steel, ferros, containers, logs, copper etc. berth at the 600 and the 700 Berth Series. Ships which carry dry bulk commodities, like zircon, rutile, sulphate, fertilizer etc. berth at dedicated berths of the 700 Series, Fig.2. These berths are equipped with special conveyor belts, which transfer the cargo to the dry bulk storage areas.

4.2 Scope of the Study

The Port of Richards Bay is perceived to be operating close to capacity regarding break-bulk commodities. There will not be additional infrastructure in place in the near future. However, NPA is expecting cargo growth in the following years. The purpose of this study is to evaluate a short and a long-term solution for this problem.

Due to historical development resulted in some of the break-bulk commodity being worked at the 600 series and some at the 700 series of berths, Fig.3. However, the cargo mix of vessels visiting Richards Bay typically requires commodities from both areas. Cross haulage occurs unavoidably when the ship does not berth next to the storage area where the cargo is stored. So the cargo is transported with haulers to or from the appropriate storage areas to the ship. That causes reduction in handling across the ship rates and subsequent increase of the berth occupancy.
Fig. 2: Port of Richards Bay

Fig. 3: NPA Terminals
Changing the allocation of commodities per berths to match the typical cargo mix of vessels can reduce cross haulage and promises benefit on the short-term. This involves moving some of the break-bulk storage areas from the 600 series to the 700 series and vice versa.

The long-term solution requires port infrastructure changes, specifically the construction of new berths and possible expansion of the rail yard. NPA must determine following parameters before commencing on the port upgrade program:

- The number of new berths needed
- The optimal timing of when the new berths should be in operation.
- The optimal layout of the additional berths (extending the 600 or 700 Berth Series)

4.2 G2 model for the port of Richards Bay

To carry out these short term and long term investigations, ITE has developed a G2 simulation model.

4.2.1 The Simulation Model

The first simulation model was built for the DBT operations and later extended to include MPT operations.

The simulated operations include:

- Berths and berthing rules
- Conveyor networks that include the dynamic management of conveyor routes
- Storage areas
- A detailed rail yard model that simulates each individual shunt, including breaking and building of trains
- The use of equipment pools and work gangs
- Specific hauler detail to accurately capture the dynamics of cross haulage
- An updated forecast for all the commodities serviced at Richards Bay

4.2.3 Simulation application

The simulation provides a scientific means for quantifying the impact of the various layouts and proposed operational changes to identify the best candidates. For instance, it is possible in a relatively short time to make following accurate predictions:

- The influence of throughputs and handling rates on berth occupancies
- Berth occupancies and additional capacity of new berths.
- The effect of port expansions on vessel queuing.
- The utilisation of storage areas and whether the storage area is sufficient for a particular commodity
- The utilisation of movable equipment such as forklifts, haulers and trailers and the effects of adding additional equipment into the system.

Various virtual ports were tested with different port layouts to maximise the potential benefit of matching the port to expected cargo mixes. This ensures the best use of available infrastructure – short-term requirement is to optimise current operations without building new berths.

Long-term expansions necessitate the building of additional berths. Where to build these berths is influenced by the following:

- There are different cost implications to extending the 600 series compared to the 700 series due to deviant stability of the seabed.
- The long-term expansion possibilities are very different at the 600 and 700 series due to the proximity of undeveloped areas.
• The current structure has to be changed in a phased approach to the future layout and the various options at the 600 and 700 series would require different expansion trances.

The focus on this study was on the port expansions, but the cargo to and from the hinterland has to be serviced mainly via the rail yard. A final verification was thus performed where the proposed port expansion options were simulated with the future rail yard extension to identify possible bottlenecks in the servicing of trains.

5. The Next Chapter

ITE has developed a simulation model for each of the 7 commercial ports in South Africa and has made these models accessible on a national level. Any model can thus be configured from anywhere in South Africa to perform a series of simulation runs. This tool enables port engineers and planning personnel to evaluate the effect of port changes in any of the 7 ports in a matter of hours.

The simulation techniques have proven themselves in the port environment in South Africa and are rapidly gaining ground in various hinterland industries that interact with the port at various levels. These include investigations focusing on:

• The import of raw materials through the port and exports finished product (impact of other port cargo on their specific vessels).
• The effect of shared conveyor belts routes between the port and the plant.
• The effect of rail delays if a common rail yard is used as opposed to a dedicated siding to name but a few.

Numerous applications have also been developed for industries not related to the shipping industry at all. In fact, discrete event logistic simulation models have endless application to problems ranging from construction, transportation even legal and financial environments. Like with any technology, the success of a particular application depends strongly on the skillful commissioning of these techniques to provide insight into real life problems.

References

BERTRAM, V. (1997), South African Ports – Artificial Intelligence, HANSA 134/12


FURSTENBERG, L. (2000), Combining Artificial Intelligence with VTS for a Traffic Management Information System, 1st COMPIT, Potsdam

SIMPSON, D., G2 revolution in planning and operations in port environment, National Port Authority of South Africa.
3D-Modelling in Shipbuilding and the Missing Bricks

Bernd Klehn, archnav.de, Raisdorf/Germany, b.klehn@archnav.de

Abstract

The Shipbuilding- and Ship Operation Process could be streamlined by 3D models. However, the IT development on most shipyards has gone in a completely different direction:

- The number of different software packages has increased
- The number of data files has increased
- The number of inaccurate plans has increased.

The reason for this development are the missing bricks between the different software-tools. There are important breaks between the ship-theory tools, the classification tools, and the detail design tools. Ship-theory, classification, and detail design have to take the same ship and the specific conditions at the shipyard into account, but there are no common data files available to guarantee this approach.

1. Introduction

Europe builds more and more special ships such as Ro-Ro ferries, passenger vessels, yachts, research ships, and naval ships. These ships can be very different in their concept and their design. Thus, it becomes more risky to base the early design phase only on old standard formulas and comparison ships. From the early beginning, there should be a 3D model in the computer to implement Virtual Prototyping (VP). The most important difference between VP and VR (Virtual Reality) is that VP emphasizes the physical behaviour of the ship, VR the visualisation.

In shipbuilding, VP has a very long tradition, because you have to know the behaviour of a ship before going to sea. Ship theory could be understood as the forefather of Virtual Prototyping. The naval architecture ship theory software packages are the central part if someone wants to develop a concept for 3D modelling in shipbuilding.

There are more or less two independent stories for software development in shipbuilding. The ship theory story has already be mentioned. The other is the mechanical engineering story. Shipbuilding detail software had roots in CNC cutting which is essentially a 2D process. Ship theory software had to handle complicated free-form surfaces (the hull). Therefore, ship theory software had to be a 3D software.

These basic findings enable us to investigate how the shipbuilding process could be streamlined. The first task was to examine, if the ship theory software (in our case NAPA) was open for further development. We found it (for the long run) necessary to be able to build 3D steel elements related to the ship theory surfaces (hull, bulkheads, decks, tank walls). If the hull, the bulkheads, the decks, and the tank walls are being modified, the steel elements should also alter and it should be possible to build a 3D model with most of the steel elements very quickly in the initial design phase. Our independent research showed that there were no restrictions in the software to reach that aim. Therefore, we started to talk to NAPA about our thoughts. We found that NAPA was also thinking into the same direction. In consequence, the software extension NAPA Steel was developed.

2. STEP Interfaces

Our second task was to look after the interfaces. It is not worthwhile to have independent steel models, one for initial design, one for scantling, one for FE-analyses, and one for detail design. The different parties (software vendors, classification societies) told us that there would be an international interface for shipbuilding steel structure promoted by EMSA (European Marine Step Association) called STEP-AP218. Over all, there should be the following application protocols in shipbuilding (see EMSA Handbook):
AP 215: Ship Arrangements
Specifies the information requirements for the exchange of product data representing a ship's internal subdivision information between different organizations with a need for that data. Such organizations include ship owners, design agents, and fabricators. This AP has been developed to support the shipbuilding activities and computer applications associated with the Functional Design, Detail Design, and Production Engineering life cycle phases for commercial or military ships. The types of design activities and computer applications supported include naval architectural analyses (e.g., Damaged Stability, Compartmentation and Access, and Floating positions), structural analysis, interference analysis, and weight analysis.

AP 216: Ship Moulded Forms
Specifies the information requirements for the exchange of ship moulded forms and related hydrostatic properties. The AP supports hull moulded forms and moulded forms for structures internal to the ship, and supports surface and underwater ships for commercial and military use.

AP 217: Ship Piping
(Significant overlap with AP227 - Possible "Merge"/Replacement)
Specifies the information requirements for the exchange of ship piping functional design, detail design, production engineering, fabrication, assembly and testing.

AP 218: Ship Structures
Specifies the information requirements for exchange of ship structural systems data for ship predesign, design, production, and inspection/survey. Product definition data pertaining to the ship's structure includes: hull structure, superstructure and all other internal structures of commercial and naval ships.

AP 226: Ship Mechanical Systems
Specifies the information requirements for exchange of ship mechanical systems information. This includes the exchange of information related to deck, propulsion and other mechanical systems in a ship product model. Connectivity between components and systems geometry, materials, topology and tolerances, noise, vibration and shock characteristics, component/system life cycle and operational history are part of the product model. Contains quality assurance information on availability, reliability, and maintainability.

AP234: Ship Operational Logs, Records, and Messages - New Work Item (NWI) (1999)"

3. Status reached by end of year 2000

In 2000, it seemed as if AP 218 could give us the opportunity to integrate initial design with the classification tools from LR (ShipsRight), GL (Poseidon), and DNV (Nauticus), but the progress suddenly stopped. Figs.1 to 9 (from our website) show the status we reached in Sept. 2000.

The status was promising, but then the activities stopped suddenly. The interface never came into practical use. The reasons:
- GL lacked the support of shipyards (we leaving HDW). Poseidon had problems with 3D. The software was section orientated (2½ D). The scantling dealt with standard ships such as bulkers, tankers and containerships, but these ship types were no longer worthwhile for European shipyards.
- DNV wanted to develop a complete own software starting with steel from the early beginning.
  Only the hull surface should be imported. DNV had trouble with 3D freeform surfaces and only DXF intersections could be imported.
- LR stopped its whole software development.
- EMSA is now more or less an inactive organisation.
Fig. 1: Modelling Transformation to Poseidon, Nauticus or ShipsRight

Fig. 2: Napa Steel data to Nauticus (Important is the transfer of the functional elements)

Fig. 3: NAPA-STEEL DATA to LR ShipRight via AP218

Fig. 4: NAPA-STEEL DATA to GL POSEIDON FE via AP216

Fig. 5: also NAPA-STEEL DATA to GL POSEIDON FE via AP216

Fig. 6: NAPA-STEEL DATA to DNV NAUTICUS FE via AP218
4. The new approach

The latest development is that there will be a direct interface between NAPA-Steel and SafeHull from ABS. We will look at the further progress. From the naval architecture side, there must be an integration between ship theory packages, classification tools, and detail design tools.

From our point of view, AP218 should be the steel definition interface in shipbuilding. All initial design tools, all classification tool, and all detail design tools dealing with steel should implement an AP218 import/export interface. Otherwise, there will be always missing bricks between the different 3D-Models. The AP218 is on a level which can go to practice as demonstrated by our examples.
All other shipbuilding application protocols and their related interfaces are not necessary. You can
cover the rest (outfitting, piping, electric, etc.), with the well-established CAD-interfaces:

AP203: Configuration Controlled 3D Designs of Mechanical Parts and Assemblies

AP214: Core Data for Automotive Mechanical Design Processes

5. Level reached at archnav.de

With the initial design tool, first manufacturing-relevant data are already available after 4-6 weeks.
95% of the final steel plates and profiles could be defined during this time. So you can control weight,
centre of gravity, profile/welding meters, coating surfaces of the total ship and of the blocks, apart
from the hydrostatics and damage stability. As far as we know, no other ship-structural 3D CAD
system is able to generate in such a short time so much draft-relevant information.

The model reacts by topological structure very flexibly. If e.g. the hull form has to be changed later,
all steel surfaces and profiles are adapted from the new hull form. It behaves similarly if bulkheads
and decks are changed.

The steel structure from the model can be exported via the VDAFS or IGES interface into other CAD
systems. Here, however a volume-oriented export is desirable. So that you are able to transfer also the
information about material thickness and profiles into other systems. This could be done by a STEP,
Parasolid, or ACIS interface. Our company uses an ACIS based Tool, which is used in the casting
industry and by automotive manufacturers and which is not more expensive than AutoCAD. Here all
equipment and components can be integrated into the steel structure.

It is also useful to model the main air-condition lines, exhaust system and the outer surfaces of the
super structure. Finally, a 3D general arrangement plan should be developed, which can be used in
detail design without any break. Subcontractors will provide 3D-equipment/components being used
here in the near future. Therefore, the outfitting-software must have on a long-term good and durably
import/export interfaces. The format could be STEP, Parasolid, or ACIS. If this point is not
considered, a smooth division of labour between different partners is not possible.

A further application is the visualization by animation tools like 3D studio MAX. Here the model can
be imported and prepared for acquisition purposes. The production of a rapid prototyping model is
likewise no problem based on the 3D data.

Further areas of applications exist. E.g. the steel model can be interlaced and be used for global and
local finite elements calculations.

The model supplies plans for the classification societies, which can be exported as a DXF file for
rework into each 2D CAD tool. The steel structure can be exported and imported with all
characteristics such as material, kind of profile, plate division, plate thickness etc. via the step format
AP218 (international CAD standard for shipbuilding steel structures). Step AP218 was developed
under the guidance of and in coordination with the European class societies [GL, LR, DNV] and the
US Navy. Unfortunately, the interface is not used now, since the class societies CAD tools are not
sufficient to work in 3D, or their business plan is to do all the modelling within their own tool, which
does not plan import of steel models.

The long desired interface of NAPA-STEEL to TRIBON-HULL will be launched in summer 2003.
The interface is not based on AP218. It will produce directly TRIBON schemes. It will have to be
observed whether this interface can close the gap between initial and detail design. If this were well
done, only the interface between the classification tools and the initial design software would remain.
There has been for several years an interface between E4 of FSG and Poseidon. A similar
development is now under development for an interface between NAPA and SafeHull.

Timely delivery can be guaranteed and expensive changes avoided if the 3D initial design model is used as a default for detail design. The detail design costs, the costs for rework and the turn-around time sink then substantially.

The computers became faster in the last 10 years at same price. The speed was going up by a factor of 1000. But the initial design process has hardly changed since 1992. Still, comparison ships and rules are used in the initial design phase. Special ships cannot be designed on that basis, so rework and late alterations are frequent. If the general arrangement plan is produced with a 2D CAD tool, manufacturing and draft-relevant data can hardly be obtained from this incorrect and incomplete 2D plan. A further problem is that often various 2D detail plans exists, which are not co-ordinated with the central general arrangement plan. This problem has been amplified in the last years, since the 2D CAD designs can be multiplied and transformed at will. At times of the drafting board, the information flowed together by a technical draughtsman and he took care of data consistency. It is a misunderstanding that you have to spend more time to produce a 3D model with same detail depth as a 2D design. In contrast to the 3D-model, in 2D 3 views must be always worked out for a correct representation. Even this definition is not enough to define free-form surface. In particular, yachts are often incorrectly represented in pure 2D descriptions.

The portion of plastic models for equipment and outfitting is reduced continuously by the high performance of today's computers. CAD models are more flexible on changes and the 3D data can be used directly in the manufacturing process. For machines, plants, airplane and automotive manufacturers, it is today natural to use 3D models from internet catalogues of the suppliers (e.g. www.partservant.de, www.web2cad.de, www.3dcontentcentral.com). These offer 49 (!) 3D data formats. Most shipyards can use none of these formats. The shipbuilding industry uses special shipbuilding software for equipment and piping. The software tools in shipbuilding have rarely good interfaces, which strongly limits an optimisation of the shipbuilding process. The shipbuilding industry must watch out that it does not manoeuvre itself on the offside here, but should strengthens the standard CAD of formats used by the machine and automotive manufactures. Nevertheless, there is a good shipbuilding market place for CAD models on the internet (www.tribon.com). It is possible to download from this market place ACIS-models, but you cannot import these models into Tribon. The only import possibility towards Tribon is VRML. Up to now, we have no experience with this format, but we think that the necessary alterations of the models are not possible. If you have a model in a STEP AP203, StepAP214, Parasolid, or ACIS format you can convert the model into native (original) formats of most of the CAD systems without any breaks. We have tested CATIA, ProE, AutoCAD, Inventor, Microstation, SolidWorks, SolidEdge, IronCAD, and Bravo. So if your software can import one of these solid formats you are in the mainstream of CAD development. You will have some problems and have to stay outside of the open CAD market. It is a pity that the shipbuilding industry has not realised this development.

Shipbuilding is an assembly industry and needs the equipment from suppliers. The suppliers must and will provide 3D models to be used directly in the initial, basic and in the detail design phase. Here are modelled only the external dimensions of the body and the connection types and represented accurately. The data models are obtained by reduction from existing, complex CAD production models. No shipbuilding software will be used for this work, but of course the CAD systems of the suppliers. It is today still common that the 3D geometry of the ship (apart from the ship theory model) is only specified in the detail design phase. This is a long time after the shipbuilding contract entered into force. Further optimisation is not possible during that stage.

The exceptional shipyards, which start already in the early design phase with a 3D model, show a substantially more favourable cost and performance level than the remainder. Our experience of many years within the range of cost estimation confirm that savings (by error minimization and design optimisation) exceed the costs of a 3D model by far, even if the first model is not transferred in a 3D
detail design tool, but serves only as an error free base for the detail design.

6. Conclusion

Today computer power and CAD systems allow to create a 3D model from the early beginning, but international standard interfaces in shipbuilding software remain to be implemented. These missing standards are:

1. For shipbuilding steel structures the standard interface has to be STEP AP218

2. For the whole ship including steel, the interface has to be STEP AP203, STEP AP214, Parasolid, or ACIS.

If these interfaces were installed in shipbuilding software for import/export (both must be possible), it is easy to create a 3D model from the early beginning. Otherwise, you can forget even a useful VR model, because it is necessary to have a robust and editable model in normal CAD environment. In this case your models are nothing more like nice pictures or movies.

Fig. 10: Rendering
References


KLEHN, B.; JENSEN, H. (1999), ShiPPS (Ship Production Planning System), bmb+f research Förderkennzeichen 18S0104; Kiel

http://cmsa.eurostep.com/
Web-based Tendering Collaboration in Project-Centric Industries

Stein Ove Erikstad, proNavis as, Trondheim/Norway, soe@pronavis.no
Arnulf Hagen, proNavis as, Trondheim/Norway, arh@pronavis.no

Abstract

In the shipbuilding industry, realising and exploiting the full value of specialisation of companies is yet to be seen. We see a gradual transformation of yards, from being complete, fully integrated production facilities, to focus on the assembly of customer-ready solution by integration of “best-of-breed” components and services. Complete packages that include engineering, multi system components and construction work are increasingly outsourced, reducing the share of value-adding taking place by means of own resources.

This development means that the focus for improving competitive position of the yards will shift from internal to external efficiency. Efficient collaboration with suppliers becomes more important relative to improving internal production processes. The supplier will be given more responsibility in terms of proportionally larger (and fewer) system deliveries. We see that the supplier’s role changes into being a critical partner to the yard in the overall build project. And, as a consequence, the yard’s ability to coordinate such partners becomes imperative to reduce costs, lead times and project risk.

We believe this development should trigger yards to rethink their eProcurement strategies. As suppliers turn into project partners, the yards need to move on from simple RFQ exchanges to collaboration solutions covering the entire value chain. Our experience from the development of a procurement portal concept for shipbuilding, SYX (previously ShipyardXchange), has underlined this – while the sourcing for suppliers and processing requests for quotations is important enough, it is merely a starting point. Both yards and suppliers are increasingly ready to take the next step.

The most immediate benefits can be made by web-based solutions covering the tendering work leading up to the contract. Shipbuilding projects are becoming smaller and with shorter duration, and lead times are being reduced. It becomes imperative to exploit the know-how possessed by suppliers for tender development and the corresponding price estimates. And the increasing relative size and completeness of the EPCI packages requires that the supplier assumes more responsibility, requiring deeper involvement in the early phases.

In this paper, we will report from our own experience with developing a toolkit for collaboration in the tendering phase, called TenderSuite, aimed at the project centric industries. We will briefly discuss the challenges and pitfalls experiences by yards and suppliers today, and then present how we believe that this situation can be improved by a web-based collaboration solution.

1. The Tendering Process in the Extended Enterprise

European shipbuilding is characterised by being highly project centric, with ship designs customised according to the individual needs of the ship owner. Traditionally, the Asian (Korea, Japan and increasingly also China) shipbuilding nations have been more oriented towards standardised tonnage in longer series, and with more repetitiveness and reuse in design, engineering, procurement and production. So far, the market focus in Asia and Europe has been complementary. Asian yards have aimed at a low-CGT market, providing advantages to scale and risk-reduction by enabling builds to be based upon known solutions. European yards have targeted a high-value market, characterised by unique specifications for individual owners.

Currently, we see trends that are likely to level out these differences:

- Asian shipbuilders increasingly target the traditional European market segments, with high complexity, build-to-order and shorter series.
- European shipbuilders increasingly strive towards cost efficiency by outsourcing work and working towards reuse of parts of the ship (modularisation).
The Norwegian shipbuilding industry is a typical exponent for the latter, where some of the most successful yards have between 70% and 80% of the value-added (design, engineering services, equipment, hired labour) performed outside the company. This shifts the companies to focus on external efficiency, or the efficiency of the value network, as opposed to the traditional focus on internal (typically production) efficiency. In some sense, this is a paradigmatic shift that transforms companies – surviving shipbuilders in high-cost countries are those that best utilise their collaborating partners.

Fig.1: Margins have a positive correlation with turnover employee, underpinning the strategic migration towards exploiting the yard’s “extended enterprise”

Indeed, a recent study by proNavis (illustrated in Fig.1) found that all yards with a turnover above 300 kEUR per employee showed positive margins, gradually increasing as the turnover increases. And a closer study of the most profitable yards showed that the increase in turnover was mainly attributed to the outsourcing of production and internally produced services. In other words, the yards’ dependency on other companies is growing stronger – success is contingent upon the ability to efficiently use the “extended enterprise”.

Thus far this has evidenced itself particularly in the activities that take place after the project is started:

- The procurement role at the yards has shifted from “simply” purchasing material to co-ordinating and managing complex systems/turn-key acquisitions.
- Distributed engineering is gathering increasing momentum as engineering more and more is performed by systems suppliers – the yard specifies, coordinates and integrates.
- Production is distributed either by module building for subsequent assembly at the yard, or by building the hull in low-cost countries, with subsequent outfitting at the yard.

Technology development has responded to this shift in focus towards external efficiency. In recent years we have seen the emergence of a multitude of web-based solutions targeted towards the shipbuilding industry, such as procurement portals, collaborative engineering tools and support for distributed manufacturing. This pattern will also be recognisable in most other project-centric industries, like the plant fabrication, entrepreneur and offshore industries. However, so far the focus has been on the post-contract phases of the ship life cycle. In this paper we will present how
web-based collaborative solutions can be applied in the pre-contract phases, during tendering development.

1.1 The tendering process

A typical medium-sized yard annually responds to between 50 and 150 tenders for shipbuilding projects. Most of these are handled rudimentary, with fairly coarse specification. However, a fair share will involve quite elaborate work that results in an extensive build specification. In view of a situation where more yards compete on each call for tender, it may be expected that it will become of increasing importance to efficiently develop good bids.

Fig.2: Involving suppliers in the tendering phase will improve the quality of decisions made in the early stages – decisions that typically bind up a major share of the total project costs

The build specification developed in the tendering phase will bind up most of the costs (a typical share cited is 70%) in a shipbuilding project. Build specifications are generally created by copying documents related to previous tendering projects, often without any verification by others in the shipyard than the sales or technical department, and even rarer outside the shipyard. This copying regularly carries over- or under-specification, errors and outdated solutions into the new potential project from old ones. The practice will tend to:

- conserve technology development
- drive the price upwards and
- increase the risk for cost overruns or price underestimation

Such errors or discrepancies may be alleviated through several means:

- Carry fewer errors into the project. This may imply to ensure that the basis (either copy or template) is the best match for the present project or at any time is updated
- Introduce QA functions in the tool. This may imply that the sales engineer is given tools to check for typical errors or to check off the effect a system specification change may have on other systems.
- Enable internal collaboration. This may imply that there exist procedures and tools to engage others in the company in QA (from technical, procurement and even production).
All these are means that may be introduced into the yard itself, given proper tools. However, this will again improve “internal efficiency”, which is important enough but which only resolves part of the problem.

1.2 External collaboration in the tendering phase

As is the case after a build contract is signed, there are several ways in which the yard can exploit the competencies and experience of external companies also in the tendering process. Examples of this are:

- **Supplier.** Though several (most) suppliers are to be exposed to competition once the build contract is signed, several (typically large systems suppliers) are also either de facto or formally chosen before such signing. Especially for those that in the project will be responsible for delivering a large share of the value adding (for instance, propulsion, HVAC, electro, bridge systems, cargo handling, special functions), it is possible to give them partial responsibility for developing the specification of their respective deliveries in the overall build specification. These are obviously also relevant when it comes to cost estimation.

- **Sub-contracted yard.** Sub-contracted yards that are to build modules or a complete hull to be assembled and/or outfitted at the shipyard may also participate in the development of the build specification or the cost estimate.

- **Ship design consultancy.** Ship design consultancies that deliver the original design are able to verify that the specification developed by the yard actually describes the most recent technology, and not the solutions that existed at the time when the project from which the build specification is copied was built. This also goes the other way; a design firm developing a build specification on behalf of a yard will obviously want to include personnel from the yard in the process.

- **Sister (or cooperating) yard.** A growing number of shipyards are owned through larger enterprises, such as Aker Kværner Yards. In these enterprises there will often be a desire to use the accumulated know-how for the benefit of all the yards, both when it comes to sharing experiences from previous projects and when it comes to developing a project in a particular tender.

- **Others.** Other external actors that have or may have roles in the tendering process are classification societies (rules, preliminary verification), rented labour companies (price estimations) or procurement portals (price requesting).

This all indicates that also the tendering, or really the project development, process becomes distributed. Indeed, it is a necessary consequence of distributing the value creation in the project – the one in this extended enterprise with the best know-how to deliver should also be a valuable contributor when it comes to specifying the delivery in the build specification.

There are some significant and non-trivial issues related to this:

- How may a yard, in a very short and intensive time-frame in a tendering phase, engage external parties to work on tenders (particularly those that are viewed to have good chances of success)?

- How may the issue be resolved regarding incentives when giving e.g. a supplier the option to comment on or even develop the specification of a delivery that the same supplier in the end may be responsible for bidding on and delivering?

2. The TenderSuite development: Web-based tendering collaboration

During the last couple of years, proNavis has, in close collaboration with industry partners, been developing a suite of software tools to support the pre-contract phases of the ship lifecycle. This solution will become commercially available during 2003. The first module to be completed within
this suite is a project calculation module. A specification module will follow towards the end of year.

![Fig.3: The TenderSuite Specification Module](image)

The overall objective of the development of the specification module is to increase the competitive position of shipyards by facilitating the development of shipbuilding tenders that (a) are cost optimised, (b) reflect the most current technology and (c) minimised with respect to both technological and cost estimation risks. This shall be done by devising means to easier mobilise the accumulated know-how in the extended enterprise as resources in the yards’ project development phase.

The following high-level functionality has been singled out as being central to the solution:

- Functionality to improve the alignment between the owners’ requirements and the build specification (more exact specification).
- Functionality to reduce the risk of basing the project on a build specification that carries with it cost- or risk-driving features (improved QA).
- Functionality to reduce costs by enabling dedicated suppliers to propose functional solutions that may be advantageous to the project at large (improved solutions).

### 2.1 The Scope of a Specification Development Tool

What is then the core functionality and features that need to be supported in a tool for the specification development process within a yard? We believe the following points are of major importance:
• Creating new projects, either from scratch, from a template, or by copying an existing specification

• Developing the specification by adding text, technical data and associated documents and drawings, working in a top-down approach based on a system-oriented information structure.

• Planning and following up the build specification process, by scheduling and assigning responsibilities at selected levels of detail, status assessment both on detailed and aggregate levels, and automated routing and notifications based on selected events.

• Access to classification rules and archived information corresponding to system-specific context and global project settings

• Quality control mechanisms, such as system specific checklists and formal approval work processes on preferred level of detail.

• Establishing the Maker’s List, and managing suppliers, providing easy access to information exchange and collaboration functionality in a system-specific context.

• Generating the final build specification document according to predefined yard standards, on the basis of the information entered in the specification development project.

2.2 Improving the specification process

With the TenderSuite development, we aim at alleviating some of the problems associated with the process that we have discussed earlier in this paper. More specifically, we have focused on the following key areas for improvement:

• Effort: Efficient reuse of information from previous specifications, while at the same time ensuring that the particular requirements and constraints of the current project are taken into account.

• Quality: Continuous quality assurance of the specification through extensive collaboration both internally in the yard and externally towards suppliers, consultants, classification society and ship owners.

• Time: Shortened development time cycle by a structured breakdown of the specification into smaller, manageable pieces, coupled with facilities for distribution, concurrent development and follow up.

These features need to be built into the application architecture on a low level. It represents a considerable deviation from the current document-centric, somewhat “ad-hoc” specification process. As a consequence, devising a suitable information and process model has been a core challenge in the development work. In the following, we will briefly discuss some of the most important issues that we faced, related to:

• The underlying data model, combining structured product data with unstructured, document-oriented content.

• The use of templates to capture and reuse the shipyard’s “corporate knowledge”, and how inheritance mechanisms can be applied for reusing information at different levels of abstraction

• Reusing previous specifications through “deep” and “shallow” copying, and what mechanisms that can be used to avoid the pitfalls discussed earlier in this paper, such as over-specifications, project-specific variants, and the carry-through of previous errors.

• The tendering process model, based on a “divide and conquer” principle, allowing for distributed, concurrent specification development process with multi-level status and progress assessment.
The involvement of external parties in the specification process, in particular system suppliers, using role-based access and security combined with both web-based collaboration technologies and simple RFI/RFP processes similar to those found in eProcurement systems.

2.3 Mixing structured product data with "unstructured" document text

To incorporate features such as automatic processing capabilities and fine-grained, filtered reuse of information, an application like TenderSuite needs to be based on a structured information model. At the same time, we need to take into account the fact that the end-product – the build specification – is typically a text-oriented document, where phrasing, style and formatting often are considered key factors in achieving a high quality end product.

We believe there is not one correct answer to what is the ideal mix between these two opposites. Too much structure tends to impede flexibility, and to enforce the end user into a process that is not adapted to his or her preferences. On the other hand, lack of structure means that the individual information elements are not recognisable, making “intelligent” reuse difficult.

In TenderSuite, we have devised the following principles to concurrently support both the data-oriented and document-oriented aspects of the information model:

- All core business domain concepts, such as Project, Supplier, Specification Item, Template, Document, Vessel, Makers List, etc., are represented explicitly in an object-oriented model. All other information elements, such as text blocks, parameter sets, documents and drawings, are logically attached to the entities in this domain model.

- The build specification is modelled as a hierarchical structure of specification items, based on yard specific group systems. For Norwegian yards, this is typically the SFI system. This is the main navigation structure in the application, and also the basis for distribution and follow-up of status and progress.

- The use of a hierarchical product breakdown structure as a backbone in the information model enables a top-down approach in the specification development process. Thus, the complexity of the specification may well range from the one extreme of consisting of a single specification item defining top-level vessel data, to a large structure of items, each defined on a detailed system/component level, reflecting a complete build specification document.

- The resulting build specification document as such is only a view on this model, compiled by parsing each item in the specification tree, extracting both structured data sets, unstructured, pre-formatted chunks of text, and attached documents, figures and drawings.

2.4 Providing a baseline by copying – pitfalls and possibilities

As was discussed in the beginning of this paper, uncritical copying of an existing specification as a baseline is perhaps the most common source of errors in build specifications. Yet it represents a powerful means to efficiently establish such a baseline, so discarding it is not an option.

In TenderSuite, we have built in support for controlling this process to avoid some of the negative consequences that may occur, such as:

- All copied items are automatically given the status “In progress” (and not “Released”), requiring an explicit approval from the person responsible. This may be overruled by changing the top node status, enforcing this status change for the complete specification. Still, it will require an explicit action that is likely to trigger a more considerate attitude.

- Each item may have a list of control points that need to be checked before the item is approved. Again, this is likely to rule out some of the typical recurring errors.

- As an alternative to use a complete, existing build specification as a baseline, the user has the option of copying existing specification items one by one. In addition to providing more
flexibility in terms of mixing items from a large repository of previous projects, it also gives more transparency to what is carried over into the new one.

2.5 Capturing and re-using corporate knowledge through templates

For most end users, a template is a familiar concept, representing an efficient way of establishing a baseline specification. The concept is simple: The template provides the information that is expected to be common across a large collection of specifications – and the user fills out what is missing.

One of the problems with templates – in this setting – is that they apply to different level of generality, and that the relations between these templates are innately hierarchical. Typically, we find an inheritance relationship along the vessel type axis.

For example, a yard may use a generic template that is valid for all different vessel types they build. This template will typically be based on the yard-specific group system, and may contain general descriptions and specifications of the yard’s standard solutions and configurations. For a particular vessel type, say, a RORO ship, part of the general template may still apply, while at the same time some parts may be irrelevant, missing or being specified on a too high level of generality. The challenge we faced in TenderSuite was to support specialised templates that would provide a high quality template for a particular vessel type, while at the same time avoid the configuration burden of maintaining a large number of partly similar templates.

![Diagram of vessel types hierarchy](image)

Fig.4: An example of a hierarchy of templates, following the vessel type axis

The inheritance mechanism we have implemented is simple, but effective. Each specification item in a template maintains, whenever relevant, an explicit relation to an item in a parent template. When generating a new project baseline based on a template, an upward recursive search mechanism ensures that the most specialised item will be used. Thus, if a user wants to create a new template, either for a particular vessel type variant, or simply capturing a set of personal preferences, it may be based on an existing template, leaving the user to specify only the differences from the previous template.

Using the template hierarchy above as an example, the “Vessel” template may contain a baseline specification of generic ship systems (painting, HVAC, electrical, etc.). The “RORO” template may reuse (inherit) all data in the “vessel” parent template. In addition, override the HVAC specification to comply with car deck requirements, and add a specification item related to the stern ramp.

A side-effect of this template hierarchy is that it represents a means to describe and maintain the yard’s vessel type and variant hierarchy, in a PDM-like approach.
We believe an efficient template mechanism like this to be imperative in increasing the efficiency and quality of the specification process. Templates are efficient for information reuse, while at the same time they don’t create some of the typical pitfalls associated with simply copying projects. They may be somewhat costly in terms of up-front investment and continuous maintenance, but we believe the payback period will be short considering the gains to be made.

2.6 The Tendering Process Model

Typically the specification covers a large array of different system areas and disciplines, requiring deep and specialised knowledge and experience. Thus, it is preferable to be able to draw on an extended pool of personnel resources to participate in this process, both internally in the yard, and by involving external parties such as system suppliers at an early stage. However, from a practical point of view, a distributed development process poses several challenges in terms of following up progress and status during development, and in configuring the final specification with respect to consistency and overall quality.

We have added several mechanisms that will alleviate the effort related to a distributed development strategy:

- Assignment of responsibility on individual specification items. This responsibility involves both the local item definition, and the complete configuration of all sub-items originating from the item. Thus, this opens up for distributed, concurrent development on a fairly detailed level, while following up status and progress on an aggregate level.

- Status assessment on individual specification item, with recursive aggregation at higher levels. Status changes are triggered automatically based on user interaction, e.g. from “Not started” to “In progress” when the item is first saved, from “Internal release” to “External release” when the item is internally approved, and so on. Status changes on a high level in the hierarchy will recursively enforce changes further down. E.g. if an item on the 2nd level is internally approved (say, 70 – Fuel System), the status of all child items (say FO transfer system, FO purification system, and so on) is also changed to maintain the consistency of the specification status.

- Notification and approval processes on specification item level. Upon selected status changes, a notification is automatically sent to subscribing users, based on formal role (e.g. sales manager, technical responsible), system association, or user-specified preferences.

2.7 Involvement of suppliers in the specification process

Both yards and suppliers agree that it would be advantageous to bring the suppliers into the specification process, preferably as early as possible. However, it happens infrequently in real life. Part of the explanation is that it is difficult to accommodate within the tight time and resource constraints that is typical for the specification process. At least to some extent, this situation can be improved by providing support for efficient information exchange and collaboration within a web-based application environment.

We have looked into three ways of involving the supplier. First, the supplier can be asked for input to the specification using the built-in RFI/RFP facility. The user will have access to suppliers for the specific system in question, based on entries in the Maker’s List, and/or the yard’s list of pre-qualified suppliers. Second, the user may grant review access to a supplier on a particular specification item, providing the means for a quality control. And third, representing the most comprehensive mode of collaboration and supplier involvement, the yard may assign the responsibility of a part of the specification to the supplier. The overhead related to this, at least from a process point-of-view, is minor in a web-based environment. The critical factor is more related to the level of thrust and confidence that is required to make this work in real life.
3. **Future enhancements**

With the TenderSuite development, our aim has been improved support for the build specification process as such. This process is based on structured data handling in a collaborative web environment. This opens up several opportunities for the yard in the longer term, having a wider scope in terms of functionality and information reuse. Examples of this are:

- Reusing the specification structure and content in procurement: Typically, a Request for Quotation (RFQ) is based directly on a specification item. Thus, considerable effort is likely to be saved if the content and documentation can be transferred efficiently, either by copying item for item, or by letting the complete specification structure be used to build up a procurement plan structured according to the yard-specific system breakdown structure. For instance, a project may directly be established in a procurement portal like ShipyardXchange.

- Reusing the specification structure and content in engineering: As mentioned previously, the build specification forms the design basis for the engineers when starting class and engineering design. Giving an engineer easy access to the parts of the build specification relevant to the system being designed will likely improve efficiency and reduce error sources. In addition, enabling engineering change orders to be related to the parts of the specification to which they correspond, both transparency, ability to trace consequences of the change and feedback into the original baseline are potential benefits.

- Extended support in the specification process. In a web environment, the user is enabled to reach out to other parties that provide various forms of support. This may for instance be access to supplier web sites offering product information and technical data, or online configuration tools. It may also be access to functionality for analysis and calculations, which may be made available either as embedded tools or as web services.

- Access to certified suppliers and products. Classification societies have registers of suppliers that have products type approved under their regimes, and increasingly also certification of the maker itself. Such information will to a growing extent be accessible in a web-based environment, such that a shipyard already upon preparation of the offer to the customer (and the Makers List) may have an indication as to which suppliers may respond rapidly to requests, dependent upon the chosen classification society for the project.

- Design dependency modelling and change handling. As previously mentioned, a significant source of errors is that changes to one specification item is not captured in those directly or indirectly affected by it. Such changes either spread through functional dependencies among the different systems or through system-wide impact from certain high-level changes. It is in principle possible to model such relationships, especially those that spread through system dependencies. A dominant share of such dependencies is the same from project to project and may be modelled as part of a template. In that case, notifications may automatically be given to those responsible for dependent systems when changes are made. The more parameterised and structured the specification is, the easier it will be to establish such a built-in logic.

We believe that web-based solutions covering the tendering work leading up to the contract will play an increasingly important role. Shipbuilding projects are becoming smaller and with shorter duration, and lead times are being reduced. It becomes imperative to exploit the know-how possessed by suppliers for tender development and the corresponding price estimates. And the increasing relative size and completeness of the EPCI packages requires that the supplier assumes more responsibility, requiring deeper involvement in the early phases.
Kernel Architecture for the Development of CAD/CAM Applications in Shipbuilding Environments

Antonio Rodríguez, SENER Ingeniería y Sistemas, S.A., antonio.rodriguezg@sener.es
Carlos González, SENER Ingeniería y Sistemas, S.A., carlos.gonzalez@sener.es
Iñigo Gurrea, SENER Ingeniería y Sistemas, S.A., inigo.gurrea@sener.es
Lluis Solano, Universidad Politécnica de Cataluña, solano@lsi.upc.es

Abstract

The capabilities of complex software products such as CAD/CAM systems are strongly supported by basic information technologies related with data management, visualization, communication, geometry modeling and others related with the development process.

These basic information technologies are involved in a continuous evolution process, but over recent years this evolution has been dramatic. The main reason for this has been that new hardware capabilities (including graphic cards) are available at very low cost, but also a contributing factor has been the evolution of the prices of basic software.

To take advantage of these new features, the existing CAD/CAM systems must undergo a complete and drastic redesign. This process is complicated but strategic for the future evolution of a system. There are several examples in the market of how a bad decision has lead to a cul-de-sac (both technically and commercially).

This paper describes what the authors consider are the basic architectural components of a kernel for a CAD/CAM system oriented to shipbuilding. The proposed solution is a combination of in-house developed frameworks together with commercial products that are accepted as standard components. The proportion of in-house frameworks within this combination of products is a key factor, especially when considering CAD/CAM systems oriented to shipbuilding.

General-purpose CAD/CAM systems are mainly oriented to the mechanical CAD market. For this reason several basic products exist devoted to geometry modelling in this context. But these basic products are not well suited to deal with the very specific geometry modelling requirements of a CAD/CAM system oriented to shipbuilding.

The complexity of the ship model, the different model requirements through its short and changing life cycle and the many different disciplines involved in the process are reasons for this inadequacy. Apart from these basic frameworks, specific shipbuilding frameworks are also required. This second layer is built over the basic technology components mentioned above.

This paper describes in detail the technological frameworks which have been used to develop the latest FORAN version.

1. Introduction

CAD Systems are complex products, the main reason being that they involve several of the most sophisticated technologies used in the development of software applications. General purpose CAD Systems require high performance visualization, computational geometry complex algorithms, advanced ergonomic user interfacing techniques, etc.

These characteristics lead to the fact that every one or two years a new release of the most popular CAD Systems is launched on the market and that every two or three releases a new version appears, which introduces major changes, mainly philosophical, that affect the design of the application.

In addition to this, there is a general trend in 3D graphic API standards, which every ten years suffer a drastic change motivated by several factors:

- Advancements in graphic and modeling knowledge.
- Increasing power on graphic acceleration chips.
• Evolution of operating systems.
• The choice of primary languages used for programming.
• The type of users for 3D graphic applications.

Examples of these important changes are the appearance of 3D GKS, PHIGS and PEX in the early 1980’s and the appearance of OPEN GL and DIRECTX in the late 1980’s. The first of these API’s was accessed with programming languages such as FORTRAN and C and the second generation API’s are accessed with C++.

Shipbuilding CAD Systems suffer somehow a very different evolution path. As these are CAD products targeting a very specialized segment of the market, with a very specific know-how, the capability to react to these important changes has been more limited compared with general purpose CAD applications.

As the periods between big jumps in the evolution of these CAD Systems oriented to shipbuilding increase, when they may accumulate much more technological changes. This paper is intended to describe the new generation of shipbuilding CAD systems based on technologies, which if not revolutionary, are at least innovative when used in this area.

This paper is basically an enumeration of these technologies and the way they are used in shipbuilding CAD Systems. Technologies described are:

• Database
• Visualization
• Topology
• Product model

The following paragraphs will explain how, by means of these technological frameworks, it is possible to build up the most sophisticated CAD tools for shipbuilding designers.

2. Database Framework

This framework must offer some of the most fundamental services when developing a CAD System. These are a persistence service, which is a basic infrastructure, and concurrent engineering facilities, which is a strategic added value to a shipbuilding CAD System. Each of these services is explained in the following paragraphs.

2.1. Persistence service

While the first shipbuilding CAD Systems used files to store information, the increasing complexity of the systems, covering more areas of the design, called for new tools to handle large amounts of information in a fast, flexible way.

At the same time the increase in the number of users working simultaneously on the same ship called for concurrent access to the ship’s information. In response to these requirements databases started to be used to store the information, partly or totally.

In the first instances relational databases were used to store the attributes of ship components, while in some of the second instances proprietary databases were developed to store geometrical and topological information of the ship model, as well as attributes of its components.

During recent years there has been a trend to replace the proprietary databases of integrated CAD Systems with commercial databases with standard access languages and database server capabilities. The main reasons for this change have been the ease with which the users can develop new
applications and the possibility to reduce the amount of information transferred between each user and the database. The latter is not very important in a LAN, but it is crucial in a WAN.

The use of object oriented databases, proposed in recent years, has not had wide acceptance, as the standardization, concurrency and access capabilities of object oriented databases are considerably behind those of current commercial relational databases.

On the other hand, there is an important issue when using a relational database to store the product model created with an object oriented CAD System. The challenge is to connect both paradigms without losing the main advantages that each of them offers.

The object oriented paradigm is the natural path to develop a consistent and powerful product model and all the software components related to it. It is accepted that this would be the leading paradigm to which a relational model would have to adapt.

This strategy consists of using the relational database as a persistent layer for the object oriented world. From the object oriented perspective, it would be easy to replace this layer should a new storage technology appear. But this layer must be able to exploit the advantages that the relational databases technology offers to the application developer. This question will be analyzed in more detail in the following chapter devoted to explain the concurrent engineering facilities.

To cope with the problem posed in this way, the persistence service must supply the following features:

- Offer a transparent access to the persistent objects. The application programmer should not need to use SQL directly to receive and access an object stored in the database
- Support for inheritance
- Support for polymorphism
- Unique object identifiers
- Associations between objects (1 to 1, 1 to n and n to m)
- Automatic caching of objects in memory
- Automatic storing of modified objects (commit)
- Access via smart pointers to avoid multiple instances of the same object and keep the pointers always valid
- Access to multiple databases
- Centralized lookup and search mechanisms (search conditions)
- Support for proxy objects for accessing parts of an object, e.g. showing a list of equipments with description without reading all the information for the equipment objects. This may be solved by lazy on-demand materialization not only for objects but also for object attributes
- Transactions (commit, rollback, save-point)
- Multi user access
- Locking of objects
- Serialization of objects

All this complex functionality can be encapsulated for the object oriented developer, as the object model is leading the development process. The best way to do this is by means of generative programming. When creating the object model, the software developer could annotate with specific persistence attributes the code to specify the storage behavior of the objects. A pre compiler process would automatically generate the corresponding object factories to which would be delegated the responsibility of managing the objects life cycles.

The use of generative programming to cover this gap is the most productive way to accomplish this task. Additionally, generative programming is the best methodology to avoid error prone development process, to improve the quality of the software produced in this critical area.
2.2 Concurrent engineering facilities

Concurrent engineering has, for a long time, been an important issue for shipbuilding CAD Systems, specially in the area of detail design. This need led to the product model concept, as the single source of product information. The existence of a product model allowed the concurrent engineering practice, at least in the context of a local area network. CAD systems based on a single product model for all the disciplines are generically called integrated systems.

In this kind of system, the coherence and compatibility of the work performed by different users is obtained by using a single database to store one single ship model for all users. This method guarantees that the quality of the engineering work is far superior to that which can be reached when a ship model is not used.

This advantage can be obtained when all of the users are working in the same local area network (LAN) to access the common database containing the ship model. When the workstations are geographically distributed, it would be necessary to access the model through a wide area network (WAN) using public resources for data transmission and the necessary bandwidth at a reasonable cost.

Until recently, several factors have contributed in preventing the extension of the use of integrated systems to this environment in a proper way. One is the high communication cost associated with the required bandwidth, which is very high in the case of most integrated systems. Another is the fact that the database management software used to manage product models has not been designed to reduce data transmission ratios.

There are two different strategies to cope with these communications problems that may result in a CAD System which is particularly well suited for concurrent use from remote sites:

1. Use a largely topological definition (rather than geometrical) of the product model. The volume of the database is lower by an order of magnitude and the amount of information to be transferred decreases accordingly.

2. Take full advantage of the infrastructures and capabilities that a relational database with SQL and database server may offer in this area. This reduces the required bandwidth by an additional order of magnitude.

The topological definition of the product model is explained in more detail in a specific chapter of this paper. With respect to the second point, two database architectures are available to deal with scalability issues.

The first one is preferred because it allows working in a WAN exactly in the same way as in a LAN. This architecture requires a single database to which remote users are connected via frame relay, ISDN or any other corporate communication infrastructure. In this way, a large number of users can be working concurrently from remote locations on the same ship model.

A good example of this is the case of the Kvaerner Masa-Yards company, which has implemented FORAN System for basic ship design in its Turku and Helsinki shipyards. The figure 1 represents the configuration at Kvaerner Masa shipyards. End-users at Helsinki or Turku run the FORAN modules from a local server. FORAN database is used from a local database server or from the other site, depending on the project. Some common files (such as hull forms and user-defined macros) are located in a shared disk to keep them exactly the same for both shipyards. There is a 100 Mbit/s network at each shipyard and the dedicated line between the shipyards is 30 Mbit/s.
This solution allows concurrent engineering between the shipyards. The performance has proven to be acceptable with a scalable number of users.

When the number of remote users is very large or the available bandwidth is insufficient, the second architecture (distributed) can be used as an alternative or a complement. This architecture is based on the distribution or replication of databases between the master location (shipyard) and the remote sites (subcontractors). The coordination performed by the shipyard, owner of the ship model, can be made in a very easy and efficient way.

3 Visualization Framework

A visualization framework has been developed in order to display and manage the 3D graphic scene of a design. This visualization framework has been implemented as an application program interface (API) that works with a graphic library. Working with a set of visualization functions provided by the API instead of working directly with a graphic library, gives a great flexibility to the system visualization in, for example, the case of changes in the graphic library. At present, the visualization release of this API works with OpenGL, but can be easily changed to work with, for instance, DirectX.

The visualization framework provides the usual visualization tools that are needed in 3D design, such as zoom, pan, different point of views with high performance in order to deal with large visualization scenes. In this sense, a Level of Detail (LOD) management has been implemented to achieve the best performance in complex scenes.

Functionalities for efficient selection and filtering tools are also available during the design process. There are also tools to hide scene components in order to speed up interactive graphical manipulation of a 3D scene. Visualization attributes of the different scene components can be easily configured by user.

The visualization framework includes tools to customize and manage different scene views. There are available different visualization methods such as wire frame, hidden lines removal or shaded view. In order to simplify and clarify the work in a complex scene, the visualization method works at component level, this means that in a same scene the designer can have different components displayed in different methods (see Figure 2). Also different levels of transparency can be applied to individual components or groups.
Fig. 2: Outfitting model with different visualization methods (Shaded and Wireframe) and transparency.

A basic navigation tool, for use during the design, is also provided by the visualization framework. This functionality allows a basic inspection mechanism through navigation in a scene. There is also a specific and high-performance environment for inspection and navigation in large scenes.

4 Topology Framework

The management of the relationships and dependencies among the parts involved and the automatic updating of the geometry are necessary in order to improve the efficiency and the flexibility during the design tasks. The relationships defined among different components allow the establishment of dependencies for changes. If an element A is modified, the elements depending on A and related with A, must be updated in position, dimension or both. We define these dependency relations among the involved elements as *topological constraints*. The topology framework provides the structure and functionalities to manage the existing topological constraints in a ship design.

*Topological constraints* have to be kept in the product model to maintain changes in the design and to update the location, shape and dimensions of the components and parts.

Some of the structural parts are directly related with the hull surface. In fact, there are components such as frames, which are defined on the hull and follow a curve on the hull surface. This means that any change in the hull surface has to be transmitted to the related components. A modification or refinement in the hull of the ship implies the redesign of the dimension and position of the structural related elements such as deck surfaces, profiles and plates. Using *topological constraints* the elements can be updated in an automatic way.

Modeling parts are related with *topological constraints*, where the location and dimension of a part depend on the dimensions and locations of others parts. This can be understood as an adaptive modeling.
Due to the concurrent nature of the application model described, topological relations must be handled for all components of the database whether they have been read or not. This is performed by the use of ghost parts and database autoload procedures.

More specifically, it can be considered the example in Figure 3 that represents a simplified transversal section of one of the sides of the ship.

![Figure 3: Simplified transversal section.](image)

Ship hull modifications must be propagated to the main deck, contiguous panels, lower decks and center plane reference, as they produce changes of their shape or location. The center plane reference must propagate its shape modifications to the different decks and plates. Changes in the shape or location of the lower decks can modify the position of the ship equipment; this forces specific modifications in the associated pipes which end up in changes in the panels that must be crossed by the pipes.

Figure 4 shows the corresponding topological constraints of Figure 3 components.

![Figure 4: Example of topological constraints among ship components.](image)
The above explained case is a simplification in order to state the problem. In fact, in real ship design, a large number of components in each section are involved. In average, there can be from fifteen to a hundred sections and from a dozen to several hundred structural components as frames, plates, decks, and panels.

A ship is designed hierarchically, by composition of simple parts into more complex ones. The natural way of representing the dependencies and relationships is by using a directed acyclic graph (Rappoport 1993). From a directed acyclic graph that represents the topological constraints among ship components, it is possible to compute the sequence to perform the automatic update of the components.

We define the **topological constraint graph** as a directed acyclic graph where nodes are components of a design and edges represent the topological constraints.

The update process of the components related with topological constraints has to be performed in a specific order. In Figure 4, the evaluation of node P has as a prerequisite the evaluation of nodes Q, B, C and H. To update the components after a change, it is necessary to process the nodes in such an order that no node is processed before any node which points to it.

Using a **topological sorting** (Aho 1983) it is possible to compute a linear ordering of the graph nodes, such that if there is an edge from node i to node j, then in the linear ordering, node i appears before node j.

**Topological sorting** is based on a graph depth-first search. The cost of a **topological sorting** is linear (O(n) where n= number of nodes). In the case of Figure 4, the **topological sorting** produces the sequence: C H Q A B F P E D.

A generic and integrated data structure, decoupled from the product model database, is proposed (figure 5). This data structure allows managing and updating of the geometric elements (parts) related with topological constraints. The data structure is based on a directed implicit graph where the nodes are ship structural components each one maintaining its own list of related components (topological constraints). The graph is explicitly traversed when the update process is needed. The ship structure update is performed by a set of class objects that have access to the ship structure components.

The present software development is based on object oriented techniques and pattern-based design (Gamma 1994) in order to have a high level of software reusability and decoupled data structure.

### 5 Product Model

One of the key factors that increase dramatically the complexity of a shipbuilding CAD System is the different modeling capabilities which are required through the different stages of the ship engineering process. This happens in the context of a very short design life cycle, except for navy shipyards, where the demands are for very flexible and powerful modeling tools.

In shipbuilding, the complexity of the product model increases through the different design cycles. Concept design contains a small amount of information, classification design contains much more information and detail design contains a very large amount of information.

In the initial (concept) and basic design cycles the most relevant elements of the product model are defined by means of surfaces. This model of moulded surfaces of the ship also constitutes a topological reference system for the definition of any other element of the model. This requires a close integration with any other modeling capability.
In detail design, the most critical requirement for the product model is the integration of all the disciplines, which in contrast is better implemented by means of a solid model with powerful interference checking capabilities. As mentioned previously, frequently the definition of the solid elements is related or based on the surfaces elements, which requires the extensive implementation of the topology with sophisticated algorithms for surface to solid conversions.

Each of these types of models is explained in more detail in the following paragraphs.

5.1 Surface Model

The surface model must contain all the valuable model information developed during the initial stages of the design. To be able to define all the moulded surfaces of the ship (hulls, decks, bulkheads, superstructures, appendages, etc) a powerful and easy to use formulation must be used.

The most difficult requirement that a surface model must cope with is the capability to support the fitting and fairing process. Fitting and fairing tasks are usually the starting point of the ship design process. These tasks take a considerable amount of time to be completed, while the resulting product is essential information for the following engineering works.

To deal with this bottleneck, several strategies have arisen. The most effective is that presented by topological integrated systems. In this kind of system, as soon as a preliminary surface model is obtained, the following processes can start to work with this information. In parallel, the surface model is refined and fairied, without preventing other tasks from going on. Once the surface model is finished, the work performed on the preliminary model is automatically updated, based on a topological definition. Topological definition is invariant for small changes (such as those changes introduced by the fairing process).
Under this scheme, a complete and effective surface model must fit the following requirements:

- Fast fitting of the ship surface model within a stated accuracy (hull, double hull, superstructures, decks, bulkheads, etc)
- Easy and intuitive fairing process based on sophisticated algorithms, allowing the user to be freed from dealing with complex mathematical concepts.
- Oriented to a topological definition. The basic geometry is defined only once. Other information is defined with respect to this geometry. This includes management of ship surface model transformations and local and global changes.

State of the art technology related to surface modeling indicates that the most comprehensive yet powerful formulation is the NURBS (Non Uniform Rational B-Splines). The cost to pay is complexity. This complexity is reflected with a joke about the acronym (Nobody Understands Rational B-Splines).

To develop the powerful tools to facilitate the fitting and fairing process based on this complex formulation, sophisticated algorithms must be implemented. But even with these algorithms, the direct manipulation of a NURBS surface is a complex task for the designers. The shipbuilding designer is traditionally more used to working with curves representing a set of planar sections as a mean to deal with the complexity of the ship surfaces, and is able to perform the fitting and fairing process based on these sections.

Thus, the best approach to build a surface NURBS model must fit these requirements:

- The user should be able to work with curves, because it is much easier to manage curves than surfaces
- Surfaces should be constructed using curves to define their geometrical properties.
- Direct manipulation of surfaces (by moving the control points) should be reduced drastically
• The philosophy of the method should be that when the surface is produced it is close to the surface required, reducing drastically further manipulations
• Trimming of surfaces is necessary to avoid limitations of quad topology
• The user should be able to directly edit points on the surface
• Automatic fairing algorithms should be available
• The user should be able to interchange geometrical information using most common standards (IGES, STEP, DXF, etc)

Fig. 7: Surface represented by sections

These have been the premises which have been used in the development of the surface model framework of the FORAN System.

5.2 Solid Model

The main goal of the solid model is to have a 3D geometric representation of the different ship components. The solid model representation is a virtual geometric prototype of the ship that can be tested and validated. The solid model must be able to offer functionalities and modeling operations related with part design and assembly design. These are two main environments with different sets of requirements.

In order to achieve a high level of productivity, the part modeling is based on primitive creation functions. A set of 3D primitives is used to model the different parts. The part modeling environment allows to model more complex parts defining a group of basic primitives and the 3D location of each part. Using 3D primitive geometric elements it is possible to have an efficient geometric representation of the elements involved in ship design.

The assembly design environment provides a logical structure for grouping and organizing parts into assemblies and subassemblies. The user is able to identify individual parts, keep track of associated part data, and maintain the relationships among parts and subassemblies. Tools for assembly modeling have been developed in order to achieve a high level of productivity. These tools include navigation,
inspection and manipulation of large, complex assemblies. Components can be selected by all types of attributes or spatial location.

Assemblies can be created using the top-down or bottom-up design methodology or a combination of both methods. Bottom-up design is the traditional method. In bottom-up design, parts are created, inserted into an assembly, and relationships are established as required by the design. Bottom-up design is the methodology used when the design is performed with previously constructed parts. In bottom-up design component parts are designed independently and thus allows the designer to focus on the individual parts. Top-down design allows to design a component in the assembly environment. The geometry of an assembly to define a new component and its relationships with the other parts is used to fix the location.

![Image of shipbuilding specific parametric objects for hull](image)

Fig. 8: Example of shipbuilding specific parametric objects for hull

The solid modeling environment has been designed to optimize shipbuilding specific modeling environments such as piping and structural parts modeling. Automatic tools for piping are required to achieve a high level of productivity. In the same way, modeling of structural parts also has a set of modeling operations for high performance in the design of profiles, plates and panels (Fig. 8).

Finally, the solid model environment provides a set of different file formats in order to export and import components parts or assemblies designed in other CAD applications. Specific methods have been implemented to optimize the geometry representation of the imported components.
References

AARNIO, M.,(2000), *Early 3-D Ship Models Enables New Design Principles and Simulations*, 1st COMPIT, pp.5-17


Practical Application of Virtual Reality Technology in Engine Room Design

Christoph Baier, ORCA Engineering GmbH, Bremerhaven/Germany, c.baier@orca-germany.de

Abstract

The possibilities of Virtual Reality (VR) technology for use in shipbuilding industry are extremely dependent on the cost and benefit ratio. This paper explains today’s possibilities of VR technology in relation to its cost and the plastic models used so far. An example of our daily routine with the VR tool is taken to illustrate the present situation. Plastic model and VR model are compared to point out the varied possibilities resulting. The basic requirements and problems of making a VR environment are discussed along the possibilities of applying the model after the initial design stage. Different technologies of presentation and their benefits in various cases of application are briefly reviewed.

1. Introduction

For the description of present day’s possibilities for the use of Virtual Reality tools I’d like to present an example of their application from our daily routine as an engineering service company.

Fig.1: Engine room as described in the example

It is Friday, eight o’clock. We are in Bremerhaven. A meeting has been arranged to discuss the current state of construction. Supervisors, shipyard representatives, and the responsible engineers of ORCA are already sitting in the conference room. Displayed on the walls around are plans to illustrate the different problem areas of the engine room. After general topics have been discussed, the room is darkened a little and a beamer is projecting the desktop of a 3000-
Euro work station on to the wall. One of the designing engineers starts a special Virtual Reality application to select the model of the current state of design. Slowly moving towards the accumulation of steelwork, pipes and various machinery units pictured now, he dives into the virtual engine room right through two frames. After a few metres of way, the shipyard’s project manager notices that the filter box for the cross-over line is still protruding too far into the catwalk. A short briefing follows and it is decided to utilize a filter of different type, while the responsible designer notes down this modification which is going to be translated into the CAD system the same afternoon. Now, the upper deck is selected to talk about the positioning of the air conditioning unit control cabinet. But, before, one of the supervisors addresses to the meeting to have a look at a main engine bedplate first which has been a matter of long discussions. Now accessibility to the manhole of the lubricating oil circulation tank is made sure in this place as he states, and all those present can continue on their way up into upper deck. Underway, their conversation brings up the question about what should be contained in the virtual documentation of the provision reefer plant. Definitions for this are to be given with much more details to recall them later on though the geometry of the virtual model from a data base. With all items at issue settled at about twelve o’clock, another meeting is scheduled to be held in a hotel nearby the shipyard a fortnight later on the occasion of keel laying. Then the owner, too, would like to get a clear idea of his ship’s engine room.

This example shows the possibilities that can be materialized with moderate financial expense. The virtual model proves to be an essential tool particularly for in-house quality control and for discussion of layout plans with the customers. Its contribution to easier communication between all parties concerned is decisive which makes for reduced cycles of planning.

2. Comparison between Plastic Model and Virtual Reality Model

The comparison with the plastic model shows the differences to traditional working methods even more clearly, Table I.

<table>
<thead>
<tr>
<th></th>
<th>Plastic model</th>
<th>VR model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost of model making</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Time of model making</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Cost of updating</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>General overview</td>
<td>+</td>
<td>–</td>
</tr>
<tr>
<td>Cost of use</td>
<td>+</td>
<td>–</td>
</tr>
<tr>
<td>Mobility of the model</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Duplicating possibility</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Need of space</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Accuracy</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Future possibilities of use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Marketing</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Documentation</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Alternative solutions on display</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Accessibility of all areas</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Supply of component-related information</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Integration of simulated motions</td>
<td>–</td>
<td>+</td>
</tr>
<tr>
<td>Aids for guidance</td>
<td>–</td>
<td>+</td>
</tr>
</tbody>
</table>

- Cost and time of model making, actualisation
Since the VR model is developed from three-dimensional CAD data, there is only little cost for its making and actualisation. The necessary steps to produce the model are export from the CAD
system, conversion into surface-oriented geometry, then import into the VR software. This process is largely automated.

- General overview
  In this field the plastic model has proven advantageous, as it is easier for the eye of the viewer to keep control of things when he/she is moving by him/herself around or through a real model. In a discussion based on a virtual model there is usually only one person to control movements through the model so that all other “passive” viewers will experience difficulties in finding their way. However, various aids exist for remedy, e.g. the overlaying of the ‘you are here’ position into a general arrangement plan.

- Cost of use
  The expenditure to use a plastic model is unbeatably low. The viewer is only required to set out to the model and open his eyes. This is more difficult in case of the VR model due to its dependence on efficient hardware and software. But almost every computer on a designer’s work station is meanwhile fast enough for such purpose and can be fitted with adequate software available at no cost for some part.

- Mobility of the model and duplicating possibility
  To be really present in one place only is a big disadvantage of the plastic model whenever there is need to have it available in different locations. A file on the other hand can be copied as often as desired for distribution to different persons with no delay.

- Need of Space
  Dependent on its scale, the plastic model may require a lot of space. For a VR model the required space is minimized down to the size of an ordinary computer working station. But special projection technologies (beamer with screen, power wall, cave) can even make the display of the VR model reach dimensions to fill a room.

- Accuracy
  The VR model is able to bring about very high accuracy on account of the exact data model in the CAD system. During the process of conversion various parameters can take effect on accuracy to make the VR model suit any specific requirement.

- Future possibilities of use
  - Training
    The model obtained from the constructional data can be used to instruct operating staff at an early stage and at very low cost. Here, mobility is the most important argument against the plastic model. However, it might be that training on the plastic model could be more easily remembered by those involved.
  - Marketing
    Particularly the sale of technically sophisticated products in small series can be supported by a virtual model. For enhanced quality of presentation it is possible to give the design model an additional upgrade with textures and motions of components. Again, the mobility of the VR model is an essential factor.
  - Documentation
    The virtual model allows to attach background information: The multiple contents of a documentation can be opened before the viewer more easily through virtual three-dimensional environments, both onboard and onshore.

- Alternative solutions on display
  Change-over type geometries can provide for alternatives to be very easily made available in a virtual model. In a real model this cannot be achieved without a certain amount of inconvenience.

- Accessibility of all areas
  In a virtual model, the viewer can explore any location desired. In a plastic model this is possible only by using special cameras.

- Supply of component-related information
  Background information can be attached to the geometry in the virtual model. Such information can be recalled simply by the touch of a button. At present, however, the linking of information is involved with considerable amount of work which again cancels the advantage mentioned before in comparison to the plastic model.
- Integration of simulated motions
  Motions of individual components can be integrated and controlled in a virtual model. This is a useful feature, e.g. when investigations for installations are concerned.
- Aids for guidance
  The viewer can more easily find his way in a plastic model. See above “General overview”.

3. Basic requirements for a VR model

The most important basis for the making of a virtual model are three-dimensional CAD data. Through a conversion process these data are processed for use in the virtual reality software. Independence inherent to such software has benefits, but also disadvantages. One of the major benefits is the ability of the VR software to cope with a considerably larger amount of data at a time as compared with similar tools in CAD systems while significantly higher quality of presentation is achieved. Besides, it is possible to collect three-dimensional data of most varied sources, as for reading and conversion purposes special software is used tailor-made exactly for these functions. The disadvantage is that this process is a one-way street. There is no chance to solve problems in the VR model. All problems must be noted down to be handled in the CAD system subsequently. Any design change must be transferred into the VR model before viewing. Consequently the VR model is not available right away at any time.

4. Cost for supply of model

It has been possible by now to cut down cost for preparation of the CAD data to an extreme minimum. But ‘availability at the touch of button’ will not be realized soon. Three hours, however, for the making of a complete engine room model with about 20 systems are a time requirement of manageable size. Single systems can then be actualized within considerable shorter periods. As to hardware it is meanwhile possible to utilise any kind of efficient work station. An all-inclusive VR software package for windows environments is presently available at approximately 25 000 €. Today plain-type viewing units of low functional range can be provided at no cost at all.

5. Ways of application

The 3-d model produced during the stage of design engineering can be used for a variety of applications. For an engineering service company as we are this is an obvious and interesting idea. The model has to be made to suit the requirements of each specific target. But its geometry is appropriate for re-use again and again.

Fig.2: Several ways of application
- Marketing
If you have a product and you don’t know its buyer from the very beginning, marketing seems easier by way of a virtual model. Therefore, the geometry obtained in design engineering can be given certain textures to build up an impression very close to reality. In addition, one can integrate motions of components and alternative solutions of construction.

- Documentation
To more than 80% of the human perception is by vision. So it is an obvious choice to offer a 3-d model as navigating environments for complex information. If you search for information in a documentation you usually start at a machinery unit or system (‘search operating instructions for waste water treatment plant’). Logically your way goes to an information via selecting a geometry followed by the selection of an associated source.

- Training
For the training of the crew, a VR environment can already be used before completion of an installation. As a result, commissioning is often possible at a considerably earlier date through better trained operators. The training model for such purpose can be based on the model from the design stage along with the information contained in the documentation.

- Operation
All functionalities of the documentation model could be expanded to cover the particulars of the systems in operation.

6. Technologies of presentation, their cost and benefit in various applications

The fundamental question “What profit will justify what expense?” leads us to the following consideration in respect of the instruments that can be used for the presentation of a VR model.

<table>
<thead>
<tr>
<th>Computer screen</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of visual immersion in %:</td>
<td>0</td>
</tr>
<tr>
<td>Cost of purchase in €:</td>
<td>0</td>
</tr>
<tr>
<td>Space required in cu.metres:</td>
<td>0</td>
</tr>
<tr>
<td>Hardware/software requirements:</td>
<td>PC, VR software</td>
</tr>
<tr>
<td>Example of application:</td>
<td>Quality control in construction</td>
</tr>
<tr>
<td>Benefit:</td>
<td>Easy look into the state of design/construction without knowledge of using a CAD system and need for a valid licence</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Beamer (as used in the example)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of visual immersion in %:</td>
<td>20</td>
</tr>
<tr>
<td>Cost of purchase in €:</td>
<td>0</td>
</tr>
<tr>
<td>Space required in cu.metres:</td>
<td>4-6</td>
</tr>
<tr>
<td>Hardware/software requirements:</td>
<td>PC, beamer, screen, VR software</td>
</tr>
<tr>
<td>Example of application:</td>
<td>Discussions with several participants</td>
</tr>
<tr>
<td>Benefit:</td>
<td>Easy access to the model at high mobility. Particularly suitable for discussions with several persons</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stereo projection</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of visual immersion in %:</td>
<td>50</td>
</tr>
<tr>
<td>Cost of purchase in €:</td>
<td>200 - 25.000</td>
</tr>
<tr>
<td>Space required in cu.metres:</td>
<td>0-6</td>
</tr>
<tr>
<td>Hardware/software requirements:</td>
<td>1 PC with tube screen and/or pair of beamers with specialised screen and converter; stereo-quality VR software</td>
</tr>
<tr>
<td>Example of application:</td>
<td>Presentation for owners/operators in shipbuilding industry; product presentation on trade fairs</td>
</tr>
<tr>
<td>Benefit:</td>
<td>Planning stages become “easy to grasp” even for the non-specialists and are kept in lasting memory</td>
</tr>
</tbody>
</table>
**Power Wall**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of visual immersion in %:</td>
<td>75</td>
</tr>
<tr>
<td>Cost of purchase in €:</td>
<td>50.000 - 100.000</td>
</tr>
<tr>
<td>Space required in cu. metres:</td>
<td>20 - 40</td>
</tr>
<tr>
<td>Hardware/software requirements:</td>
<td>1 - 2 PC’s, 2 - 4 beamers, large-size re-projection type screen, converter, stereo-/cluster-quality VR software</td>
</tr>
<tr>
<td>Example of application:</td>
<td>Design studies in automotive industry</td>
</tr>
<tr>
<td>Benefit:</td>
<td>Enables presentation very close to reality, notably regarding ratios of dimensions e.g. in case of a motor car. No waste of real prototypes.</td>
</tr>
</tbody>
</table>

**L-shaped projection**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of visual immersion in %:</td>
<td>85</td>
</tr>
<tr>
<td>Cost of purchase in €:</td>
<td>100.000 – 175.000</td>
</tr>
<tr>
<td>Space required in cu. metres:</td>
<td>6 - 15</td>
</tr>
<tr>
<td>Hardware/software requirements:</td>
<td>2 – 4 beamers, 2 screens, 1 – 2 PC’s, 1 – 2 converters, stereo-/cluster-quality VR-software</td>
</tr>
<tr>
<td>Example of application:</td>
<td>Design studies in automotive industry; installation analyses in aircraft and space industry</td>
</tr>
<tr>
<td>Benefit:</td>
<td>Three-dimensional conditions to be easier understandable</td>
</tr>
</tbody>
</table>

**Cave™ 4-5 sides as projection surfaces**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree of visual immersion in %:</td>
<td>90 – 100</td>
</tr>
<tr>
<td>Cost of purchase in €:</td>
<td>400.000 – 500.000</td>
</tr>
<tr>
<td>Space required in cu. metres:</td>
<td>50 - 80</td>
</tr>
<tr>
<td>Hardware/software requirements:</td>
<td>8 – 10 beamers, 4 – 5 screens, 4 – 5 converters, stereo-/cluster-quality VR-software</td>
</tr>
<tr>
<td>Example of application:</td>
<td>Design studies in automotive industry; installation analyses in aircraft and space industry</td>
</tr>
<tr>
<td>Benefit:</td>
<td>Enables preview to a product very close to reality. Even non-specialists will get clear idea of sophisticated structures</td>
</tr>
</tbody>
</table>

This comparison is on the understanding that a PC with VR software as well as a beamer with one projection surface are at hand. The cost for the presentation of a model using these instruments are therefore defined at zero. Presentation with stereo projection can be done either on computer screen or with two beamers in form of an active or passive stereo projection. So in respect of cost and required space there exists a wide range.

A power wall is generally a combination of two projection surfaces at a size of about 5 m long and 2.5 m high. These dimensions are guided by the size of a motor car considering that this instrument is frequently used for scale 1:1 design studies in automotive industry. A power wall can be run with or without stereo projection.

7. Summary

The financial threshold against the use of virtual reality has fallen down to an extent today to allow utilisation of many benefits of this technology at rather low cost. All what is left to do for building up the structures of each specific user is to put together the necessary elements of hardware, software and data information. But an overall solution to be generally recommended does not exist here. As we see it at the moment, Virtual Reality at the touch of pushbutton, especially in shipbuilding industry, remains to be a vision – but maybe within a PDM system this vision might come true in the future.
The Development of a Training Software to Reduce the Risk of Wake from Fast Ships

Trevor J.T. Whittaker, Björn Elsässer, Queen's University, Belfast/UK, t.whittaker@qub.ac.uk
Christopher Cain, Stenaline Ltd., Stranraer/UK, chris.cain@stenaline.com

Abstract

Wash produced by fast ships can pose a significant risk to the users of the coastal region unless the routes and operational procedures are optimised. Consequently it is essential that ferry operators and their staff understand the nature of the potential risk and the measures which can be applied to reduce and even eliminate the problem. With this in mind, Stenaline Scotland commissioned the Fast Ship Wash Group at Queen's University to develop an interactive computer based training package to instruct naval officers on the nature and potential risks associated with the operation of fast ships in restricted coastal waters.

The training software, which has been developed, is divided into four sections, basic wave theory, waves produced by ships, historic problems and solutions for route optimisation and risk assessment. Wave theory and ship wash is a complex subject so the basic concepts have been explained using computer graphics to visually show the main processes. In particular the differences between the wash characteristics of slow conventional ships and high speed craft operating in shallow water are shown so that the user can understand how large period energetic waves are generated which surge and suddenly break on shorelines. The primary risks to coastal users both on land and on sea are presented and related to the wash pattern and the operation of the ship. Reduction and elimination of problems is illustrated by presenting a series of case histories of route and operational procedure optimisation using Stena Voyager, a HSS 1500 operating between Scotland and Northern Ireland, as an example. Finally an example of a risk assessment procedure is outlined which can form a significant part of the documentation needed to obtain a permit to operate in UK waters.

The rationale behind the software development is discussed with particular emphasis on how such training can improve the safety of operation of these types of ship with respect to wash. It is argued that by understanding the nature of the problem captains can avoid potentially dangerous situations particularly when they are forced to deviate from approved courses as a result of unforeseen circumstances. Consequently the training software has and will continue to make a significant contribution to the safety of users of the coastal zone when fast ships are operating in the area.

1. Introduction

During recent years the rapid growth in numbers of fast ferries has often produced problems for operators and the public close to fast ferry routes. People are at risk of being trapped against sea walls and being knocked off their feet as large waves arrived unexpectedly. Small craft are at risk from capsizing, or grounding on submerged banks, in the troughs of the very long period waves produced. Larger vessels at exposed piers can surge and break mooring lines, occasionally with damage to the dockside. The designers of fast ships may have been aware of all the effects their vessels would generate, however, the operators of such vessels often had limited understanding of the effects created when large fast vessels travel at high speed particularly near to land in shallow coastal waters. Although the application of the results of academic studies has offered solutions to some of the problems created the opportunities for improved safety from a better understanding are not necessarily available to all fast ferry operators. It should also be noted that there is no statutory requirement for this subject to be learned by operators.
2. Early Experiences Of Fast Ship Operation

The early experiences of operating a large fast ship can be illustrated by using Stena Voyager as an example. On 21st July 1996 the Stena Voyager, a HSS 1500 started service on the Stranraer to Belfast route. The vessel is 120m long, 40m beam, 4.80m draft at a displacement of around 4,000T laden and is capable of travelling at 40 knots in water depths as little as 10m. Three months earlier her sister vessel the Stena Explorer had commenced service on the Holyhead - Dun Laoghaire route and on both routes it was found that the wake effects being experienced by others at sea and around the shorelines were unacceptable. The original operational procedures closely followed the experience of the operators, gained whilst operating other smaller fast craft, as well as the limited experience gained at that time with the Stena Explorer.

The HSS's were much larger and heavier than other fast ferries operating at that time. Some time earlier, Seacat, a wave piercing 74m Incat, had started service from Belfast. Seacat’s difficulties in Belfast Lough, and how they had been tackled, had been studied by the operational crew before taking delivery of the Stena Voyager. However, a full understanding of the origins of the problems was still lacking. Seacat operated a voluntary system of increasing clearance distances off problem areas thus reducing complaints. Reduction in speed was a decision resisted until it was the only option. The operators of Stena Voyager, all had previous experience of operating smaller fast craft, but had little understanding of the problems that were to be caused by a much larger vessel.

Reviewing the mistakes made at the start of service illustrates how little was understood of the creation of wash/wake effects at that time. For example, when approaching Belfast with a tanker proceeding up the dredged channel into Belfast ahead of the Stena Voyager, a reduction of speed was decided upon. On that occasion speed was reduced to 20 knots to maintain adequate clearance from the tanker. Unfortunately this produced dramatic waves on the beach at a nearby yacht club. If a specific speed had been required to make large waves then reducing to around 20knots would have been considered ideal, because it coincided with the critical values of depth Froude number and the length Froude number.

Complaints were being received along the north and south shorelines of Belfast Lough and also at P&O’s Cairnryan berth in Loch Ryan. As a response to these problems, speeds close to the coast were reduced with the immediate target of preventing any further complaints and eliminating the potential risk to others. The passage times however were extended unfavourably and this was considered likely to effect the commercial success of the service. The self-imposed speed restrictions had halted the complaints but the need for a study to find reasons for the unacceptable wake was apparent. Studies were commissioned initially with SSPA of Gothenburg and then later with Queen’s University Belfast. These studies showed that it was possible to produce effective operating procedures that minimised the risks associated with wake. Work was also commissioned by The Maritime and Coastguard Agency, (MCA) to study the problem and reports MCA 420, MCA (1998), and MCA 457, MCA (2001), were commissioned.

3. Requirement For Training In Wake

It was soon identified that this process was to be an ongoing development of procedures as knowledge of the subject became more and more detailed. At the same time it was noted that the masters and navigators were keen to understand the theory behind the varying approaches to resolve the wake problems. They had no previous education regarding this subject and frequently asked the same questions. These questions were mainly directed to the senior master, who was the point of contact for the studies being undertaken. Careful explanation was required and this was a time consuming process. In addition when new officers were appointed to the vessel, this process had to be repeated with each new recruit starting at the most basic level. At this time it was noted that other companies and personnel operating fast ferries for the first time were repeating the same wake mistakes which had been made earlier by the established fast ferry operators. This was felt...
to be arising from a high use of seasonal staff by those companies, combined with training regimes that were not fully effective

Cain (2000) was used as part of the earlier attempts to train both the Masters and Navigators of the Stena Voyager. As an educational tool this was not sufficiently comprehensive, as repeatedly the same questions were raised from those reading it.

Therefore there was a need to develop an effective training package that could explain both the basic theory, and it’s application through the operational procedures developed. In addition it was thought that such a training package would help minimise the risks associated with wake, by ensuring what had been learned from the studies so far was fully understood by all. It was felt that an adequate training package would additionally be of value in fulfilling the requirements within the High Speed Craft (HSC) Code Chapter 18 ‘Operational Requirements’. Although wake risk prevention is not specifically mentioned within Chapter 18, provision is made for the authority to revoke the ' Permit to Operate', if safety is not maintained. This permit is only to be issued, if amongst others "[...] crew qualifications and training, including competence to the particular type of craft and service intended, and their instructions in regard to safe operational procedures [...]” are satisfied, HSC (1995), paragraph 18.1.3. and specifically 18.1.3.7). In addition it was identified that an adequate training package could assist in meeting the requirements of the MCA. The MCA requires under the HSC Code Chapter 1.9.3 that all fast craft should produce a ‘risk assessment of passage plan with respect to wake. Clearly such risk assessments can be produced without the masters and navigators fully understanding the content. However, lack of understanding can lead to problems if it is necessary to deviate from the passage plan. The risk assessment passage plan is considered a “live” document by the MCA and can be the subject of review and modification in the light of complaints received. This document should be updated in the light of experiences gained and any alterations to risks identified. To produce the desired training package Stena approached the 'Fast Ship Wash' group at Queen's University Belfast.

4. Experience with Computer Based Training at Stena

The use of computer based training (CBT) had been successfully implemented for training the crew in the operation of the marine evacuation system (MES). Ability to train/educate ship’s staff is not a skill that is automatically gained with rank or responsibility, and it had been seen that better results were achieved when CBT was used rather than relying on ship’s officers to deliver training courses. Captains and ship’s officers were seen as ‘authority figures’, and junior staff are often reluctant to ask for clarification of points they did not fully grasp. For this training package the target group was of widely varying educational capabilities, and consequently the CBT package was specifically designed to train those with low capabilities as well as those more gifted. Individual learning ability, determines the rate at which people progress with the fully interactive CD. This package was seen as a positive learning experience by all levels within the crew with mistakes being only seen by the computer instead of the rest of the crew.

5. Design Brief for the Wake Training Package

At the outset it was decided that the wake training package would be based on slightly different criteria to the MES training package. The wake training package is designed to form a basic educational source that could be referred back to as an authoritative reference as well as providing a positive learning experience with repeated confirmation of what had been learnt by the trainees.

The target group for the wake training package was deemed to include not only the Masters and Navigators of fast craft but also those involved in higher levels of marine operations who may have restricted knowledge of this developing subject. Many ex seafarers in higher levels of marine operations often have no experience of fast craft operations at a practical level, but they are still responsible for the implementation of safe marine operations within the International Safety Management (ISM) code.
Although the target group has a reasonably high level of education only the naval architects have studied ship hydrodynamics and basic wave theory. Consequently the main requirement was to present a highly technical and difficult subject in a way that could be understood by the non-specialist. Therefore it was necessary to start from a basic level and proceed to a level of the latest international knowledge and experience of the subject. It was also considered to be essential that there should be a strong emphasis on illustrating the lessons learned from past mistakes and the solutions implemented by applying the evolving knowledge of the subject.

6. The Wake Training Package

6.1. Overview

Although there are specific computer packages for developing and presenting interactive training with highly developed graphical user interfaces, they are very specialised. This training package however was considered to be a live document as it was anticipated that the training package would need ongoing development as knowledge of the subject increased in future years. Consequently it was decided to use the commonly available package, Microsoft Powerpoint®. However, it was necessary to utilise all the more advanced features of the software in order to produce a user interface which was both attractive and obvious to use. The appearance of each screen is shown in Fig.1. This is one of the introductory screens and explains how the user navigates throughout the training package. Buttons are provided to navigate backwards and forwards from screen to screen as well as information buttons which bring up either pictures, graphics or video clips to illustrate particular points being made in the text. In addition it is possible to jump between sections using the buttons in the bottom left hand corner of the screen.

![Using The Navigation Bar](image)

Fig.1

6.2. Contents of the training package

The training package has been divided into four main technical sections as follows, which are preceded by the introductory screens.

- Basic wave theory
- Waves produced by ships
- History – problems and solutions
- Risk management
Introductory screens

The introductory screens include a welcome page, acknowledgements and explain the use of the navigation bar, Fig.1. In addition the requirements for training, the learning objectives and the contents of the package are presented. The screen showing the requirement for training states that:

- unlike the majority of slow conventional ships, high speed ships operating in confined coastal waters produce unnatural energetic long period swell waves which arrive at the shoreline unnoticed and break unexpectedly causing risk to coastal users
- risk from wash waves produced by the increased numbers of high speed ships in recent years has raised public concern and the problem must be managed
- a basic understanding of wash waves is essential for ship operators to avoid or reduce potential risk and therefore training is essential

The knowledge the trainee should have gained by the end of the training session has been summarised as:

- the characteristics of gravity water waves
- how waves are transformed by sea bed topography
- the influence structures have on wave patterns
- the nature of wash waves created by fast ships
- how fast ship wash differs from conventional ship wash
- how vessel operation effects the wash waves produced
- how risk from wash can be avoided or reduced
- risk management procedures.

Having completed the course introduction the trainee can choose the next section to study but if a choice is not made one automatically progresses to the section on basic wave theory. Each section has a specific set of learning objectives which are clearly stated at the beginning and end.

Basic wave theory

Before trying to understand the three dimensional wave patterns produced by moving objects such as ships it is essential to study basic wave theory. This subject is often taught in coastal engineering courses at both undergraduate and postgraduate level. It is a difficult subject and would normally require 8 hours of lecturing for students to gain a basic understanding. In the training package the absolute minimum of technical material is presented and extensive use has been made of computer graphics to visualise the physical processes taking place. A description of the key variables defining a water wave is followed by details of the fluid particle motion. Computer graphics show the fluid particle motion in deep, intermediate and shallow water; the latter being the most significant for fast ship wash in coastal waters. The wave transformation processes of shoaling, refraction and diffraction are presented, again illustrated with graphics. Shoaling and refraction are the two most significant transformation processes with respect to

Finally the concepts of dispersive and non dispersive waves are introduced. This is the most significant aspect as it causes the primary difference between high speed and conventional slow speed ship wash at the shoreline. In deep water waves, which are those where the wave length is greater than twice the depth, energy is dispersed from wave to wave in the group. Consequently one wave quickly becomes two and then three due to the longitudinal spread or dispersion of energy. As the number of waves increases the energy and hence height of individual waves decreases with time. However, in shallow water, where the wavelength is twenty or more times the depth, the energy in a single wave is conserved and the single wave remains without reduction of energy. The very long period leading waves in fast ship wash largely behave like this.

Waves produced by ships

This is the second and most relevant theoretical section and builds on the knowledge gained about basic wave theory. It comprises 16 main screens with a further 33 accessed by information buttons. The content is described by the opening screen and is presented in Fig.2.
Fig. 2

This section describes waves created by ships. It explains how the speed of the ship and the water depth effects the characteristics of the wash waves produced.

At the end of this section you should understand the following:

- Wash wave patterns
- Length and depth Froude number
- Supercritical, critical and sub-critical wash patterns
- Differences between ship waves and naturally occurring waves
- Importance of ship speed when travelling in different water depths
- Potential risks in the coastal environment

The section commences with a brief historic review of the work of Lord Kelvin, William Froude and others who developed the basic theory of ship wash. The 'Kelvin' wash pattern is presented and the concepts of length and depth Froude number is discussed. Naval architects normally use length Froude number, which is proportional to the ratio of ship speed and the square root of waterline length. Conversely the depth Froude number, which is primarily used to study wash waves in the coastal environment, is defined as the ratio of ship velocity and the square root of water depth. Its physical interpretation is the ratio of ship speed to the maximum velocity a wave can travel in a given depth of water. It is this wave speed / depth limitation, which determines the characteristics of high speed ship wash in shallow coastal waters.

Screen 5 and the associated information screens provides an overview of the different types of wash pattern produced in various depth Froude number ranges and illustrates the points with a series of photographs diagrams and computer animations. Fig.3 shows an example set of aerial photographs of a fast ferry operating at sub-critical, critical and super-critical depth Froude numbers.

Fig.3: Pictures of Sub-critical, Critical and supercritical wash

Unlike slow conventional ships fast ships can operate in the super-critical and critical depth Froude number ranges and it is the different characteristics of these wave patterns which causes different risk to coastal zone users. Consequently the training package concentrates on these wave patterns, produced by high speed ships, and draws the distinction between them and other wave systems such as slow speed sub-critical wash, as produced by many ships, and wind seas. In this context high speed ships have been defined as those capable of exceeding a depth Froude number of 0.85 rather than the 'High Speed Craft Code' definition.
A typical wash wave time trace measured several kilometres from the track of Stena Voyager operating within the super-critical depth Froude number range is presented, as shown in Fig.4.

The time series is divided into zones and each is studied in detail. Additional technical information can be obtained at each stage and examples of the wake close to the ship and further away at the shoreline can be viewed. It is emphasised that it is the very long period waves in the first zone and the very short steep waves in the third zone which are peculiar to fast ships and that slow conventional ship wash is similar to zone 2. A comparison is then made between the very long period but relatively small amplitude waves in the leading part of the wash and the much higher and shorter wind induced waves produced during storms. The trainee is then reminded of the concepts of shallow water waves, non dispersive waves and shoaling, covered in the wave theory section and told that the leading waves in the super-critical wash behave like this.

![Fig.4: Super-critical wash trace showing zones 1, 2 and 3](image)

Having covered super-critical wash attention is turned to the critical and near-critical operational zone. Here the ship speed and depth limited wave speed are the same or similar. The effect on wash pattern is analogous to an aircraft hitting the sound barrier. The photographs, graphics and output from numerical models show that the largest and most energetic waves are produced in this
operational region. Consequently this region should be avoided. In practice it can not, simply because as a ship leaves port, accelerates to cruising speed and then heads for deeper water it will transcend the critical zone twice. It is explained that ships should avoid the near critical zone. A depth speed curve is presented which clearly shows the critical range of related depths and speeds, Fig.5.

However, there are circumstances when ships can become trapped in the near critical zone without sufficient power. Inadvertently generating a large near critical wash is one of the most common situations leading to wash problems. The situation where a heavily loaded fast ship possibly with the additional drag of hull fouling is entering an estuary from deep water and is down on speed is illustrated. If the ship is travelling at speed lower than the speed at which wave drag is a maximum for a given depth, resistance will rise as the water depth reduces. Thus speed will drop unless additional power is available. Consequently unless speed is reduced so that depth Froude numbers of less that 0.85 prevail, the ship can travel for a considerable distance at near critical speed dropping in speed with reducing depth and producing large transverse non dispersive waves which grow in height and spread in width. The training program instructs the trainee to recognise this situation by using the depth speed curve and to take the appropriate action.

In the final part of this section risk tables are presented in which different potential risks to coastal users in a variety of situations are related to the various zones in the wake time series. The screen shown in Fig.6 summarises the cause of the main risks posed by fast ship wash.

![Image of section 2: Waves Produced by Ships](image)

**Fig.6**

**History – problems and solutions**

The third section in the training package describes a series of case histories relating to the operation of Stena Voyager on the Belfast to Stranraer route. A range of studies have been carried out at different locations to optimise the route and operational procedure used by Stena Voyager with the aim of minimising wake problems. The work has included numerical modelling of the wash transformation processes from the ship to the shore and a substantial number of physical wake measurements at critical locations around the shoreline which bounds the route. A substantial number of the scenarios mentioned in the risk tables have been experienced during the lifetime of operation on this route. Consequently excellent training examples of what can happen, the lessons learned and the procedures implemented have application to other ships and locations. In particular the trainee can see the application of the technical aspects of the subject, which were presented in the previous sections.

Six case studies are presented highlighting different aspects of risk associated with fast ship wash and the measures adopted to manage the risk.
• **Inner half of Belfast Lough**
  This is a very shallow area with mud flats and sandbanks exposed at low tide. A dredged channel enables ships to enter the port. High speed operation results in inundation of banks and breaking waves on the shallow areas at mid to low tide causing risk to bait diggers and small boats passing over the shallows. During the top part of the tide waves surge and break on slipways and there is grave risk of people on beaches being trapped against the sea walls behind. At high tide the sea walls are overtopped in several places and coastal paths and nearby gardens flooded. In this instance the only solution has been to limit speed to 17 knots and keep well within the sub-critical range.

• **Ballyholme Bay (outer southern shore of Belfast Lough)**
  This is a horse shoe shaped bay close to the entrance of the Lough. The bed slope is very gentle and the beach is covered at high tide with water up to the sea wall in front of the promenade. The potential of trapping people against the sea wall during the top two hours of the tide and overtopping at high tide were the main risks. The problems were resolved by changing the course of the ship approaching the Lough so that the leading super-critical waves dissipated their energy on the eastern rocky headland which then sheltered the beach. As an additional precaution the ship slows at the 30m contour between one hour either side of high tide.

• **Killroot coal jetty (outer northern shore of Belfast Lough)**
  12,000T ships discharge coal to a local power station at an open jetty located on the northern shore of the outer part of the Lough. A conveyer discharges the coal from the hold into a hopper on the quay. The ship is moored with fore and aft lines, brest lines and springs plus a short midship line to prevent the ship ranging on the berth and the coal missing the hopper. The long period leading waves in the wash causes the ship to range several metres breaking the short lines which end up taking most of the strain. At present the problem has been resolved by slowing when the jetty is 70° off the bow during the inbound passage and delaying acceleration on the outbound passage until after the 70° bearing. An additional precaution has been the fast ship warning the moored ship of its approach. However, these procedures are only applied when the berth is occupied. Otherwise high speed operation continues to the fairway buoy and acceleration starts at buoys 3-4 on the outbound passage.

• **Beaches at Crawfordsburn and Helens Bay**
  Occasional large wash events were reported. This was identified as being caused by prolonged near critical speed operation in the outer part of the Lough for the reasons discussed in a previous section. The problem was resolved by dropping speed when the critical speed trap was encountered.

• **Loch Ryan inner part**
  This is a similar situation to the inner part of Belfast Lough except the channel is 4m shallower and narrower. An open Ro-Ro ferry terminal is located at the seaward end of the channel. Even with the fast ships travelling at sub-critical speeds the link spans at the terminal were excited by the short period but steep waves in the Kelvin wash. This could only be resolved by very low speed operation of 12 knots.

• **Loch Ryan (outer part)**
  This case study demonstrates the use of speed steps, wave focusing on the inside of turns and seasonal variations in ship operation to accommodate nesting of 'Little Terns' on low lying pebble banks. In Loch Ryan there is a deeper trench to seaward of Cairn Point. This can be used in the deceleration phase to transcend the critical zone very rapidly by dropping speed simultaneously with a sudden increase in depth. Wave focusing inside the turn into the loch is shown which can cause problems in some areas. However, in Loch Ryan the wave is concentrated on a remote rocky headland. During May early June a colony of Little Terns nest on a bank which is normally over-washed at high tide by the leading waves in the super-critical wash travelling down the loch. Early slow down on the few sailing effected avoids the problem and shows how pragmatic speed restrictions at certain times can avoid a blanket speed restriction.

The most important outcome from this section is that it shows what can be done to manage the consequences of fast ship wash. Also it clearly demonstrates that single measures do not resolve the
problems in all cases. Consequently it is emphasised that each part of every high speed ship route close to land must be assessed individually and the most appropriate operational procedure adopted. Often it is possible to vary the procedure to account for tide level, the proximity of other ships and even the nesting times of wildlife.

**Risk management**
In the final section the trainee is presented with a list of measures which can be adopted to manage the potential risk from high speed ship wash as well as the methods of risk assessment and the production of passage plans. At present there is not a statutory standard for preparing the documentation required for the route assessment with respect to wake wash as part of the paperwork required for the permit to operate which must be obtained from the MCA. The screen shown in Fig.7 explains the current position.

![Fig.7: Road to Legislation](image)

In some countries in the world simple formulae are applied to determine the maximum height of wash wave that is permitted at a certain water depth off the shoreline. Unfortunately this tends to be over simplistic and it can be argued that it does not necessarily prevent risk in many cases. In the UK the risk assessment document is live and incidents or complaints about wake are logged. The offenders are required to reconsider their route assessment document and if deemed necessary alter their operational procedure. Consequently the trainee is left with the clear picture that if he or she is operating a fast ship it is their responsibility to ensure that the potential for wake incidents is minimised particularly when the ship is forced to deviate from the specified operational procedure due to unplanned circumstances. This can be as simple as avoiding other shipping on the planned route and for example the consequences of course changes and the resulting wave focusing on the inside of turns needs to be understood. A further example is the potential consequences of passing other vessels both large and small when accelerating though the critical speed range particularly in channels and estuaries of confined width. The possibility of grounding of large vessels constrained by their draft is a further example.

**7. Consumer Feedback**
An important consideration when producing an effective computer based training package is to seek a critical review of the work produced before the final version is implemented. Ideally the training package should be presented to a selection of the target group it is intended for, with constructive feedback being encouraged. As had been learned with the production of the MES training CD, this was a fundamental step in the process to achieve a successful training package for wake.
During this process the first draft of the CD was distributed throughout Stena Line in the areas where feedback was considered as being assured. Stena Line’s own naval architects in Sweden were sent a copy of the 1st draft as well as senior managers involved with the safety management of Stena Line both in Sweden and the UK. By including these groups it was hoped that the review would have input from as wide a group as possible. Some of these naval architects were involved with the design of the HSS craft, and therefore had detailed knowledge of the craft. The CD was also made freely available to the masters and navigators of the Stena Voyager and critical review requested from all. Adequate time for reviewers to consider the training package was essential. The review criterion for others not involved in the making of the CD was to consider the total concept and the delivery of the training. This process took a little longer than expected but the comments received have been of considerable value and were considered carefully by the development team at Queen’s.

This feedback process was a success and very little was identified as requiring change. The main addition suggested was the inclusion of more video clips of actual wash action. A consistent comment received during this feedback process from all areas was, “[..] this CD should be distributed to all the HSS 1500’s”.

Feedback from the naval architects made very little reference to the basic theory section however they did regard the explanation of how theory could be used to provide the answers to problems as very good. The naval architects were not being shown anything within the theory section that they did not have full knowledge of, however, attempting to teach this theory to either reduce or eliminate problems created, was new to them.

These comments were very positive as they indicated that the target group of masters and navigators as well as the most senior management of the company valued the work achieved so far, and felt it should be shared throughout the company. Consequently after the inclusion of the suggested additions, copies of the CD were sent to both the Stena Explorer and the Stena Discovery. This training package now forming the company standard for all HSS 1500 vessels with respect to wake issues.

8. Potential for Improved Safety as a Result of Training

Reduced risk from passage plan deviations
The package, by raising the understanding of both theory and the solutions to potential problems, provides the best option to help to prevent future incidents where coastal users are placed at risk. As previously discussed, this is particularly important when deviation from the passage plan is forced upon the vessel. Wake problems have occurred due to altering course to avoid other vessels resulting in deviation from the specified course line. Also, unplanned reductions in speed to a critical value have caused problems. After training the masters involved with these incidents readily accept that they have the capability to reduce the risk to others from wake in these situations now that they understand the consequences of the different actions they can take.

Reduction in environmental impact
Knowledge enables the operators of fast ships to act in a responsible manner with respect to the environment. The potential risk from wake to a very important breeding area for terns in Loch Ryan during their breeding season is a notable example. This breeding colony contains different varieties of terns including some rare species. Working with the RSPB and applying the knowledge of wake propagation to the situation, an early reduction in speed at times of higher tides resulted in removing the risks to the terns. There was a small impact on crossing times for the Stena Voyager but this was more than offset by the very positive thanks received from the local RSPB representative. Instead of being viewed as commercial company operating purely for profit the company was seen as being a safe operator who was prepared to make efforts to operate with due regard for environmental issues.

The wider application of training
There is a need to produce a similar training package for all operators and crews of fast ships as a
method of raising awareness throughout the marine industry. This could be extended to the provision of informative material to yacht clubs, boat clubs, sea based leisure organisations and schools.

A recent incident involving a smaller fast ship near Cowes resulted in the Marine Accident Investigation Branch, (MAIB), investigating the causes of this incident. The final report on this incident has not yet been published, however, during their investigations the MAIB became aware that Stena Line had produced their own training CD and requested a copy. This was provided on a ‘commercial in confidence’ basis to them and during a visit to them the reasoning behind the production and format of the training software was explained. The MAIB were interested to know how problems had been identified and addressed to improve safe operating procedures.

Incidents involving wake are mainly preventable by a full understanding of the problems involved, It seems likely that this subject will soon have to be studied by all operators, masters and navigators of vessels capable of creating wake problems. It is not however correct to wait for possible legislation changes to identify and provide a solution to the lack of education for those involved on a day to day basis.

*Measures of success*

As procedures have been developed in the light of the results of the various studies completed there followed a drop in insurance claims from wake issues. Commercial companies are largely driven by an improvement to the balance sheet as well as safety factors. To illustrate the benefits of close cooperation between the operators and the academics and the development of training, *figure 8* shows the wake Insurance Claims made against the claims paid for during the first three years in service of the Stena Voyager. It should be noted that the claims made also include the claims that have not been honoured due to being false.

![Wave Wash Insurance Claims](image)

Fig.8: Insurance claims with respect to wake from Stena Voyager

Much can be achieved by ensuring that the public, are aware of the dangers from the wash of high speed ships. Warning signs are of most benefit to visitors to the area, since the local population are usually experienced with the effects from other ferries and adjust to the new risks. However those involved in marine operations at a high level should be able to understand why and how improvements to reduce risk can be achieved.

The illustration regarding insurance claims reinforces the success of applying the lessons learnt from theory to operational procedures. Increased safety through improved operational procedures can also
be seen to have a positive financial impact. Safety is often thought as a cost alone but by proactive management insurance liabilities can be minimised and costs reduced for both operators and insurers.

9. Concluding Remarks

A computer based training package has been produced on the various aspects of wake produced by fast ships. This is in response to public concern regarding the additional risk resulting from the characteristics of the waves produced. The development of a computer based training system has significantly reduced risk to others and has the potential for wider application. It has already been adopted by all the HSS 1500's operating in Stena Line. Training and information transfer has been shown to have a direct impact on the number of wake related incidents and the resulting insurance claims.

References


Internet-Based Cooperation between Supplier and Shipyards in the Maritime Cooperation Network

Martin-Christoph Wanner, Matthias Neumann, Björn Weidemann
Fraunhofer Application Centre, Rostock/Germany, mcw@hro.ipa.fraunhofer.de

Abstract

Cooperation or collaboration is a very important item in the future. Even in periods with low economic growth companies are reducing their costs to keep competitiveness. But not only in stable also in increasing economic growth can enterprises benefit from collaboration processes. Currently and in the future the shipyard is seeking for new forms of product to reduce their costs. These products are system-products with high complexity. To supply this new demand it is necessary to develop new possibilities for the supply industries and to reform the existing processes. Working more efficiently, elected companies of the maritime supply industry, have analysed their processes. They have identified various gaps in the data exchange, information-system and product presentation. To solve these problems it is necessary to implement standards for data exchange, a knowledge database and to set standards for product description. Also the formation of the network is mentioned by the companies as an important fact. The network has to secure efficient cooperation and collaboration processes among their members. It is realised with the framework and the network content, implemented in the Maritime Cooperation Network.

The analysed methods and functionalities, the supported data exchange and the content of the Maritime Cooperation Network will help the maritime supply industry with regard to their daily manufacturing process and to their cooperation processes.

1 Introduction

The vision of Maritime Cooperation Network is the establishment of a network of enterprises able to respond immediately to the demands for multiple services. The Maritime Cooperation Network intensifies cooperation among its participants by offering architecture of services designed to meet the requirements of both manufacturers and customers. It is developed for suppliers.

This architecture of services includes a wide range of measures in order to streamline cooperation, integrating methods like as the provision of knowledge, tools for optimizing communicational processes and the identification of eventual central service holders.

The purpose of the project is:

1. Development of requirements necessary for generating virtual units within the maritime network of cooperation for the system component product “ship”, including the option of applying this concept to transhipment facilities.
2. Creation of an interactive, team-based platform of cooperation with the intention to integrate this platform into national and international structures.

The term virtual unit refers to the constitution of supply chains from a pool of companies, all of which are economically independent but participating in one and the same project, at best interacting with the customer as one entity.

The methods and tools to be created within this network of enterprises are supposed to serve all partners involved, including activities such as negotiating conditions, commissioning and long-term customer care during utilisation time. This platform of cooperation is an essential instrument for the efficient exploitation of the region’s recourses and the development of international trade areas.

For the further development of this architecture of services, the following main focuses have been identified:
Field 1: Fundament of the platform for data-security and capable work within the system
Field 2: Tools to generate system products and to support system component suppliers during preparation and utilisation phase. Creation of an open, expandable e-industrial-business-platform.

Field 3: Development of a framework for conceptual details of the network.

Working closely together with the identified partners, solutions for these three fields will be developed in the course of further efforts to complete the project.

Field 1 refers to the development of the base of each platform; the administration of different users. Furthermore the base has to secure the correct data exchange, capable working within the system and web ability. To ensure the right choice of the base-system, it is necessary to analyse the market. Therefore, the need of the users and the supply of the software-companies have to be harmonised.

Field 2 particularly contains of solutions simplifying the preparation phase of the development of system components and services. Beginning with an identified demand, the products are set up, suppliers for the chain of processes are found, and data to be exchanged is defined in the course of developing a guide for the generation of new/ renewed/ modified system-products. It deals also with the build-up of an internet platform and the implementation of supportive tools for the participants of the network. After contractual membership to the project has been confirmed by its participants, the internet platform will be available and successively enlarged at www.mariconet.de. Later on, the developed solutions shall become part of the union MAO e. V. or its predecessor at www.mao-ev.de. On the basis of the participants’ experiences and demands, additional tools supporting the processes within the network shall be developed.

The search and selection of partners joining the network will also be carried out through the internet platform; the same applies to the presentation of the participants’ specific competences. Common standards for presenting the member companies and their system products or services to the outside will be developed according to the customer’s primary demands.

Setting up the requirements to visualised CAD-files on a Viewer and afterwards implementing the Viewer will be another target.

Field 3 refers to the development of a structure of Maritime Cooperation Network which is accommodated to the intention of realizing the implementation of the described architecture of services. In this process, the definition of role concepts, beside other things dealing with the identification and arrangement of the service holders’ central functions within the network, must be taken into consideration.

For instance, the integration of central service holders providing energy or conversion is necessary starting the first utilisations of the project. Also, one service holder in lead of Maritime Cooperation Network must be determined. The functions must be identified and described in cooperation with the partners. Also, the processes of embedding the service holders into the system component suppliers’ work must be defined.

2 Classification of project into the context of InnoRegio

The Maritime Cooperation Network is one sub-subproject of the project “Baltic-region Maritime Alliance” and it is integrated in the subproject maritime innovation and cooperation network. The “Baltic-region Maritime Alliance” is one of 23 projects of the InnoRegio-programs and it is administered by the Federal Ministry of Education and Research (BMBF) to raise the value and the capacity to complete in the state-aided regions, www.innoregio.de.

The project Maritime Cooperation Network has started in April 2002 and will end in April 2004. The Fraunhofer Application Centre Rostock in cooperation with Fraunhofer- Institute IPA-Manufacturing Engineering and Automation in Stuttgart are developing the Network. The Maritime Cooperation Network shall in close connection with the other sub-subprojects found the mandatory requirements for the development of sustainable cooperation within the Mecklenburg West Pomeranian maritime
industries, www.innoregio.de, but not only this region is targeted. The main focus is on Mecklenburg West Pomeranian but companies from other regions have the same opportunity to join the Maritime Cooperation Network as members.

Fig.1: Classification of Maritime Cooperation Network into the sub-subprojects

3 Maritime cooperation Network

3.1 State of the art of current supply cooperation

An implementation of companies into networks or clusters will be a key role for nationalised and internationalised competitiveness. Current developments of competition are globalisation, dynamic sampling, crosslinking and data maintenance and mass customization mentioned in Steven and Meyer (2000). Grouping the core competence will raise the opportunity to develop new system-products. Cooperation-networks are the groundwork to offer products out of one hand. Schuh et al. (1998) identified the potential of cooperation out of cooperation-networks as an additional value. To secure the efficient cooperation it is necessary to target a collective aim, but the actors do not work solely within the network, Kiesel and Klink (1998).

Currently, component suppliers and service holders work largely uncoordinated; the existence of competitive component suppliers for the final manufactures is given only to a limited extend. So called Supply-Networks, for instance, consisting of a shipyard as the final manufacturer and a wide range of independent component suppliers, are constituted specifically for individual orders and must be steered by the final manufacturer at high cost.

At present, the region’s potential is therefore not satisfactorily exploited. The methods, tools and structures necessary for the creation, handling and coordination of cooperation involving different companies are literally not existent. Furthermore, an integrated platform serving as a source of information, communication and cooperation for the maritime network involving customers, system component suppliers, service holders and final manufacturers is missing.

The functions to be developed within project affect the whole network and are therefore of enormous importance to the realisation of the concept. The focus is on the creation of a general network build-
up, cooperative structures for the generation, planning, handling and accounting of activities involving different companies.

The Maritime Cooperation Network found the mandatory requirements for the development of sustainable cooperation.

3.2 Status of work

Currently the Maritime Cooperation Network is developed. It is an enhanced process in which the maritime supply industry exert a high influence. Therefore, during the start-up-phase, the Maritime Cooperation Network is working very closely with companies which are working in the maritime sector. These companies are working in four different branches and are producing different products. They are situated in the engineering sector, shipping equipment, marine engine and maritime consulting. This assures that a wide range of variety needs from the maritime industry are considered to secure the usage of the Network after the start-up-phase.

The companies have identified their main problems during the cooperation process which should be solved by a network.

The first problem is the undefined data exchange between supplier and supplier and between supplier and customer. This process makes the workflow more inefficient and to solve it, it is necessary to locate and analyse the interfaces during the whole manufacturing-process of system-products. Because of this, the Maritime Cooperation Network has picked up a specific past project and is analysing the data exchanges.

The seeking for qualified partners, the second problem, is divided into two aspects. The first aspect is the admission of qualified partners into the Network. To define the term qualified partner it is essential to analyse the requirements towards suppliers. The second aspect is the search itself. The Maritime Cooperation Network has to implement at least one company of the whole range of the maritime supply industry, to serve the various demands of different suppliers or customers. If it is not possible, the search has to be extended to the surroundings, e.g. platforms which offer products, to secure the efficiently of the supply-search-process. Parallel to the content of the network a supply structure will be developed to navigate easily through information of suppliers. Generating products out of the supply structure is also an aim, to give new impulses towards innovation or vertical product enrichment.

The third problem is linked to the second problem. Additionally to the seeking of new partners is interesting, also the provision of information is important, especially news and links.

Opportunities to view different CAD –files have been identified as problem number four. A viewer visualising different file-formats, which are elected from the companies, has to be implemented. During the cooperation process the CAD-files, exchanged between companies in different CAD – formats, are constantly enhanced up to the end of the detail-design-phase. To give collaborating enterprises the chance to work closer together, the opportunity to mark modifications or constructional fault helps to optimise the engineering-phase.

The last problem, named indirectly, is the absence of project-management-support. Currently a lot of software solutions are available, but the implementations into small enterprises are not economical. To support the cooperation/ collaboration process it is urgent necessary to offer a web-compliant project-management-software.

The project is divided into three equal parts, the basement of the platform, the implementation of methods and software and the structure of the network.
3.2.1 Base of the platform to meet demands of the maritime industry

The base of the platform is a content management system. It was chosen by potential users. The first step was to analyse the needs and requirements of potential users. The ascertained requirements were easy working within the system, secure the safety of data, ability to implement different user-groups, low costs and web compliant integration. After different systems test it is implemented a content management system from active web.

This server is a content management system platform in order to manufacturing high-performance distributed applications which can be adjusted to individual needs at any time. The only tool needed to work with the contentserver is a web browser.

Based on an open information architecture, the system allows users to create and maintain web pages only with the ability to fill out a form. This is achieved by the separation of layout and content, so that the designer will provide the layout and publishers only have to care about the content.

Offering a huge amount of information, its presentation must be individualized to either personal or regional data by introducing specialised user or group profiles. Furthermore not all users are allowed to read, write or delete information so a sophisticated right model secures that information will be addressed to the right areas and target groups.

Another functionality is that users can publish and share documents of all kinds such as technical drawings or CAD-files. All these documents are sorted by topics in catalogues and are subject to the above mentioned right model too.

3.2.2 Implementation of methods and software into capable solutions

The initial point of the process was the increasing demand of system suppliers for new solutions during the cooperation processes. Analysing this demand emerged various areas of possible improvements, simplifications and cost reductions. These improvements were piecewise evaluated concerning their feasibility. The result was a catalogue of possibilities partitioned into the following two categories.

![Fig.2: Phases of manufacturing process](image-url)
Practical methods and functionalities implemented in the Maritime Cooperation Network:
The identified information gaps during the cooperation process will be solved by the functionalities and the methods. To offer the expected support, identified from suppliers, the manufacturing-process has to be analysed. Step one was to depict the manufacturing-process and separate it into typical phases. It was shown that the processes can divided into two phases the initial phase and the operating phase. These parts of the manufacturing-process are shown in Fig. 2.

The support the Maritime Cooperation Network will offer is linked to the analysed phases. Each interface to the next step will be supported by methods or functionalities. The functionalities could be:

- **demand pool**
The demand pool is a tool, which enables system suppliers to identify demands that shipyards have determined.
This pool can be separated into two different parts. The first and more important part is the online solution where the ideas of the shipyards concerning system products are collected and presented in the form of notes and sketches. These sketches become evaluated by professional persons (customers) or Maritime Cooperation Network partners. Finally the ideas are either declined or accepted. In that case the idea will be adopted to the innovation pool.
In contrary, the second part deals with an offline solution. Here the duty of the Maritime Cooperation Network operator is to attend to the requirements of the customers. He is in permanent contact with the customers and thus capable to respond to changes in demand. Furthermore he establishes contact between customers and suppliers.

- **innovation pool**
The innovation pool shall ease and support the communication in the innovation phase. The innovation pool represents a collection of ideas, files and conversation protocols from all parties concerned ordered by different ideas. The originator(s) of the idea is able to distribute login licenses to invite other companies to an exchange of ideas. During the distribution of the login licenses the legal situation between the parties is recorded. The participation at an exchange of ideas is only allowed if the party accepts the contract.

- **tool-supplier search**
The supplier search tool enables the Maritime Cooperation Network enterprises to search faster and consequently cheaper for possible suppliers. In particular this tool assists the enterprises in locating potential strategic partners for the realisation of their innovations during the product- and achievement configuration of the innovation phase. The search engine offers different search criteria, namely branch, product group and enterprise name. In addition to this basic search a specific search returns those suppliers out of the multiplicity of suppliers that are demonstrable in a position to deliver the required achievements and are suitable to the processes of the actors of the Maritime Cooperation Network. After the job end the contractor should evaluate the work of the supplier so that suppliers are forced to do their work at the best and to establish a mutual trust to enterprises which have never been in cooperation.

- **reference site**
The reference site ought to establish and strengthen the confidence of future customers by publishing the finished deliveries of system products and system achievements.

- **news site**
The news site provides Maritime Cooperation Network partners and external, confirmed companies the opportunity to present changes in the own company and new offers to interested persons and companies, e.g. new product groups, new technologies of manufacturing and capacity increases. Non-permissible information are filtered by the system administrator / editor of the Maritime Cooperation Network.

- **presentation of competencies of the network actors at the platform**
For the presentation of competencies the partial collected information from an other subproject shall be used and especially represented at the Maritime Cooperation Network. Here specific information of existing processing technologies, manufacturing technologies, manufacturing processes, assembly technologies, assembly processes and of the service sector are recorded.
These information document the orientation and problem solving competence of the Maritime Cooperation Network and its actors as a realisation-oriented network. The maintenance is the job of the company of the Maritime Cooperation Network.

- Search engine for products / achievements and competencies at the platform
  The search engine should support the daily business by searching for suppliers, which offer the required product, inside the Maritime Cooperation Network. In the case that this search leads to no result the search engine should ease and support the search outside the Maritime Cooperation Network.

- Environment for the presentation of products and achievements
  This category provides the opportunity for system suppliers to present their products and achievements with specific descriptions. The way of the presentation is determined by the network in cooperation with the customers in order to meet their requirements. The description of the documents can be published and maintained in written form, as files and especially as downloadable CAD-representations. It is the enterprises task to guarantee the realisation of customer requirements and up-to-dateness. In particular the presentation of 3-D-volume models in the defined data format at the platform could affect companies to cooperate with Maritime Cooperation Network.

- enquiry / bid invitation tool
  The enquiry / bid invitation tool should disburden the purchasing of complex products. The bid invitation tool is a collection of information, e.g. product descriptions and CAD-drafts, for a single bidding. The system supplier prepares a bid invitation, i.e. sharing required information and invite defined suppliers and the potential customer in advance. The three parties involved could present their misleading and request further information. The bidding then continuous anonymously, that means the supplier make the offer at the platform and the system supplier defines his criteria of accepting a bid.

- auction tool
  The auction tool should ease and cheapen the procurement of standard products. The bid inviter contacts all possible suppliers and enables them to make a bid. The process of making bids in the defined group is public that means that every supplier is able to have a look at the bids of the other suppliers and thus adjust the price of his offer. After a predefined period of time it is not possible to make a bid and the one supplier awarded the contract who made the lowest offer.

The companies, especially the companies of the project-phase, have established priorities of the functionalities. The demand-pool and the environment for the presentation of products and achievements are named first. Implementing the opportunity for the presentation of product requires the harmonising of data exchange, as mentioned in category two.

*Harmonise the inefficient data exchange*

The data exchange between supplier and supplier and between supplier and customer is one of the main focuses of the Maritime Cooperation Network. Currently two different data-steps will be developed.

The first step concerns the early stage of engineering. It was shown by the shipyards that during the project design the correct information are not available. Klehn (2002) mentions the efficiency of 3-D data. To encourage the project design it was identified that a simple 3-D data file helps them during the early stage of engineering. Therefore it will be implement into the platform a possibility to check in various 3-D data.

After the early stage it was identified a second step. Step two will help supplier and customer during their data exchange. Currently it exists a plenty of different CAD-systems, especially AutoCAD, CATIA, Pro/E used by supply industry and TRIBON, NupasCadmatic, MicroStation used by Shipyards. It is urgent necessary to harmonise the data-files to work more efficient. Therefore the Maritime Cooperation Network wants to implement some standards for data exchange for their members. These standards would be some neutral file formats like STEP, VRML or IGES and some native standards formats like CATIA or Pro/E. This file formats will be harmonised with elected
companies of the supply industry. After selecting the formats they will be implemented for the supply industry to encourage them in the collaboration process.

![Fig.3: Comparison between network frame and network content](image)

![Fig.4: Content of the network frame](image)

3.2.3 Building a network for efficient cooperation

The conception to set up the structure of the Maritime Cooperation Network as an enterprise network has to contain the following two areas of assignment.

1. network frame = 1.component
2. network content = 2.component

The Maritime Cooperation Network consists of two components, as illustrated in Fig.3.

1. Network frame: With the development of the cooperation network one component is generated, which presents a solid platform for the processing of specific tasks and which makes equal demands on partner and tasks in the Maritime Cooperation Network by a framework. This component is the basis for the success of the work of the Maritime Cooperation Network. The framework must be
acquired, specified and extended in dependency on the requirements of the actors. Fig.4 shows possible contents of such a framework and become more specified and enhanced concerning the emphasis in the course of the project.

These aspects of the framework will be filled in close connection with the maritime companies. Currently the roles and tasks of cooperation within the Maritime Cooperation Network are developed. A result, which was developed first, is the central role of a service enterprise. Elected service companies will convert the data files from the supply industry into specific shipyard data files, to support the data exchange-process. The undefined data-exchange from the supply industry to the shipyards is depicted in Fig.5 and it is confronted with the new implementation of service companies into the data exchange process in Fig.6.

2. Network content: The second component generates solutions targeted at the support of specific cooperation projects and cooperation compounds of system suppliers. It refers to the processes of the product- and assignment configuration and the presentation of these achievements to customers and partner in different phases of the performance process.

The Maritime Cooperation Network is based on the Partnerpool of the Baltic-region Maritime Alliance, which maps the competencies of the enterprises in the region. Companies, out of the Partnerpool, have the possibility to become a member of the Maritime Cooperation Network. Before they were integrated into the Maritime Cooperation Network they will run the cooperation check. The system products were produced within the Maritime Cooperation Network in two possible ways. The first is that the Network can offer the whole range of companies of the maritime supply industry. Scenario one shows the production of the product only with members of the Network in Fig.7.
Another manufacturing process is mentioned in Fig. 8. If the Maritime Cooperation Network will not offer the whole range of companies of the maritime supply industry, it is necessary to integrate possibilities to canvass other companies into the manufacturing process.

3.3 Establishing a private operating agency to maintain cooperation process

Companies can use the Maritime Cooperation Network for their daily working processes and for cooperation. For their daily working process, the stage one, the companies can use every help the Network offers without the project-management software. They have to pay a fee for the usage and their presentation on the platform. This will value their competitiveness towards other enterprises. The project management software will be used only for system-products. Stage two would mean that the companies have to pay an additional fee to announce a system-product. This includes the permission
to use the project-management software with some other partners which will be working on this announced project. With the announcement the partner-pool has also some other advantages. Companies can save the rights of a system-product, to make sure that no other company of the network can announce the same system-products in the Maritime Cooperation Network. After finishing the system-products the companies can renew their enrolment or they can abandon it.

References

KIESEL; KLINK (1998), *Die Renaissance der Kooperation*, Stuttgart, in ZWF 93, pp.18-21

KLEHN, B. (2002), Das 3D Datenmodell, Hansa, December, pp.20-21

SCHUH; MILLARG, GÖRANSSON (1998), *Virtuelle Fabrik*, Carl Hanser Verlag

Generic Model Building for Concept Exploration

Thomas Dirix, Delft Univ. of Technology, Delft/NL, ThomasDirix@hotmail.com
Hugo T. Grimmelius, Delft Univ. of Technology, Delft/NL, H.T.Grimmelius@wbmt.tudelft.nl
J. T. Ligtelijn, Wärtsilä Propulsion Netherlands, Drunen/NL, Do.Ligtelijn@wartsila.com
Joost C. Moulijn, Wärtsilä Propulsion Netherlands, Drunen/NL, Joost.Moulijn@wartsila.com

Abstract

Building dynamic simulation models is a complex task, even with the aid of modern user friendly computer applications utilising graphical interfaces, e.g. Matlab Simulink®. This task can be much simplified by using other advanced possibilities of the software. Models can be built automatically only requiring parameter input from the user. The graphical representation of the model is still available to the user for easy reference and inspection. The potential and the limitations of this approach will be discussed. The case shown here is a Concept Exploration Model for Manoeuvrability to be used in the preliminary design stage. This model was developed as a sales support tool for Wärtsilä Propulsion Netherlands B.V. It offers great flexibility in number and location of propulsors, rudders, tunnel and azimuthing thrusters. It allows for both pre-defined (standard) and custom manoeuvres.

1. Introduction

When designing ships, a choice about the means of propulsion has to be made in an early design stage. In most cases decisions are based on arguments of desired speed, fuel economics and/or available engine space. Decisions concerning the required manoeuvring capabilities are generally based on rules of thumb, or experience with a similar type of ship. Only in special cases the actual manoeuvring characteristics are obtained during the design process, either through computer simulations or later through model tests. These characteristics are then used for evaluating whether or not the selected propulsion configuration provides the desired manoeuvrability. The result of such a procedure is that the behaviour probably suffices, but is not optimised. However, optimisation of the manoeuvring capabilities can be justified economically if for example a cruise ship making some 200 port calls a year needs less tugboat assistance due to better maneouvring capabilities.

In order to be able to be of more service to its customers in the early design stage, Wärtsilä Propulsion Netherlands B.V. (WPNL) together with the TU Delft designed and developed a computer program capable of predicting the influence of propulsion configurations on a ship’s manoeuvrability in the concept design stage. A computer program in Matlab Simulink is available now, capable of evaluating the changes in manoeuvring behaviour due to changes in the propulsion configuration when only the ship main particulars are known. This program uses a generic modelling technique that is described in this paper for the case of a maneouvring model. Most of the work was done as part of the MSc project of Thomas Dirix.

2. Degrees of freedom

For a co-ordinate system as shown in Fig.1 and for a body with six degrees of freedom, symmetric in the x-y plane and having principal axes of inertia coinciding with the x and z axes, the following equations apply:

Fig.1: Body-fixed co-ordinate system
$$X = m \cdot \left( \dot{u} + gw - rv \right)$$
$$Y = m \cdot \left( \dot{v} + pu - bw \right)$$
$$Z = m \cdot \left( \dot{w} + pv - qu \right)$$  \hspace{1cm} (1)

$$K = I_{xc} \dot{p} - (I_{yy} - I_{xy}) qr$$
$$M = I_{y} \dot{q} - (I_{xz} - I_{xy}) \varphi$$
$$N = I_{zc} \dot{r} - (I_{zz} - I_{xy}) pq$$  \hspace{1cm} (2)

Since the main goal of the model is to simulate the manoeuvring behaviour of ships, the motions in the horizontal \(x-y\) plane (surge, sway and yaw) need to be studied. If no waves are present and the mass of the ship does not change during the manoeuvre, then no forces other than the constant gravity and buoyancy forces are present in the \(z\)-direction. Furthermore, if the distribution of mass does not change during the manoeuvre there will be no moments acting around the \(x\) and \(y\) axes. As a result there will be no vertical movement, heave, or rotation around the \(x\) or \(y\) axes (pitch or roll). The degrees of freedom are thus reduced to three:

$$X = m (\ddot{u} - rv)$$
$$Y = m (\ddot{v} + m)$$
$$N = I_{zc} \ddot{r}$$  \hspace{1cm} (3)

Thus, calculating the horizontal forces \(X\) and \(Y\) and the yaw moment \(N\) suffices to compare the manoeuvring characteristics for various propulsion configurations.

\textit{Hirano and Takashina (1980)} proved that for high-speed manoeuvres the influence of roll on the turning circle of a ship cannot be neglected, especially for car-carriers and large container vessels. This effect is neglected here because it is not the aim to make a highly accurate quantitative prediction of the manoeuvring characteristics but rather to be able to predict the qualitative differences in these characteristics for the same ship with different propulsion configurations.

\section*{3. Modules}

In a modular structure, all modules must the same number and type of inputs and outputs to make them easily interchangeable. Therefore, first an inventory of all the necessary modules and their inputs and outputs is made. This is followed by an investigation whether or not it is possible to reduce or manipulate the number and type of required inputs and outputs in such a way that they become the same for all modules involved.

\subsection*{3.1. Module inventory}

From section 2 it follows that all elements of the propulsion configuration generating forces and/or moments in the \(x-y\) plane are of importance for the manoeuvring performance of a ship. As a consequence at least the hull, propeller(s), rudder(s), tunnels thruster(s) and azimuth thruster(s) have to be taken into account when modelling. However, the engines driving these components may prove to play a part in the manoeuvring behaviour as well. Therefore the prime mover(s) of the ship, normally a Diesel engine or a gas turbine, has to be taken into consideration.

The term ‘hull’ here refers to the actual submerged part of the hull only. The part of the hull above the water surface together with the superstructure is referred to as ‘superstructure’.

Strictly speaking the term ‘propulsion configuration’ is incorrect. More correct would be ‘propulsion and manoeuvring aids configuration’ since rudders and tunnel thrusters are normally associated with manoeuvring rather than with propulsion. But for reasons of convenience the term ‘propulsion configuration’ will be maintained here.

A first inspection of the aforementioned elements shows the major parameters and variables influencing their behaviour. Table I summarizes the input and output variables.
### Table I: Summary of input and output variables

<table>
<thead>
<tr>
<th>Module</th>
<th>Input variables</th>
<th>Output variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hull</td>
<td>Ship speed</td>
<td>Water resistance (force)</td>
</tr>
<tr>
<td>Superstructure</td>
<td>Ship speed, Ship heading</td>
<td>Wind resistance (force)</td>
</tr>
<tr>
<td>Propeller</td>
<td>Rotational velocity, Ship speed</td>
<td>Thrust (force), Load Torque</td>
</tr>
<tr>
<td>Prime mover</td>
<td>Rotational velocity</td>
<td>Delivered Torque</td>
</tr>
<tr>
<td>Rudder</td>
<td>Flow velocity, Rudder angle</td>
<td>Rudder force</td>
</tr>
<tr>
<td>Tunnel thruster</td>
<td>Rotational velocity, Ship Speed</td>
<td>Thrust (force)</td>
</tr>
<tr>
<td>Azimuth thrusters</td>
<td>Rotational velocity, Ship speed, Azimuth angle</td>
<td>Thrust (force)</td>
</tr>
<tr>
<td>Equations of Motion</td>
<td>Horizontal Forces, Horizontal Moment</td>
<td>Surge speed, Sway speed, Yaw speed</td>
</tr>
</tbody>
</table>

### 3.2. Desired Input and Output

In Table I the majority of input variables for the elements of the propulsion configuration are speeds, while all the outputs are either forces or moments. For the equation of motion module this is reversed. From a generic model building point of view it is necessary to create modules that solely depend on the same inputs, here the ship speeds in the horizontal plane, and have the same outputs, here forces X and Y and moment N, so element modules can be easily ‘connected’ to the equation of motion module. In the following sections, ‘ship speed’ denotes the speed vector containing $u$, $v$ and $r$. Similarly, ‘force’ denotes the vector containing the elements $X$, $Y$ and $N$.

In the following section a simple diagram shows the principal input and output variables and parameters for each module. Fig.2 shows its legend.

![Fig.2: Legend for module diagrams](image)

### 3.3. Hull

Forces acting on the hull are generally calculated using hydrodynamic derivatives obtained through extensive model testing. However, at the design stage in which the program is intended to be used these tests have not yet been performed. Therefore, a way of expressing the hydrodynamic derivatives based on principal ship dimensions was sought and found in Inoue (1981). Inoue also gives a set of equations by which the actual calculations can be performed:
\[ X_{hull} = -m_x \dot{u} + (m_y + X_{re})v + X(u) \]
\[ Y_{hull} = -m_y \dot{v} - m_S u r + \frac{d}{2} \rho L T U^2 \left( Y' v' + Y' r' + Y' r' v' \right) \]
\[ N_{hull} = -J_{zz} \dot{r} + \frac{d}{2} \rho L T U^2 \left( N' v' + N' r' + N' v' r' \right) \]

\[ X' = \frac{X}{\frac{1}{2} \rho L T U^2} \quad N' = \frac{N}{\frac{1}{2} \rho L T U^2} \quad Y' = \frac{Y}{\frac{1}{2} \rho L T U^2} \quad U = \sqrt{\dot{u}^2 + v^2} \]

With seemingly valid arguments, Kobayashi (1988) states that Inoue’s equations do not hold for low speeds and large drift angles. He therefore proposes a different set of equations. However, since his equations depend on hydrodynamic derivatives that are not readily available from principal ship dimensions, Inoue’s equations are used for all cases.

The straight-line resistance \( X(u) \) is derived from a simplified Holtrop and Mennen (1978) approximation, requiring only principal ship dimensions. The ships’ mass \( m \) and hydrodynamic masses \( m_x, m_y \) again are approximated on the basis of the main particulars alone, as is the added moment of inertia \( J_{zz} \). Altogether this results in an element module, Fig.3, in accordance with the desired input and output from section 3.2. The module requires ship speed as input variable, while returning a force as output variable. The necessary parameters are the main particulars of the ship.

![Fig.3: Hull module](image)

![Fig.4: Wind force coefficients for a fully loaded container ship, Brix (1992)](image)
3.4. Superstructure

The forces acting on the superstructure depend on the shape and size of the superstructure and the wind speed relative to the ship. Wind speed itself is user defined in the simulation, but the ship speed relative to the ground, the ground speed, has to be calculated from the (also user defined) current and the actual ship speed. Once this is done not only the relative wind speed but also the relative wind angle is known. With these two arguments it is then possible to enter the graph in Fig.4 to obtain the wind force coefficients $C_x$, $C_y$ and $C_N$. The actual forces are calculated by multiplying the coefficients with the actual superstructure size.

$$X_{\text{wind}} = C_x (\epsilon_{\text{wind}}) \cdot A_{\text{lat}} \cdot \frac{\rho}{2} \cdot U_{\text{appwind}}^2$$

$$Y_{\text{wind}} = C_y (\epsilon_{\text{wind}}) \cdot A_{\text{front}} \cdot \frac{\rho}{2} \cdot U_{\text{appwind}}^2$$

$$N_{\text{wind}} = C_N (\epsilon_{\text{wind}}) \cdot A_{\text{lat}} \cdot L \cdot \frac{\rho}{2} \cdot U_{\text{appwind}}^2$$

(5)

As a result the superstructure module, Fig.5, only requires the speed vector as input variable while returning a force vector. The necessary parameters besides the wind data are the current data, the ship dimensions and the initial ship heading $H_0$. The latter is necessary to calculate the actual ship’s heading based on the integration of yaw speed.

$$H_s = \int_{t=0}^{t_{\text{end}}} r \cdot dt + H_0$$

(6)

![Fig.5: Superstructure module diagram](image)

3.5. Propeller module equations

Propeller thrust and torque are calculated through the use of four-quadrant curves. These curves are available for various propellers from the Wageningen B-series. Although this approach is not directly suited for simulation of controllable pitch propellers (CPP), it was chosen because it enables the calculation of thrust and torque in situations of large drift angles or propeller reversing. With the argument of inflow angle $\beta$, obtained on the basis of inflow velocity and propeller rotation velocity, the four-quadrant graph is entered to obtain the thrust and torque coefficients $C_{T^*}$ and $C_{Q^*}$, from which the actual thrust $T$ and torque $Q$ can be calculated:

$$C_{T^*} = \frac{T}{\frac{D}{2} \left( \nu_a^2 + (0.7 \cdot \pi \cdot \nu_a \cdot D)^2 \right) \frac{\pi}{4} D^2}$$

(7)

$$C_{Q^*} = \frac{Q}{\frac{D}{2} \left( \nu_a^2 + (0.7 \cdot \pi \cdot \nu_a \cdot D)^2 \right) \frac{\pi}{4} D^2}$$

(8)

The thrust deduction factor $t$ and the wake fraction $w$ are approximated by empirical formulas only requiring the main ship particulars - length, beam, draft and block coefficient - as input parameters.
This results in the module shown in Fig.6: the module requires not only the ship speed but also the rotational velocity as input variable, while delivering both a force and a torque. This does not satisfy the desired input and output from section 3.2.

3.6. Prime mover

The only prime mover presently available in the program is a Diesel engine. The Diesel engine model, a mean value model based on a 5-point Seiliger cycle, is capable of simulating various makes and types of engines on the basis of project guide data alone.

The basic idea of the model is that, based on the geometry of the engine and data available from the project guide, the theoretical amount of nominal indicated work can be calculated, assuming no heat losses. The found quantity of work is then corrected for heat losses and mechanical losses to fit the actual nominal effective work derived from the project guides value for break mean effective pressure. Stapersma and Grimmelewius (2001) argue that the mechanical efficiency $\eta_{mec}$ of a modern 4-stroke Diesel engine can be expressed as a function of charge air pressure $p_1$:

$$\eta_{mec} = 1 - \frac{c_m}{p_1}$$

(9)

$c_m$ is an engine specific constant which for most 4-stroke turbocharged modern Diesel engines does not differ much from 0.4. They also show that for a modern 2-stroke Diesel engine a constant mechanical efficiency of about $\eta_{mec} = 95\%$ can be assumed.

A similar relation, albeit valid for both 2 and 4-stroke engines, exists between the heat input efficiency, $\eta_q$, and the charge pressure, Stapersma and Grimmelewius (2001):

$$\eta_q = 1 - \frac{c_q}{p_1}$$

(10)

$c_q$ is an engine specific constant. The value of this constant varies for different engines, and can only be calculated directly using data not provided in the engines project guide. However, here $\eta_q$ is used to match the calculated value for $hmep$, and thus the total delivered power of the engine, with that given in the project guide. This is done through variation of $c_q$. The result is a ‘heat input efficiency’ which not only accounts for the heat losses during heat input, stages 2 and 3, but also accounts for the influence of the absence of the isothermal combustion stage, for the influence of the assumed
isothermal compression and for the influence of possible error in the choice of the polytropic exponent for stage 4.

To simulate the turbocharger, the charge air pressure is estimated based on the fact that the heat release at stage 5 is a measure for the energy available to drive the turbine and thus also for the work performed by the compressor. Based upon this, an ‘efficiency’ is divined between the heat available and the work needed to create the charge air pressure. For the nominal situation this efficiency is calculated, from data available in the project guide. This efficiency is then assumed constant over the working range of the engine, enabling the calculations of charge air pressure in part load conditions.

All in all this results in an engine model that only requires data regarding engine dimensions and nominal operating values from the project guide as input parameters, as well as a value for the desired rotational velocity, Fig.8. Based on the difference between the desired rotational setting and the input variable of actual rotational velocity the module returns the torque as output variable. As for the propeller, this does satisfy not the desired input and output from section 3.2.

![Fig.8: Engine module](image)

3.7. Combining prime mover and propeller

The output from the diesel engine, the delivered torque $M_{del}$ and the input into the propeller module, the load torque $M_{load}$, not matching the desired input and output are eliminated by combining the two modules. The difference between the delivered torque and the load torque is the accelerating torque working on the mass moment of inertia of the whole shaft system $I_{total}$, i.e. prime mover, gearbox, shafting and propeller. When the delivered torque is higher than the load torque, the rotational velocity $\omega$ increases and, vice versa, when the difference is negative the rotational velocity decreases.

\[
M_{res} = M_{del} - M_{load} = I_{total} \cdot \frac{d\omega}{dt} \Rightarrow \omega = \int \frac{M_{res}}{I_{total}} dt
\]  

![Fig.9: Combined engine-propeller module diagram](image)
The calculation of Eq.(11) is performed in a module called ‘rotational dynamics’. Fig.9 shows the combined modules. Note that it is possible to ‘connect’ multiple prime movers to a single propeller by summing the delivered torques to a ‘total delivered torque’.

3.8. Rudder

The normal force generated by the rudder is obtained through:

\[ F_N = \frac{1}{2} \rho \cdot A_R \cdot U_h^2 \cdot f_u(A) \cdot \sin \alpha_\text{eff} \quad \text{(12)} \]

\[ \text{With:} \quad f_u(A) = \frac{6.13A}{A + 2.25} \quad A = \frac{h^2}{A_R} \]

The component of the normal force acting on the ship in x and y direction and the rudder moment are:

\[ X_R = -F_N \cdot \sin \alpha_R \]
\[ Y_R = -F_N \cdot \cos \alpha_R \]
\[ N_R = -\frac{1}{2} L \cdot F_N \cdot \cos \alpha_R \]

(13)

The major difficulty of the rudder module is the determination of the inflow velocity. This velocity not only depends on ship speed, but also depends on the generated propeller thrust. It proved not to be possible to eliminate this dependency, therefore the inputs for the rudder module are both propeller thrust and ship speed. The output variable is rudder force, Fig.10.

The steering gear is assumed to be able to generate a constant rotational velocity of the rudder. This rotational velocity is an input parameter for the module.

![Rudder module diagram](image)

**Fig.10: Rudder module**

3.9. Reduced Tunnel Thruster

For a tunnel thruster, the influence of the engine dynamics on the ship’s manoeuvring characteristics is assumed to be limited. The desired speed of the tunnel thruster is reached instantaneously. This simplification results in the tunnel thruster module shown in Fig.11. When desired, for more detailed simulations, an approach similar to section3.7 can be used to create a combined engine and thruster module, allowing to investigate the influence of the engine dynamics.

![Tunnel thruster module diagram](image)

**Fig.11: Reduced Tunnel thruster module diagram**
The program checks if the user’s input of desired tunnel thruster power, propeller revolutions and tunnel diameter are within the range of a series of standard thrusters for which pump curves are known. If the input is not consistent, the diameter is varied until the operating point is within range. This calculation uses a series of formulas inventoried by Hermsen and Hendrikse (1995) to calculate the total tunnel resistance coefficient \( \zeta_{\text{tunnel}} \). When the operating point is within range, its exact location is calculated by determining the point where pump curve and system curve intersect. The non-nominal behaviour is then related to the number of revolutions.

\[
T(n) = T_{\text{tot, nom}} \cdot \left( \frac{n}{n_{\text{nom}}} \right)^2
\]

(14)

The total force acting on the ship is subsequently derived from

\[
Y_{\text{tunnel}} = \frac{2 \cdot T_{\text{tunnel}}}{1 + \zeta_{\text{tunnel}}}
\]

(15)

Hendriks and Hermsen (1995) also compared several researches into the effect of ship speed on thruster effectiveness. They arrived at the conclusion that a general trend is present. This trend is shown in Fig.12 where the ratios of actual thrust and moment compared to the nominal situation are shown as a function of the ratio between ship speed and jet velocity, \( m \). The nominal values for thrust and moment are those at zero advance speed of the ship.

![Fig.12: Thruster effectiveness](image)

**3.10. Reduced Azimuth Thruster**

The same procedure as for the propeller module (section 3.7) is followed for the azimuth thruster module. In addition, it is assumed that the speed at which the thruster turns to a new azimuth angle is independent of the forces involved. The same assumption was made for the rudder. Similar to a normal propeller, the thrust and torque are calculated using four-quadrant curves. But contrary to the situation for a ‘fixed’ propeller it proved not possible to find a generic set of equations enabling the calculation of thrust deduction and wake fraction. This is mainly due to the large number of possible positions for an azimuth thruster. Therefore the thrust deduction is not accounted for in the thrust calculation. The wake fraction only accounts for the influence of the strut.

**Thruster interaction effects**

Lehn (1980) performed experiments to obtain an insight into thruster interaction effects at zero speed. Nienhuis (1992) developed a mathematical model to calculate these effects and validated it with Lehn’s test results as well as with some new tests performed at low speed. He arrived at the conclusion that the main interaction effect derives from the flow velocity induced by neighbouring thrusters. However, the numerical approach he used to calculate the slipstream velocities is not suitable for use in a dynamic model. Furthermore he only validated his results at low speed.
Fig. 13: Combined engine-azimuth thruster module diagram

Since the thruster interaction is too large to neglect, it is accounted for by extrapolating the results obtained by Nienhuis with regard to slipstream diameter and slipstream velocity distribution to higher advance ratios. Also it is assumed that the differences between the various types of thruster nozzles are of minor influence on this effect. This results in an azimuth thruster module that depends on the input variable ship speed and on the slipstream velocity and direction of any other present thrusters, Fig. 13.

4. Model composition

All the discussed modules return a force (vector) as output. The sum vector of all these forces is the input variable for the equations of motion module, which in turn returns the surge, sway and yaw speeds, \( u \), \( v \) and \( r \). For all modules (except the rudder and azimuth module), this ship speed vector returned by the equations of motions is sufficient input to calculate the next time step. The rudder additionally requires the propeller thrust. The thrust is propagated from the propeller module. The azimuth module requires additional data concerning the slipstreams of the surrounding thrusters. Since the position of the surrounding thrusters does not change, these can be entered as parameters. The actual slipstream position and trajectory do change, and are calculated on the basis of the actual ship speed and azimuth thruster thrust.

Fig. 14: Example model configurations

It is now possible to compose a seemingly unlimited variety of models, a few of which are shown in Fig. 14. In the right-hand figure the thruster force feedback is not connected to the other azimuth thruster by a connection line but via ‘send’ and ‘receive’ elements. This is done in order not to lose sight of the overall structure of the model.
5. Implementation

The method of modelling presented here is implemented in Matlab Simulink. Simulink allows the user to create his own library containing user-defined simulation blocks. In this case every single module is represented by such a user-defined block, enabling the user to use the drag-and-drop facilities Simulink offers to build a model. The parameters necessary for every module can then be entered through the blocks interface. However, a Matlab routine was developed performing the composition of the model automatically to simplify the use of the program further. The user now only has to enter the desired number of each element and define the necessary parameters. The program then connects the proper library blocks to create the Simulink model. Subsequently all the required parameters are loaded into the model. Simulations required for the manoeuvres requested by the user are then performed automatically. Results are presented either through standards plots available from a menu in the program or by user-defined plots. The resulting Simulink model is still available to the user for inspection and detailed evaluation.

Fig. 15: Hull definition window

Fig. 16: Engine specification window

Fig. 17: Model composition windows
The equations used for every module reduce the number of parameters necessary to an absolute minimum and to parameters that in general are easy to obtain. After all necessary parameters are entered, a module instance is created and a new model can be composed by selecting the desired module instances and entering some additional over-all model parameters. The parameters necessary for such a module instance – e.g. parameters of a specific engine, a hull shape, a propeller, etc. - are stored in the database. This allows for the user to select a specific instance of a module without having to define the parameters in every simulation. Fig. 15 shows the initial screen asking the user to define a hull shape. This hull shape is then stored in the database for future use. Fig. 16 shows the screen asking for the user input for a Diesel engine. Similar windows exist for the definition of new propellers, superstructure and tunnel thrusters. Fig. 17 shows several windows necessary for the composition of a new model. The same procedure is followed with regard to the manoeuvres. Various standard manoeuvres such as turning circles and zigzag manoeuvres are stored in a database, enabling easy selection for the user. User-defined manoeuvres can be added to the database for future use.

6. Results

The method of modelling and composing described here proved very useful in creating and comparing various propulsion configurations for a given ship. Fig. 18 and 19 show two propulsion configurations created by the program after the user defined the number, type and position of each element. The results obtained from a simulation are easily presented either through the built-in Simulink ‘scope’ blocks or through the customized diagrams and graphs available through the program. Fig. 20 shows three of these customized graphs. The first displaying a starboard turning circle, the second displaying the actual engine revolutions compared to the desired engine revolutions during a turning circle and the last graph presenting the ship’s heading and rudder angle during a zigzag manoeuvre.

Fig.18: Twin azimuth thruster with triple tunnel thruster configuration

Fig.19: Single Propeller with two tunnel thrusters configuration

Fig.20: Starboard turning circle track plot; engine revolutions during turning circle; ship heading and rudder angle during Zigzag manoeuvre
7. Conclusions

For certain types of ships, a fairly good comparison can be made regarding the maneouvring behaviour for various propulsion configurations. This way of modelling has almost no restrictions with regards to the number of elements or combinations of elements, and is therefore well suited for concept exploration. The way the model is implemented proved to be user friendly, allowing the program to be used as a sales support tool by users with no modelling or Matlab experience. The accuracy of the model depends strongly on the algorithms used inside the modules. E.g., the accuracy can be improved by replacing Inoue’s hydrodynamic derivatives and equations for hull resistance by actual data found through model testing. The advantage is that such an improvement does not require a totally new program, only the addition of a new user-defined block containing the new hull module to the Simulink block library. It is then up to the user which of the two hull modules he wants to use, depending on available data. Within a generic model the modelling of a larger number of interaction effects is possible, but requires a lot of programming and may decrease the flexibility of the program. Therefore it is recommended that the number of interaction effects is kept at a minimum necessary to create useful simulation results.

References


GERRITSMA, J. (1987), Scheepsbewegingen, sturen en manoeuvreren 2, Delft University of Technology


KOBAYASHI, E. (1988), A simulation study on ship maneouvrbility at low speeds, Mitsubishi Technical Bulletin No 180


STAPERSMA, D., GRIMMELIUS, H.T., (2001), Concept Exploration applied to Diesel engines, 23rd CIMAC Conference 2001, Hamburg
Three Levels of Fuel Optimization at Sea

Thomas Hellström, Umeå University, thomash@cs.umu.se

Abstract
This paper discusses the experiences of a challenging research and development project: Design and implementation of a fuel saving control system for large vessels. The developed hardware and software has been successfully installed on around 20 ferries around Europe and has reduced the fuel consumption by as much as 5-15% in many cases. The fuel saving is achieved by optimizing control at three levels ranging from low level propeller and main engine control up to route planning for optimal speed profiles compensated for varying depth and weather conditions. The control problems involve classical control functions as well as numerical optimization. Other important issues that are discussed in the paper are the safety aspects in designing and building semi-autonomous systems for high-level control of large vehicles.

1 Introduction

The fuel costs are the second largest item (after salaries) on a big vessel’s budget. The fuel consumption for a large ferry ranges between 1000 and 5000 liters per hour. This means that the ship consumes more oil per hour than a one-family house does for one whole year’s heating (in northern Sweden). The annual fuel budget for a ferry running 20 hours per day is in the order of millions of dollars. Even small reductions of a few percent off the fuel consumption means considerable annual savings.

This paper discusses the experiences of a challenging research and development project for fuel saving and top-level control of a ship’s performance. The developed system is operated from the central unit placed on the bridge (see Figure 1). The operator, normally the ship’s First Officer or Captain, inputs the required values for speed, arrival times, and complete route plans from the keyboard. The main engines (5,000-30,000 kW) and propellers are then automatically adjusted, to reach and maintain the required speed at the lowest fuel consumption. Fuel saving is typically 5-10%, corresponding to at least 1 cubic meter of heavy fuel oil per day. The saving is achieved by optimizing control at three levels:

1. Pitch optimization. The pitch angle of the blades on a controllable propeller acts as a kind of gear box, and affects the ship’s speed together with the main engine’s revolutions (rpm). The optimal combination of pitch/rpm depends on a number of external and time-varying conditions, and therefore must be subjected to dynamic optimization to be optimal.

2. Dynamic control of speed to avoid sudden peaks in fuel consumption caused by low water depth, or unanticipated changes of the weather conditions.

3. Route planning. The fuel consumption for a ship depends not only on speed, but also on water depth and weather conditions. The optimal speed distribution along the route can be computed in advance, if a weather forecast is available.

The described hardware and software have been implemented as part of the Seapacer system which has been installed on around 20 ferries across northern Europe. The most recent installation was on MS Gotland (Gotlandsbolagen) in 1997. This paper discusses the underlying principles and the experiences of the research and development of the system. Section 2 describes the design and basic operation. The three levels of optimizing control described above are covered in more detail in Sections 3,4 and 5. Section 6 discusses general experiences and difficulties encountered during the project.
The central unit placed on the bridge of the vessel. The left screen is used for real-time control tasks such as speed settings while the right screen is used for long term voyage analysis and follow-up.

Block diagram of a basic system. The central unit takes over control from the maneuvering handles and controls the main engine speed and propeller pitch.

2 The Design of the Control System

The basic layout of the system is shown in Figure 2. The central unit takes over control from the maneuvering handles of the main engine’s speed (revolutions per second) and of the propeller pitch. The most important components are shown in the figure. More information is found in Sections 2.2, 2.3, 2.4.

2.1 Basic Modes of Operation

From the user’s perspective, the system is a tool for high-level control of the vessel’s speed and fuel consumption. The main functionality can be described as a number of control systems aiming at obtaining and keeping set values for speed, fuel consumption or arrival time. During operation, any of the following operational modes may be selected:

- SPEED Fixed speed.
- L/NM Fixed fuel consumption per nautical mile (liters/nautical mile).
Figure 3: Off-line analysis of fuel consumption versus water track speed. Each dot corresponds to one voyage. The two clusters correspond to different numbers of main engines.

- L/H Fixed fuel consumption per hour (liters/hour).
- POWER Fixed power from engine.
- POS/ARR Adapt speed to planned arrival and waypoint defined by latitude and longitude.
- DIST/ARR Adapt speed to planned arrival and distance.
- ROUTE Describe which mode to use at what time during the voyage. The system does the alternation between modes automatically. This mode is described in Section 5.

Besides the pure control functions, the system contains extensive support for data logging and analysis. Various types of lists and charts can be produced, enabling for instance, long-term analysis, comparative evaluations before and after a reconstruction, maintenance at the shipyard, etc. Examples can be seen in Figure 3 and 4. In Figure 3, data from a large number of voyages is presented. The fuel consumption (liters/nautical mile) is shown as a function of the water track speed. Each dot represents the mean values for one voyage. The two clusters correspond to day and night trips, each having a different number of active main engines. The diagram in Figure 4 shows the relation between speed, fuel consumption, and wind strength for a two-month period. Diagrams like these ones aid in the crew’s constant effort to minimize fuel consumption by route planning and scheduling of timetables.

For educational purposes, a simulation function is provided. It is built up around similar parameters to those of the vessel in question. More information about the functionality of the system can be found in the User’s Guide: *Seapacer Optimizing System Mark II* Hellström (1996).

### 2.2 Inputs

The system interfaces with a large number of sensors and sub-systems on the vessel. The types of inputs vary from ship to ship, and the software has to be easy-to-configure. Many sensors
supply digital outputs with a frequency or pulse length proportional to the measured quantity. Other, more modern equipment, output data in ASCII format via an RS232 connection. Quite often, signals are already used for other purposes. It is therefore essential to connect the system so that these other functions on the ship are not affected. All inputs are galvanically isolated from the computer (the voltage differences can be more than 200 volts between two different ground points). The following signals are normally connected to the system:

- Ship’s speed over ground and through water
- Fuel consumption
- Propeller revs (revolutions per second)
- Main engine power (horsepower)
- Water depth below keel
- GPS navigator. Outputs position and optionally bottom track speed
- Trim
- Number of engaged main engines
- Shaft generator engaged (starboard and port)

Many of the signals are highly noisy and also give off completely incorrect signals from time to time. This has to be handled in a stable manner by the software by filtering and outlier detection. Sensor fusion is also utilized for the estimation of bottom track speed. The primary source for speed is the Doppler log, which measures the echoes of ultrasound pulses against
the bottom. This normally works well, but can sometimes give off incorrect values due to false
echoes or too large water depth. A differential GPS navigator provides an alternative speed
source. In older systems, this signal is often updated too slowly or too much delayed to be
useful as input in the actual control system. However, the GPS speed is useful as a backup for
the speed log, if and when the speed log fails. Likewise, the speed log is used as complement
to the navigator. The navigator sometimes loses the signal from the satellites. The speed log
is then, in combination with the last estimate of the ship’s course, used for dead reckoning to
update the estimate of the ship’s position.

2.3 Outputs

The main unit is integrated into the vessel’s existing control system by galvanically isolated 4-20
mA outputs (separate for starboard and port sides):

- Main engine revs control. Connected so that a zero signal corresponds to idle speed of the
  vessel (not zero revs). This is important for safety reasons, since a disconnected or failing
  system should not cause the engines to come to a complete stop.

- Propeller pitch control. The pitch of the propeller is the angle of the propeller blades, and
  acts as a gearbox for the propulsion of the vessel. Zero angle results in zero propulsion, even
  with the engines running at a high speed. The direction of the blades makes it possible for
  the vessel to move forwards or backwards without changing the rotation direction of the
  propellers. The way to connect to the pitch control system is therefore very important, not
  least for safety reasons. The normal way to do it is by interfacing to an existing system.
  The load controller, responsible for the low-level control of the pitch. The load controller
  can be controlled in such a way that central unit only gets access to reduce the pitch from
  the current set point, down to around 80%. In this way the safety issues are left with the
  design of the load controller, and the system has no possibility to output fatal commands.

- Takeover. Relay contacts, which close when the system should take over rpm and pitch
  control from the manual manoeuvering handles. The way this function is implemented is
  extremely important, since a failure in operation could cause the vessel to be completely
  inoperable. Also, the Takeover module must have intelligence to handle some of the safety
  issues described in the next Section.

2.4 Safety Issues

A typical vessel with the system installed has a main engine power of 10-40,000 horsepower.
The vessel itself may be well over 100 meters long, and has a considerable momentum, which
makes the control problem both difficult and important. For safety reasons, the system must be
installed in such a way that it can be bypassed by the operators at any time. This bypassing
functionality has to be implemented at many levels, and designed in such a manner that the
takeover procedure is totally intuitive for the operator, even under mental stress. Control may
be transferred back to the operator handles by any of the following procedures:

- Issuing a *Handle* command on the main keypad. This is the normal way to transfer control
  from the system to the operator. However, it cannot be expected to be used by a stressed
  operator in a situation of emergency

- Turning a designated *In Control* switch to the OFF position. This disconnects the system
  from the ship control functions electrically and is used as a safety precaution, e.g. when
  servicing the system.
• Moving the manoeuvring handles of the vessel below a certain set point. This is the most natural way for an operator at sea. In a situation of emergency, the speed of the vessel has to be reduced drastically in most cases, and the normal way to do this is to pull the manoeuvring handles to zero or even to full backward speed (a.k.a. crash stop). By sensing the manoeuvring handles, the system can automatically disconnect itself and transfer all control back to the operator.

In addition to these manual ways of transferring control from the system to the manoeuvring handles of the vessel, the system automatically transfers control by two functions:

• Watch dog. An electrical timer function that automatically transfers control back to the handles if the timer is not periodically (e.g. every second) reset by the software. This ensures that a computer error (software or hardware) does not cause the system to hang or issue unpredictable control signals to the main engines or propellers.

• Power failure. The takeover electronics automatically falls back to manual control in the case of a power failure.

Another related issue to bear in mind when designing a high-level control system such as the one presented, is how to disconnect the entire system when the vessel is serviced or repaired. There should always be a simple means, by which the entire installation can be removed, leaving the ship in a fully operational mode. Just as the potential fuel saving is attractive to the ship owner, so an interruption in the operation of the ship is totally unacceptable.

3 Pitch Optimization

The pitch angle of the blades on a controllable propeller acts as a gearbox, and controls the ship’s speed along with the main engine’s revolutions (rpm). The optimal combination of pitch/rpm depends on a number of external conditions, and therefore must be subjected to dynamic optimization to be optimal. The system minimizes the consumption of fuel by maintaining an optimal ratio between the propeller’s pitch and the speed of rotation. The optimization aims at minimizing the fuel consumption, measured as consumed oil per nautical mile, for a given set speed \( s_{set} \). The directly measurable entities are water track speed \( s_{wt} \) (nautical miles per hour) and fuel consumption \( c \) (liters per hour). Both \( s_{wt} \) and \( c \) are functions of the pitch \( p \) and main engine revs \( r \). Hence, the pitch optimizer tries to solve

\[
(r_{opt}, p_{opt}) = \arg \min_{r,p} \frac{c(r,p)}{s_{wt}(r,p)}
\]  

(1)

with the constraint

\[
s_{wt}(r,p) = s_{set}
\]  

(2)

where \( s_{set} \) is the set speed for the vessel. \( s_{set} \) is given explicitly by the operator if running in SPEED MODE or implicitly if running in POS/ARR or DIST/ARR mode (see Section 4.3 for details). The optimization problem has to be solved in real-time with both \( c \) and \( s_{wt} \) being extremely noisy. Furthermore, the time constants involved in the processes generating \( c \) and \( s_{wt} \) are large. This means that a change in \( r \) or \( p \) not immediately causes a measurable change in neither \( c \) nor \( s_{wt} \). By the time \( c \) and \( s_{wt} \) respond, the process may very well have a new characteristic, i.e. the optimal values \( (r_{opt}, p_{opt}) \) may have changed. Altogether the optimization problem is indeed very hard. The implemented solution uses an algorithm that first controls \( r \) and \( p \) such that constraint 2 is fulfilled. In the next stage, \( r \) and \( p \) are moved in one direction until a local min value for \( c(r,p) \div s_{wt}(r,p) \) along this direction has been detected. The step sizes
for \( r \) and \( p \) are set so the reduction in engines revs \( r \) is approximately balanced by the change in pitch \( p \). In this way the constraint 2 is approximately fulfilled during the search operation. If necessary, \( r \) is finally adjusted so the constraint is not violated. The system then waits, either a predefined period of time, or until a detection algorithm signals that a new search may be fruitful. The search direction is now reversed.

The algorithm has worked well, but needs steady and fast responding fuel signals to be meaningful. This is seldom the case with ordinary fuel meters installed on the ship for ordinary purposes.

4 Dynamic Control

The system’s basic functions are a set of controllers for speed, fuel consumption liters/hour, fuel consumption liters/nautical mile, and shaft power. These controllers may be used as such by issuing set points from the keyboard. One example can be seen in Figure 5, where the user has issued a SET SPEED command that governs the main engines so that the bottom track speed maintains 17.8 knots. The controllers are ordinary PID controllers, which control a linearized version of the physical entity to be controlled. For example, to control the speed \( s \) of the vessel, a model \( f \) for the static dependency between \( r \), issued rpm (main engines revs), and \( s \), is utilized. The relation is given by \( r = f(s) \) where the function \( f \) is approximated from sampled data and linear interpolation. Different functions have to be used for different numbers of engaged main engines. The speed controller acts on the \( f \) entity:

\[
    r = k_p E + k_i \int E \, dt + k_d \frac{\partial E}{\partial t}
\]

(3),

with the control error \( E \) defined as

\[
    E = f(s_{set}) - f(s_{act})
\]

(4),

where \( s_{set} \) is the commanded set speed, and \( s_{act} \) is the ship’s actual speed. In practice, the derivative part is not used, i.e.: \( k_d = 0 \) in most cases.

The control of the main engines has to be done in a gentle way to avoid unnecessary rapid thermal changes. Of course, this can be adhered to in the tuning of the PID controllers, but other functions have also been added. It is possible to limit the speed, by which the controllers are allowed to change the main engine revs (i.e. the unit for the limit is revs/sec²). This causes the main engines to operate more smoothly than when run manually.

4.1 Handling Fuel and Power Limits

Figure 5 also illustrates some additional control functions in the system. In the example, the user has entered a fuel consumption limit of 180 l/nm (liters per nautical mile). This is treated as a constraint in the control algorithm, and has a higher priority than the set speed, which is the actual control entity. In the same manner, a lower (2000 kW) and a higher (19000 kW) power constraint was entered. The fuel limit and upper power limit serve as safeguards against temporary and unanticipated increases in the load, caused by changing weather conditions, or low water depth below the keel. The lower power limit is necessary to ensure acceptable working conditions for the main engines. In the control system, the constraints are handled as penalties by modifying the control error \( E \) to reflect the violated constraint, for example, excessive fuel consumption. The modification is done by a model function \( g \) that relates the constraint entity to the control entity (normally the speed). In speed mode with a fuel consumption limit, a function \( s = g(c) \) is used. \( c \) denotes the fuel consumption and \( s \) denotes the speed that approximately corresponds to \( c \). Like the function \( f \) in the previous section, the function \( g \) is approximated
from sampled data and linearly interpolated. Assuming a set fuel consumption limit \( c_{\text{max}} \) and a sampled fuel consumption \( c_{\text{act}} \), the control error \( E \) is now computed as

\[
E = \begin{cases} 
  f(g(c_{\text{act}})) - f(g(c_{\text{max}})) & : \quad \text{if } c_{\text{act}} > c_{\text{max}} \\
  f(s_{\text{set}}) - f(s_{\text{act}}) & : \quad \text{otherwise.}
\end{cases} \tag{5}
\]

In practice, the sharp switch point in the definition of \( E \) is smoothed so the penalty starts to work already before the limit is violated. The effect of the modified control error is that too high a fuel consumption (i.e.: \( c_{\text{act}} > c_{\text{max}} \)) makes the controller act as if the speed is too high, even if \( s_{\text{set}} > s_{\text{act}} \). Since expressions 4 and 5 only compute differences of the functions \( f \) and \( g \), the absolute values for these models are not critical. The purpose of using them is to linearize the process for the PID controller.

4.2 Dynamic Set Points for Fuel and Power Limits

The fuel and power limits described in the previous section serve as safeguards against temporary increases in the load. The result of such an increase, e.g. caused by shallow waters, is a slowing down of the vessel to reduce the fuel consumption or power below the set upper limit. This can be a very useful function as such, provided the settings of the limits are done carefully. However, too hard limits prohibit the system from keeping the arrival times, while too loose limits are without effect. To eliminate the need for manual choice of limits, a dynamic computation of suitable values has been developed. It works by slowly lowering the upper limit until the limit almost becomes active, i.e. when the actual fuel consumption, or the main engine power necessary to maintain the speed, almost reaches the dynamically set upper limit. In this way a sudden increase in load causes the limit to be violated and the system to lower the speed. However, after a pre-defined delay time, the dynamically set limit is slowly adapted upwards, to allow the vessel to run at the necessary speed in the long run. The result of the dynamic set points for fuel and power limits is that of evening out the power over the entire route. This results in lower total fuel consumption.

4.3 Automatic Computation of Speed Set Points

The main objective for the crew of a ferry is normally to keep the set arrival times. For this purpose, a function that dynamically designates the SET SPEED of the speed controller, is implemented in the DIST/ARR running mode. The user enters the arrival time and distance to the goal. The vessel now runs at the lowest speed possible, while still arriving in time. The speed necessary to travel the distance is updated continuously. No updating takes place for the five last minutes ahead of estimated arrival time. When 0.5 NM remain of the stated distance, the system automatically changes control-mode to SPEED MODE, using the last set speed as the new set speed. The distance is computed by integrating the log signal (speed relative to ground).

A similar run mode POS/ARR works by using a given geographical location (waypoint), instead of a certain distance to travel. In this way the system is more tolerant for deviations from the originally intended routes than in DIST/ARR mode. A more complete handling of entire routes is implemented in the ROUTE PLANNING run mode described in more detail in Section 5.

5 Route Planning

The route planning of a ship with varying speed in different parts of the route is designed to keep the set arrival time, while reducing the total fuel consumption. The system automatically optimizes the speed distribution between the route legs. Legs with different depths and/or weather conditions then run at different speeds to minimize the total fuel consumption.

5.1 What Affects the Fuel Consumption

Following are some of the most important factors that affect the fuel consumption of a ship:
Figure 5: Main screen showing real-time values for all connected sensors and given commands.

- Ship-specific parameters such as form of the hull, weight, main engines, propellers, etc.
- Number of engaged main engines.
- The ship speed relative to the ground, measured in knots (denoted "bottom track speed").
- Water currents (direction and speed in knots).
- Water depth under the keel.
- The ship’s draft (depends on the cargo).
- Wind and waves (direction and strength measured in Beaufort points).

5.2 How Can the Fuel Consumption Be Reduced?

Route planning consists of varying the ship’s speed in different parts of a route. Since the external conditions (wind, current, and depth) vary, it is evident that fuel consumption cannot be maintained at a minimum, if a constant speed is kept throughout the route. Therefore, we get a minimization problem that has to be solved numerically: we have to find the speed distribution that minimizes the total fuel consumption within the constraint of keeping the scheduled arrival time. Route planning of some kind or another is done on all ships. Most often the "calculation" consists of manual estimates, based on previous experience from the same route. The developed system contains functions for automatic route planning. Wind, current, and water depth can be input by the operator before departure or during the voyage. The system then automatically calculates a speed profile that minimizes the total fuel consumption. Based on the computed speed profile, the computer regulates the ship's speed by controlling the main engines and the propellers. The arrival time is kept without unnecessary margins. Following is a description of the basic route planning system.

5.3 Models

The dependency of fuel consumption upon speed, wind, and water depth is essential for the route planning and optimization. Analytical models are rare and are not general enough to be used
for all sorts of ships. Therefore, data is sampled at different running conditions, and gathered in tables. These tables serve as models for the optimization and route planning. The values shown in the following tables are examples from a ferry running between Hook van Holland and Harwich on the English Channel. The sampled values do vary from one ship to another, but they normally share the same characteristics. Values in-between points are estimated by a 1- or 2-dimensional linear interpolation.

5.3.1 Speed Models \( F(x_w) \)

Table 1 shows fuel consumption \( Cons \) (liters/hour) for different speeds through water \( x_w \) (knots). The data has been sampled with no influence from limiting water depth. The function \( F(x_w) \) is defined as linear interpolation in the table.

<table>
<thead>
<tr>
<th>( x_w )</th>
<th>( Cons )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.4</td>
<td>650</td>
</tr>
<tr>
<td>13.2</td>
<td>875</td>
</tr>
<tr>
<td>17.0</td>
<td>1300</td>
</tr>
<tr>
<td>20.1</td>
<td>2120</td>
</tr>
<tr>
<td>20.7</td>
<td>2900</td>
</tr>
</tbody>
</table>

5.3.2 Depth Model \( D(x, d) \)

In limiting water depths the ship’s speed decreases due to the increased water resistance. In very shallow waters (typically a few meters below the keel), the so-called “squat effect” pulls the ship downwards, thereby reducing its speed further. Table 2 shows the fuel consumption increase (\%) for different water depths \( d \) and speeds \( x \). The function \( D(x, d) \) is defined as a 2-dimensional linear interpolation in the table.

<table>
<thead>
<tr>
<th>( x )</th>
<th>( d )</th>
<th>Increase (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

5.3.3 Wind Model \( W(w, w_d) \)

The wind is specified by the direction relative to the ship’s heading and the wind strength expressed in Beaufort points. The parameter \( w_d \) is the wind direction (degrees). \( w \) is the Beaufort degree (0-10). Generally speaking, wind from the side and even slightly from behind increases the fuel consumption due to the large open surface on the sides of many ships. Table 3 describes a typical and approximate relation between increased wind strength, direction, and increased fuel consumption for each unit of Beaufort. The wind direction is measured clockwise, relative to the ship, with zero degrees defined as a wind blowing along the ship from bow to stern. The function \( W \) is defined as: \( W(w, w_d) = w \cdot I(w_d) \), where \( I(w_d) \) is determined by a linear interpolation in Table 3.
Table 3: Increase in fuel consumption (%) due to wind from different directions

<table>
<thead>
<tr>
<th>$w_d$</th>
<th>Type</th>
<th>Increase (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>315-360, 0-45</td>
<td>Head wind</td>
<td>4.0</td>
</tr>
<tr>
<td>45-135, 225-315</td>
<td>Side wind</td>
<td>2.0</td>
</tr>
<tr>
<td>135-225</td>
<td>Tail wind</td>
<td>1.0</td>
</tr>
</tbody>
</table>

5.3.4 Example:
The ship is running at 18 knots. The current is 1 knot along the direction of the ship. The wind blows 4 Beaufort points straight against the starboard side of the ship ($w_d=90$). The water depth $d$ is 15 meters below the keel. The speed through water is then $18 - 1 = 17$ knots. Table 1 shows the fuel consumption, before the wind and limited depth effects have been taken into account, at 1300 liters/hour. Table 2 shows the fuel consumption increase by 10%, caused by the water depth. Table 3 gives a $4*2% = 8\%$ increase due to the side wind. The total estimated fuel consumption in the example is therefore $1300 * 1.10 * 1.08 = 1544$ liters/hour.

5.4 Formulating the Optimization Problem

The route is divided into $n$ parts, each having a constant depth, wind, and current conditions. $n$ is typically between 2 and 40. Each part is denoted "route leg" or just "leg". For each leg $i$, the following data is available:

- Length $s_i$ (nautical miles);
- Longitudinal current component $c_{t,i}$, (unit: knots) parallel to the ship’s direction
- Transversal current component $c_{t,i}$, (unit: knots) perpendicular to the ship’s direction
- Wind strength $w_i$ (Beaufort)
- Relative wind direction $w_{d,i}$, (0-360) (see Table 3);
- Water depth $d_i$ (meters) below keel
- Minimum allowed bottom track speed $x_{min,i}$ (knots);
- Maximum allowed bottom track speed $x_{max,i}$ (knots)

For the route as a whole, the following data is provided:

- Total travel time $T$ (unit: hours);

The ship's speed relative to the ground (also called bottom track speed) is given in knots (1 knot equals 1 nautical mile per hour). The speed relative to the ground, $x_i$, is related to the speed through water, $x_{w,i}$ (also called water track speed) and the current components $c_{t,i}$ and $c_{l,i}$ according to:

$$x_{w,i} = \sqrt{c_{t,i}^2 + (x_i - c_{l,i})^2}.$$ \hspace{1cm} (6)

In other words: the water track speed $x_{w,i}$ is the vector difference between the bottom track speed and current vector. The fuel consumption (liters) on leg $i$ is denoted $C_i$ and is a function of speed over ground $x_i$ and the properties of leg $i$; the length $s_i$, the current ($c_{t,i}$, $c_{l,i}$), the wind strength $w_i$, the wind direction $w_{d,i}$, and the water depth $d_i$:

$$C_i = \frac{s_i}{x_i} F(x_{w,i})(1 + D(x_i,d_i)/100)(1 + W(w_i,w_{d,i})/100)$$ \hspace{1cm} (7)
where $F$, $D$, and $W$ are given by Tables 1, 2 and 3 respectively. The factor $\frac{s_i}{x_i}$ is the time (hours) that the ship is on route leg $i$.

For the optimization algorithm it is practical to express the fuel consumption as a function of the bottom track speed $x$. We therefore define the fuel consumption (liters) on leg $i$ for $x$ knots bottom track speed as

$$ C(x, i) = \frac{s_i}{x} F(x_w)(1 + D(x, d_i)/100)(1 + W(w_i, w_{d,i})/100) $$  \hspace{1cm} (8)

where

$$ x_w = \sqrt{c_{t,i}^2 + (x - c_{t,i})^2}. $$  \hspace{1cm} (9)

The vector $x$ is defined as $(x_1, x_2, ..., x_n)$, i.e. the unknown bottom track speeds on the $n$ legs. The total fuel consumption for a voyage is given by:

$$ \Phi(x) = \sum_{i=1}^{n} C(x_i, i). $$  \hspace{1cm} (10)

The objective of the optimization is to find the speed vector $x$ that minimizes $\Phi(x)$.

**5.4.1 Constraints**

As constraints in the optimization of $\Phi(x)$ we have:

1. $\sum_{i=1}^{n} \frac{s_i}{x_i} = T$. I.e. the ship has to arrive on time

2. $x_{\text{min}} \leq x_i \leq x_{\text{max}}, \forall i$. These constraints can be used to define speed limits on parts of the route, and also to set the available speed register for the ship.

$\Phi(x)$ should now be minimized with respect to $x$ under the above mentioned constraints.

**5.4.2 Start Value Algorithms**

As start value for $x$, three methods have been considered:

1. Assign an equal speed to all legs. I.e.: $x_i = \sum_{j=1}^{n} \frac{s_j}{T}, \forall i$. This method hardly needs any calculations, but on the other hand does not take current, wind or water depth into account.

2. Compute one value $x_w$ for the speed through water, same for all legs, that makes the ship arrive on time (i.e. $x$ fulfills constraint 1 above). This means that legs with a current are run at a lower speed (through water) than legs with the current along. I.e.: Compute a value $x_w$ that solves

$$ \sum_{i=1}^{n} \frac{s_i}{x_i} = T $$  \hspace{1cm} (11),

$$ x_w = \sqrt{c_{t,i}^2 + (x_i - c_{t,i})^2}, \forall i. $$  \hspace{1cm} (12)

Expression 12 assigns values to all bottom track speeds $x_i$ in such a way, that the water track speed becomes equal in all legs. If constraint 2 above out rules the necessary bottom track speed for a leg, assign the relevant end point in the constraint to $x_i$. This method gives an $x$ vector that compensates for the current, but not for the water depth $d$ or the wind $w$. 

332
3. Compute a fuel consumption \( \lambda \) (liters/hour) that, if used on all legs on the route, makes the ship arrive on time (fulfills constraint 1 above). This means that legs with a counter current are run at a lower speed over ground than legs, in which the ship runs with the current. It also means that legs with a heavy load due to shallow waters and/or wind are run more slowly than other legs. The algorithm involves solving two nested equations:

4. Find the fuel consumption \( \lambda \) (liters/hour) that solves:

\[
\sum_{i=1}^{n} \frac{s_i}{y_i(\lambda)} = T \tag{13}
\]

where \( y_i(\lambda) \) is the bottom track speed achieved on leg \( i \) if the fuel consumption is \( \lambda \) liters/hour. I.e., Each \( y_i(\lambda), \forall i \) has to satisfy

\[
C(y_i(\lambda), i) \cdot \frac{y_i(\lambda)}{s_i} = \lambda \tag{14}
\]

where \( C \) is given by equation 8.

Equation 13 is solved by the secant method. Each term in the sum requires solving equation 14 for a particular value on \( \lambda \). This is also done by the secant method. If constraint 2 above out rules the computed bottom track speed for a leg \( i \), the relevant end point in the constraint to \( x_i \) is assigned to \( y_i(\lambda) \).

Start value algorithm 3 gives an \( \mathbf{x} \) vector most often very close to the optimum for \( \Phi(\mathbf{x}) \), and can actually be used to compute the speeds on the different legs on the ship’s route. Further attempts to improve the reached optimum can be found in Dotzauer, Rosendahl (1995), where a number of optimization routines are applied to the problem. The tables with models are approximated with continous functions so the derivatives can be computed analytically. Both quasi-Newton and conjugated gradient methods are applied together with a variant of Fletcher’s line searching algorithm Fletcher (1987) and also with a golden section search algorithm. The combination of quasi-Newton and Fletcher’s line searching gives best results, but a general conclusion is that start value algorithm 3 most often give a good enough solution vector, and definitely much faster. Therefore, the route planning module uses this algorithm to compute optimal bottom track speed values for each leg in the route plan. The optimization is repeated at certain interval and also when new data for current, wind or depth is input during the voyage.

5.5 Using the Optimized Route

The input to the optimization consists of positions for the legs in the route. The following data is also given for each leg:

- **CURRENT** - Strength \( c \) and absolute direction \( c_{dir} \) of current. Unit: knots.

- **WIND** - Strength \( w \) and absolute direction \( w_{dir} \) of wind. Unit: Beaufort.

- **DEPTH** - Mean water depth below the keel. Unit: meters.

\( c_{dir} \) and \( w_{dir} \) are absolute values, and are entered as any of the following abbreviations: N. S. W, E, NE, NW, NNW, NNE, SE, SW, SSE, SSW, ENE, ESE, WNW, WSW. The relative directions are computed automatically by the system, depending on the ship’s actual course at each moment. Additional inputs are the ship’s mean draft and required departure/arrival times. The route optimizer computes the bottom track speeds \( x_i, \forall i \), that minimize the total fuel consumption for the voyage. The arrival time is always kept as requested. The route plan
shown in Table 4 has been computed with algorithm 3 described above. The set values $x$ for speed are automatically executed as POSITION/ARRIVAL commands (see Section 4.3). The speed control is combined with the dynamic limits for fuel consumption (see Section 4.2). In this way the engines are controlled in a smooth and economical way throughout the route. The route is automatically re-optimized every 10 minutes or when new current or wind are entered.

6 Conclusion

The propulsion of ships offer many interesting and challenging control and optimization problems. The control problems are characterized by high-time constants and noisy sensor signals. For reasons of robustness and generality, simple and intuitive solutions are often to be preferred. Furthermore, the noisy and time-affected nature of the problem makes the search for global and exact optima pointless, and even sub-optimal, if it involves a slower system with a higher risk for volatile behavior.

We have successfully implemented a number of control systems that aim at lowering the fuel consumption by optimizing control. The systems have worked particularly well for vessels with a wide speed-control range. This gives room for intelligent route planning, which really makes a difference for the total fuel consumption. The pitch optimization has worked best for older ships, where the initial rpm/pitch combinations are far from optimal. Newer ships have partly recognized the importance of having correct rpm and pitch at varying running conditions, and allow less room for a separate optimizer such as the presented system. The general trend in bridge equipment has, for a number of years, been integrated systems, where the same manufacturer delivers integrated equipment for many bridge functions. Along this trend, some radar manufacturers are offering primitive route planning functions and speed control functions as options in their systems. Also, advanced electronic sea chart systems are likely to include more and more route planning options in the future. Advanced optimizing functions, such as the ones described in this report, have still not been implemented in other products, to the authors knowledge.

7 Acknowledgments

I wish to acknowledge the invaluable domain knowledge and general help during the development from Göran Ekefsors and Gösta Kjellberg

References


Table 4: Optimized route plan for the route Göteborg-Kiel. The $x$ values are optimized speed (bottom track) values compensated for the varying depth values and weather conditions on the 31 legs. The $x$ values are used as set values for the speed control along the route.

**SEAPACER PC VOYAGE CONDITIONS SET UP**

**MS Emmaräng**

**Voyage conditions 1**

**Route number 1: Göteborg - Kiel**

**Mean draft: 6.3 meters**

<table>
<thead>
<tr>
<th>leg</th>
<th>Name</th>
<th>$c$</th>
<th>$c_{dir}$</th>
<th>$w$</th>
<th>$w_{dir}$</th>
<th>Depth</th>
<th>$ME$</th>
<th>$x$</th>
<th>$x_{nc}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KAJ GÖTEBOURG</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>8</td>
<td>1</td>
<td>7.00</td>
<td>7.00</td>
</tr>
<tr>
<td>2</td>
<td>Göteborgsgrund</td>
<td>0.0</td>
<td>S</td>
<td>3.0</td>
<td>SW</td>
<td>10</td>
<td>3</td>
<td>13.00</td>
<td>13.00</td>
</tr>
<tr>
<td>3</td>
<td>Brandnäs</td>
<td>0.0</td>
<td>S</td>
<td>3.0</td>
<td>SW</td>
<td>22</td>
<td>4</td>
<td>17.81</td>
<td>17.81</td>
</tr>
<tr>
<td>4</td>
<td>Trubaduren</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>40</td>
<td>4</td>
<td>19.13</td>
<td>19.13</td>
</tr>
<tr>
<td>5</td>
<td>Vangardsgrund</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>45</td>
<td>4</td>
<td>19.13</td>
<td>19.13</td>
</tr>
<tr>
<td>6</td>
<td>No. 4</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>30</td>
<td>4</td>
<td>18.63</td>
<td>18.63</td>
</tr>
<tr>
<td>7</td>
<td>Anholt Knob</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>30</td>
<td>4</td>
<td>18.49</td>
<td>18.49</td>
</tr>
<tr>
<td>8</td>
<td>No. 7</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>15</td>
<td>4</td>
<td>17.32</td>
<td>17.32</td>
</tr>
<tr>
<td>9</td>
<td>No. 11</td>
<td>0.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>18</td>
<td>4</td>
<td>17.52</td>
<td>17.52</td>
</tr>
<tr>
<td>10</td>
<td>SNR</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>22</td>
<td>4</td>
<td>17.08</td>
<td>17.94</td>
</tr>
<tr>
<td>11</td>
<td>No. 15</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>16</td>
<td>4</td>
<td>16.75</td>
<td>17.54</td>
</tr>
<tr>
<td>12</td>
<td>No. 20</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>20</td>
<td>4</td>
<td>16.97</td>
<td>17.96</td>
</tr>
<tr>
<td>13</td>
<td>No. 23</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>22</td>
<td>4</td>
<td>17.15</td>
<td>18.09</td>
</tr>
<tr>
<td>14</td>
<td>No. 25</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>16</td>
<td>4</td>
<td>17.17</td>
<td>18.00</td>
</tr>
<tr>
<td>15</td>
<td>No. 26</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>24</td>
<td>4</td>
<td>17.24</td>
<td>18.24</td>
</tr>
<tr>
<td>16</td>
<td>No. 28</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>30</td>
<td>4</td>
<td>18.16</td>
<td>19.08</td>
</tr>
<tr>
<td>17</td>
<td>Vengeance grund</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>30</td>
<td>4</td>
<td>17.72</td>
<td>18.72</td>
</tr>
<tr>
<td>18</td>
<td>Agersö</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>20</td>
<td>4</td>
<td>16.96</td>
<td>17.80</td>
</tr>
<tr>
<td>19</td>
<td>No. 2</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>20</td>
<td>4</td>
<td>16.97</td>
<td>17.96</td>
</tr>
<tr>
<td>20</td>
<td>No. 3</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>20</td>
<td>4</td>
<td>17.01</td>
<td>17.95</td>
</tr>
<tr>
<td>21</td>
<td>DW 51</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>20</td>
<td>4</td>
<td>17.18</td>
<td>17.76</td>
</tr>
<tr>
<td>22</td>
<td>DW 53</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>20</td>
<td>4</td>
<td>16.89</td>
<td>17.82</td>
</tr>
<tr>
<td>23</td>
<td>DW 55</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>30</td>
<td>4</td>
<td>17.72</td>
<td>18.72</td>
</tr>
<tr>
<td>24</td>
<td>DW 57</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>18</td>
<td>4</td>
<td>16.77</td>
<td>17.69</td>
</tr>
<tr>
<td>25</td>
<td>DW 59</td>
<td>1.0</td>
<td>N</td>
<td>3.0</td>
<td>SW</td>
<td>18</td>
<td>4</td>
<td>16.86</td>
<td>18.83</td>
</tr>
<tr>
<td>26</td>
<td>DW 61</td>
<td>1.0</td>
<td>W</td>
<td>3.0</td>
<td>SW</td>
<td>14</td>
<td>4</td>
<td>17.88</td>
<td>17.10</td>
</tr>
<tr>
<td>27</td>
<td>KO 2</td>
<td>1.0</td>
<td>W</td>
<td>3.0</td>
<td>SW</td>
<td>10</td>
<td>4</td>
<td>17.54</td>
<td>16.65</td>
</tr>
<tr>
<td>28</td>
<td>Kieler Förde</td>
<td>1.0</td>
<td>W</td>
<td>3.0</td>
<td>SW</td>
<td>10</td>
<td>4</td>
<td>17.20</td>
<td>16.76</td>
</tr>
<tr>
<td>29</td>
<td>Fartbojen</td>
<td>1.0</td>
<td>W</td>
<td>3.0</td>
<td>SW</td>
<td>10</td>
<td>3</td>
<td>11.00</td>
<td>10.77</td>
</tr>
<tr>
<td>30</td>
<td>Friedrichsort</td>
<td>1.0</td>
<td>W</td>
<td>3.0</td>
<td>SW</td>
<td>8</td>
<td>1</td>
<td>8.00</td>
<td>7.50</td>
</tr>
<tr>
<td>31</td>
<td>Kiel 11 Reede</td>
<td>1.0</td>
<td>W</td>
<td>3.0</td>
<td>SW</td>
<td>7</td>
<td>1</td>
<td>7.00</td>
<td>6.72</td>
</tr>
<tr>
<td>32</td>
<td>Scweedanek</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>End of route plan</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimization data:
Mean: 3078 l/h Total: 40320 litres Total dist: 234.6 nm Total time: 14:00
Successful optimization
Cyber-Ships – Science Fiction and Reality

Volker Bertram, ENSIETA, Brest/France, bertram@waves.insean.it

Abstract

Artificial Intelligence technologies are discussed particularly for combatants. The nautical tasks could be largely automated. Commercial expert systems for automatic navigation including collision and grounding avoidance are on the market. Natural language interfaces and gestures allow better man-machine interaction. Machine vision is seen as a frontier technology to enable further automation. Humanoid robots appear rather useless for navy ship operation, but micro-robots and simple fixed robots may be used for assorted tasks. Virtual reality is predominantly attractive as training tool. Transponders now allow automatic ship-ship and ship-shore communication, but may also be implanted to humans as a convenient “key” to interact with computers.

1. Introduction

Computers take over controls in cars, trains and planes. Bertram (2002) quotes intelligent highways, self-driving ‘seeing’ cars, automatic trains and metros, automatic planes, notably the highly autonomous, intelligent UCAsVs (Uninhabited Combat Air Vehicles), Fig.1, now under development and testing in the USA. In view of these spectacular projects, naturally questions arise if there are comparable developments for ships. In the last 150 years, transatlantic cargo ships have reduced necessary crews from 250 to 15. In view of the technical progress and the shortage of qualified crews, Scandinavian countries advocate crews with just 6 men on certain routes. Crew reduction of up to 90% is sought in specifications for future destroyers of the US navy.

Of course, the ultimate in ship automation would of course be the unmanned ship. Unmanned ships have been envisioned for at least three decades now as reviewed by Kasai and Bertram (1996), Bertram (1999,2002). In general, the proposed unmanned ship concepts found in the literature can be classified into:

• ‘Shore Captain’ concept
  The control system is transferred ashore. The ship retains only a largely self-regulating propulsion plant together with the equipment needed for reception, transmission, and decoding of the control signals received from the shore and supervision of onboard systems.

• ‘Captain Computer’ concept
  The ship is equipped with sufficient hardware and software to perform all tasks and decisions autonomously using Artificial Intelligence.

• ‘Master/Slave’ concept
  Convoys of unmanned ‘slave’ ships are remote-controlled from a highly automated ‘master’ escort ship. While such a concept poses the least technical problems, one large ship with the same crew as the master ship would be simpler and more economical in most applications. However, this concept makes sense if explosives or other dangerous cargo shall be transported apart from the crew. The concept is investigated by navies for mine-sweeping and related projects.

In practice, usually a mix of local and remote control will be employed with redundancy for vital systems in case the communication link breaks down or local systems fail. Even if such a system could cope with all normal conditions, the repair of defects is unlikely to be handled satisfactorily. But failures in the ship machinery occur now about once in 100 hours. Approaches to increase reliability include expert systems for fault-diagnosis and maintenance, as surveyed by Kaeding and Bertram (1997), Bertram (1998). All large diesel manufacturers offer such systems by now.

Drastically reduced crews will mean far less time for predictive maintenance. This will require new design paradigms and possibly more harbor time for maintenance for navy ships. Despite all expected improvements of engine reliability, onboard maintenance and occasional fault repair will most proba-
bly still characterize future ship operation. The issue at present is not the unmanned ship, it is the ‘intelligent combatant’ with drastically reduced crews. Both concepts “unmanned” and “autonomic” share the task for extending automation. Artificial Intelligence technologies and modern telecommunication are expected to be the key to this envisioned further progress in ship automation. Individual techniques and applications are discussed and highlighted in examples in the following.

Fig.1: Unmanned combat planes (UCAV)

The issue of automation evokes traditionally emotions. Cyber-ships are bound to evoke strong emotions in the traditional seafaring community. A rational, engineering approach may help making the discussion less emotional. A rational approach to ship automation appears to be:

- Machines should do what machines do better than humans.
- Humans should do what humans do better than machines.
- Machines should support humans.

The technological progress will (or should) shift more and more tasks from humans to machines. However, navies have their particular inertia and combatants are designed often 10 years before entering service. As a result, more tasks than necessary are performed by humans and efforts concentrate on making humans machine-like rather than shifting the tasks to machines.

Machines are superior to humans in the following aspects, Schneiderman (1992): perform repetitive preprogrammed actions reliably; exert great, highly controlled physical force; monitor pre-specified events, especially infrequent; perform several activities simultaneously; count/measure physical quantities; make rapid and consistent responses to input signals; operate in life-threatening environment.

Humans are (at present) superior to machines in the following aspects: act in unanticipated and novel situations (common sense); reason inductively; generalize from observations; take actions for self repairing; interact socially with other humans; perform acts of fine motorics. Originally, the list of human superiority included detecting, especially using vision. In view of the recent progress of machine vision, this has to be modified. Machines appear by now to have sometimes better, sometimes worse pattern recognition capabilities, depending on the particular application.

In order to progress with automation, one should review all crew members asking:
- What are the functions of this crew member?
- Can functions be performed on shore or via telecommunication from shore?
- Can functions be performed by machine (computer) as well or better?

Such an analysis should include both large task packages (macro-automation) and small tasks that are short, but performed very often, e.g. retrieving certain information, logging into systems etc. (micro
automation). Macro automation focuses on making certain people on board obsolete, micro automation making the remaining more efficient. It is important to break down functions to sufficient detail to reveal potential reduction, e.g. to separate knowledge in diagnosis and treatment/repair. Diagnosis may require expert knowledge, but can be performed often via tele-presence, therapy may involve simple manual tasks to be performed by anybody given proper instructions.

2. Lessons of the USS Yorktown

Even if a reasonable approach to automation is taken with prototyping in steps and field-testing, there will opposition, driven by irrational technological phobias or by rational business interest, as the case of the USS “Yorktown” illustrates. In 1995, the USS “Yorktown” was selected as platform for the “smart ship” ideas developed in the 1990s, www.chinfo.navy.mil/navpalib/allhands/ah0997/pg20.html. Several automatic systems helped reducing crew size drastically with subsequent need to re-design work assignments on-board, e.g. with flexible damage control teams instead of traditional general quarters concepts. Lookouts were eliminated as one realized that the lookouts rarely spotted a contact before the signalman or officer of the deck. Maintenance was found as a bottle-neck in crew size until (installed, but unused) engine room automation was used.

In 1998, a major computer crash on board the “Yorktown” brought the ship back into the headlines, www.sciam.com/130.94.24.217/1998/1198issue/1198techbus2.html. After a crew member mistakenly entered a zero in a data field of an application, the computer system proceeded to divide another quantity by that zero. The operation caused a buffer overflow and the error eventually brought down the ship’s propulsion system. The “Yorktown” was dead in the water for more than two hours. The incident provided ample ammunition for the critics of automation. However, the “Yorktown” can still be seen as a success story for smart ships. It implemented part of the saving potential between traditional ship operation and future ship operation. Teething problems are to be expected in a technology demonstrator.

Automation will progress in steps. With anything new, there is a period of acceptance. Initially, there will be back-ups for manual control, human confirmation required, etc. until there is sufficient confidence in the technology that the back-ups are considered more of a hindrance. This may then again open the door for better performance as it is suspected that in some cases manual back-ups or semi-automatic systems lead to more complex designs which are more error-prone than fully automatic designs could be.

3. Artificial Intelligence offers key technologies for future combatants

The tasks involved for further ship automation and faster threat response share many characteristics with the quest for safer and more effective cockpits for airplanes (the following was found on a sub-page of www.unibw-muenchen.de/campus/LRT/LRT13 for flight deck automation):

- to understand the abstract goals of a (flight) mission
- to assess needed information about mission, [ship or] aircraft environment and [ship or] aircraft system
- to interpret the (flight) situation in the light of the mission
- to detect pilots’ [or captains’] intent and possible errors
- to support necessary re-planning and decision making
- to know which information the crew needs and how to present it to the crew in the most effective way

The above tasks involve knowledge processing, improved man-machine interaction, and “intelligent” sensor interpretation. Many of these tasks will involve techniques commonly grouped under the label “Artificial Intelligence” (AI). The scope of AI is not clearly defined. In its broadest sense, AI is concerned with the investigation and simulation of human intelligence with the ambition to replicate the processes in machines. A (not exhaustive) list of branches of AI encompasses knowledge-based systems / expert systems/case-based reasoning/Bayesian networks, natural language processing, machine
vision, robotics, machine learning, and artificial neural nets. Selected branches of AI and their potential or actual applications to (navy) ship operation will be discussed in the following.

3.1. Knowledge based systems and related techniques

Knowledge-based systems are arguably the most widely established branch of AI, at least in naval applications, Bertram (2000). The terms 'knowledge-based system' and 'expert system' are often used synonymously. Some reserve 'expert systems' to such knowledge-based systems incorporating expert heuristic knowledge not documented explicitly in books. Knowledge-based systems (KBS) are, as their name suggests, systems which use knowledge and reasoning to arrive at conclusions, http://best.me.berkeley.edu/~aagognino/me290m/s99. They differ from traditional data-processing computer programs in their method of operation. Conventional programs are also optimized for numeric-processing, whereas the knowledge-based system concentrates on the representation and manipulation of information as symbols. Another noteworthy feature of KBS is their suitability or large and complex problem solution characterized by inexact, incomplete and uncertain information. Their structure includes an explicit body of embedded knowledge and a separate, identifiable inference mechanism. Using these facilities, the KBS builder is able to construct a mechanism capable of 'mimicking' human reasoning (the inference mechanism or inference engine), and the knowledge engineer is able to elicit and code expert knowledge which the inference mechanism may use to provide solutions to problems in a similar fashion to a comparable human expert. In spite of all this, however, knowledge-based systems are merely computer programs which have been written in a different way, in a deliberate attempt to isolate the various components of human (expert) problem-solving. The isolation of the program flow directives which represent components of knowledge most often in the form of rules permits an explicit body of knowledge to be created and enlarged/modified in a way which would be difficult in conventional data-processing programs.

- Monitoring of machinery and ship

The monitoring of engines and the ship itself involves the automatic observation of a flood of data which has to be checked against acceptable or expected values. For the machinery, early detection of deviations from standard values is already used to support predictive maintenance and fault diagnosis. Similar tasks are involved in detecting fires or the risk of a collision in dense traffic. The individual tasks are simple and the amount of data and the need for constant vigilance make it clearly a task better handled by computers. The performance of diagnosis systems depends on the (sensor) input. E.g. for the risk of collision, ARPA's automatic target acquisition reliability is limited. Small ships/boats are sometimes not detected. Furthermore, ARPA cannot diagnose the type of ship, e.g. sailing ship, which is a vital information for certain rules of collision avoidance. Japanese attempts to use video cameras and pattern recognition in the late 1980s were not successful. The recent successes with 'seeing cars' described above may re-open the discussion about the feasibility of this approach. However, the problem is better solved by making transponders mandatory. Transponders would allow determination of ship types, detection of wooden or plastic boats and even special treatments for ships with hazardous cargo or ships with problems like blocked rudders.

- Advisory systems for maintenance and repair of engines and other systems

The monitoring involves just the detection of a problem. Increasingly, also decision support systems, often based on expert system technology, are used to advise the crew what to do if such a problem occurs. This may involve fault diagnosis for machinery coupled to an 'electronic manual' that tells the crew what to do to remove the problem, i.e. guide the repair. The trend is to make maintenance and trouble-spotting easier rather than avoiding faults at all cost. The system of the future will have a self-diagnosis function which instructs the operator how to repair the malfunction. This will drastically reduce time needed to find the reason for malfunction and allow multi-purpose crews to perform jobs now requiring experienced specialists. Similar systems for weapon systems have been occasionally reported.

339
- Collision avoidance

For collision (and grounding) avoidance, the analogous task is planning an avoidance route. In restricted waters, this should also include grounding avoidance. Collision-avoidance systems use expert systems to incorporate traffic rules and regulations, but also the experience of ship masters. An avoidance route is automatically selected usually based on the criteria of minimum collision risk, length of avoidance route, and steering action. In Japan ship trials with an automatic collision avoidance system was performed near the Bay of Tokyo in an area of dense traffic, e.g. Kasai and Bertram (1996). The ship steered safely in the congested sea traffic solving all collision risk problems. The avoidance judgment and actions appeared reasonable, even though crude compared to an experienced helmsman. Further refinements resulted in the commercial "SuperBridge" system installed for the first time in the 258,000 tdw tanker "Cosmo Delphinus". "SuperBridge" continuously monitors the dangers of grounding and collision on the basis of the electronic chart (checking for shallows) and radar/ARPA (detecting surrounding ships). The expert system determines secure avoidance route based on maritime traffic regulations and good seamanship practice. For legal reasons, "SuperBridge" is an advisory system requiring a confirmation of the system's decisions by the helmsman. By 2002, 14 systems were installed including 7 voice-controlled SuperBridge-X systems.

Over the past decade, the USA has developed and installed a number of comparably mature 'intelligent' navigational decision aids, http://maple.le moyne.edu/~grabowski/. The 'Exxon Valdez' accident triggered the development of the Shipboard Piloting Expert System (SPES) which was operated and tested on Exxon Shipping tankers since 1992. Since 1995, the experience gained was used to develop the Navigation and Piloting Expert System (NPES) for the San Francisco Bay as part of the SmartBridge program. A SmartBridge prototype was installed on the "Chevron Colorado", a 7000 tdw tanker, in 1997. By the end of the 1990s development of distributed intelligent piloting systems (DIPS) started, Grabowski (1996,1999), Sudendhar and Grabowski (1996). "These systems have grown from stand alone intelligent piloting aids to embedded intelligent systems within a distributed information system, i.e., ship and shore-based information systems. Originally, these decision aids focussed on enhancing the performance of individual vessels and pilots in the waterway. Currently, they also coordinate traffic interaction between multiple vessels on the waterway as well as distribute intelligent reasoning which underlies this coordination to all concerned parties on the waterway: to vessels currently on the waterway, to vessel traffic controllers facilitating the flow of traffic, and to vessels planning to be on the waterway in the near future," Grabowski (1996).

All collision-avoidance systems are advisory systems requiring a human confirmation of the system's decision. This appears a natural first step in the introduction of these systems. With growing confidence in the automatic processes, the adoption of 'unmanned' bridges at night and on open sea (probably with a 'watch' resting on the bridge only to get into action if alarmed by the system), and ultimately the use of such a system during all times also in congested waters with dense traffic seems feasible. Quite possibly, we will see de facto fully automatic operation long before regulations follows reality.

- Emergency response / Damage control

The conventional approach to damage control relies on human intervention under crisis conditions to integrate, evaluate and initiate actions. The contingency plans and emergency procedures are often distributed into several manuals like a ‘Damage Control Booklet’, ‘Bridge Procedures’, ‘Emergency Check Lists’ and ‘Ship Fire Fighting Manual’. The ‘booklet’ for damage control may typically comprise several hundred pages covering a wide range of possible cases. Information retrieval from each of these sources is time consuming and error-prone under stress. Expert systems have been developed to improve this situation. Expert systems may incorporate 'early failure' detection or event trending to establish 'pattern recognition'. More advanced systems cross-reference functionalities, e.g. fire fighting and ballasting ("What are the relative priorities between attacking a fire, drain off water from compartments or even flooding them to preserve ship stability").
Similar systems have been developed by advanced navies. Decision aids will allow rapid and remote reconfiguration of auxiliary systems in response to damaged ship scenarios. The systems know the particulars and characteristics of inbound missiles. These data along with characteristics of the ship and tactical scenarios are processed in a damage prediction model. Based on the model output, systems predicted to be damaged will be reconfigured or rerouted to minimize impact in the ship's ability to operate, Fig.2, Ditzio et al. (1995), Hoyle (2000). Due to sensor limitations and time constraints it is not always possible to make decisions with absolute certainty about what threats are being faced. Also, often conflicts occur such that weapons systems cannot fire at two or more different targets as ideally desired. New systems are able to resolve these conflicts and produce an optimal solution working around the physical constraints on the self defense assets in 'real-time'. Such systems incorporating uncertainty in Artificial Intelligence are under development, see also the next subsection on Bayesian Networks.

Both the UK Royal Navy and the US Navy have introduced first-generation combat advice systems, Scott (1995). These offer tactically correct defensive recommendations - derived from an embedded rule base - allowing the command to focus on the application of human reasoning and intuition. Meyrowitz (1999) reports that an initial collaborative data fusion system for the US navy has been constructed and employed in complex tactical simulations to perform situation assessment at multiple levels of data fusion in 1996. Further ahead, work is progressing on the development of more advanced automated planning and decision aids designed to support situation assessment, resource allocation and weapon coordination at both single-ship and force level. However, the currently prevalent view is that "the task of making tactical decisions in a naval context is too complex to be accomplished effectively by humans or computers alone", Kushnier et al. (1996). Instead, systems are developed where humans and computers work together and assist each other in doing what each does best.

Case-based reasoning (CBR) system are another form of knowledge-based systems. The principle is related to rule-based production systems. Instead of a knowledge base, there is a case base. Instead of an inference engine, there is a case-based reasoner employing a similarity function to select related cases, Aha et al. (1999). Conversational CBR are the most successful CBR technique and commercial shells are available. CBR systems have been investigated as an alternative, or at least adjunct, to rule-based reasoning, www.aic.nrl.navy.mil/~aha/cbr/ccbr-research.html. In 1995, the US navy's Fleet Technical Service Centers deployed an application of a conversational CBR for trouble-shooting a weapon system (MK41 vertical launch system). Subsequent research has focussed on simplifying the application of CBR and widening the applications. One of the results has been the NaCoDAE (Navy Conversational Decision Aids Environment) as a retrieval tool. However, the applications appear to be few and the technology less mature than rule-based expert systems.

3.2. Bayesian Networks

Bayesian networks can be regarded as a sub-branch of knowledge-based systems incorporating aspects of uncertainty and probability, www.cs.berkeley.edu/~murphyk/Bayes/bayes.html. There are several introductory textbooks, e.g. Jensen (1996). Bayesian networks are in principle simple diagrams that organize the knowledge in any given area by mapping out cause-and-effect relationships among key variables and encoding them with numbers that represent the extent to which one variable is likely to affect another. Programmed into computers, these systems can automatically generate optimal predictions or decisions even when key pieces of information are missing. Bayesian networks have long been an obscure sub-branch of mathematics, and only with the wide availability of sufficiently powerful computers in the 1980s Bayesian networks with enough variables to be useful in practical applications became feasible.

In the late 1970s and 1980s, the predominant approach to knowledge-based problems was based on expert systems representing the knowledge in if-then rules. These so-called production systems are still popular and quite widely used in ship operation and fault diagnosis systems. But these systems were time-consuming to develop, and problems involving uncertainties (which appeared in all cases where you could not answer all the computer's questions clearly) were not as easily handled as in
Bayesian nets. After some mathematical breakthroughs by Danish scientists and successful pilot applications by Prof. Judea Pearl (UCLA) in the late 1980s, Bayesian networks were perceived by an ever growing community of scientists as an efficient way to deal with lack or ambiguity of information. The real breakthrough for Bayesian networks happened when it became public that Microsoft saw it as a future technology and invested heavily into research and development of Bayesian network applications in the mid-1990s to support their software. Bayesian networks are reported to being used to develop next-generation user-friendly software interfaces.

Ray Rimey at the Univ. of Rochester/NY has combined Bayesian networks with machine vision for robots. A robot equipped with two video cameras resorted to Bayesian networks to extract relevant clues, set them in relation to each other and draw conclusions, (www.rimey.com/ray). Rimey selected the analysis of a dinner table as application, teaching the robot to analyze and make conclusions about different type of place settings. In principle, the problem is how to teach a computer to scan a scene and zero in on the most important information. Honeywell is reported to be interested in Rimey’s system to analyze infrared images taken by roving vehicles. For many problems, the world is too complex to enable a system to see every detail at all times and then act with sufficient speed. So the system must be selective in where to put its attention. Rimey’s work is a contribution to having automatically prioritize where to look and how to look. Bayesian networks have been applied to automated target recognition by Ulf Grenander and Anuj Srivastava, www.dam.brown.edu/mpic/atrcdrom.php3. Bayesian networks are also applied or proposed to ship applications. Scott Musman (Intelligent Systems Division at Integrated Management Services) has developed a Bayesian network for the US Navy that can identify enemy missiles, aircraft and vessels and recommend which weapons could be used most advantageously against incoming targets, (Ship Self Defense Tactics Engine), Musman and Lehner (1999). The work draws on early work to identify ships, Musman and Chang (1993), Musman et al. (1990,1993). Peter Friis Hansen (Danish Technical University) has applied Bayesian networks to risk analysis for solo watch keeping in ship operation, Hansen and Pedersen (1999), and maintenance scheduling connected to fatigue strength and crack propagation, Friis-Hansen (2000).

3.3. Natural language processing and other new Man-Machine Interfaces

The issue of communication between man and machine is crucial for progress in automation. Integrated bridges with one common interface increase user-friendliness and thus safety. Still, the officer of the bridge typically has to type in commands and view screens to interact with the machine. This can lead to stressful situations in one-man bridge operation. Such stressful situations can be analyzed in the risk-free environment of a ship simulator to derive recommendations for future bridge systems. The Japanese have done this and developed the navigation system SuperBridge-X based on natural language as a new element in human machine interfacing in ships. The master addresses the system by speaking (ordering changes in speed or course, changing displays on computers, etc.) and the system announces via a loudspeaker relevant information (confirmation of accepted orders, warnings and alarms, etc.). The voice-operated SuperBridge-X system allows in principle ‘no-touch’ operation of the ship, Yamamoto (1999). The advantages of keeping the ship master’s view free to monitor his environment are obvious. SuperBridge-X had 1998 a capability of approximately 80 announcements (for replies, warnings and alarms) and approximately 30 commands or inquiries. This suffices for the most important monitor and control functions in ship handling. Commands concern changes of course and speed and visual displays on the bridge. A typical control sequence may look like this:

Human: Course 5 degree starboard!
Computer: Course 5 degree starboard, OK?
Human: OK!
Computer: Course has been set 5 degree starboard.

The computer thus always repeats a command and waits for a confirmation before execution. Alternatively to voice confirmation, a key on a keyboard may be pressed. Alarms interrupt normal dialog sequences. All alarms are also displayed on a screen. The system is based on two microphones, one directly at the commanding officer and one in the room. Comparison of input signals to both micro-
phones allows to filter out the commands of the commanding officer. Back-ground noise and also conversation by other people on the bridge posed no problem to system in trials. The voice recognition is not tuned to one particular speaker and does thus not require retraining at each change of the shift. The system is so far based only on Japanese as language, but English language are commercially available and should be relatively easily connected to the rest of the system.

The advantages of voice-operation are obvious: The hands and eyes are free for other tasks, e.g. watching the traffic and checking sea charts. The interaction with the bridge system then becomes more like the traditional way of interacting with other humans on the bridge. Speech-control is important when hands are otherwise busy (controlling e.g. an object) or when vision is impaired (e.g. wearing a virtual reality helmet, as demonstrated in the Virtual Reality ship familiarization system of Wauchope et al. (2003)). It is also a useful technology to reduce space for keyboards.

People have cognitive limitations that make them sensitive to interruption. These limitations can cause people to make mistakes when interrupted. This is particularly an issue for navy ship operations in combat situations. Future man-machine interfaces will therefore use knowledge about the importance of an information and the importance of a current activity of a user to decide whether to interrupt or “leave a message”. The HAIL project (Human Alerting and Interruption Logistics) points in this direction, www.atc.nrl.navy.mil/hail/index.html.

High noise levels (e.g. on navy ships in battle situations) degrade human listening performance and are likely to affect even more severely automatic recognition systems. Research has been devoted to improve performance of navy speech recognizers in noisy environments. Alternatively, other modes of communications may be employed. Gestures may be used to communicate with computers and robots. Siemens and IBM develop virtual keyboards: The computer traces hand motions of users via a small camera. Users can either unroll a plastic-foil template with a keyboard layout, or tap on screens, or use a laser-projected virtual keyboard. The user may also interact with programs, e.g. turning or shifting objects by corresponding hand motions. This saves weight and space and allows hygienic and indestructible keyboards. It supports also extremely small, portable computers. Communication by gestures is also important as an alternative to speech in very noisy environments or in situations where silence is of tactical importance. Meyrowitz (1999) reports a combined natural language and gestural interface to a mobile robot. Ambiguities in language directions are resolved by gesture understanding, and ambiguities in gesture are resolved by language understanding.

3.4. Robotics

The Encyclopaedia Britannica defines a robot as follows: “Derived from the Czech word robit ("work"), it passed into popular use after 1923 to describe [...] mechanical devices so ingenious as to be almost human.” Naturally one may then be tempted to just substitute human crew members by “ingenious” and “almost human” machines in a quest to reduce crew size. Indeed, Katagi and Hashimoto (1990) predicted ships with robots with sensors, ability of movement and "judgment similar to or better than those of man". At the beginning of the 21st century, this appears still unlikely even though robotics develop rapidly.

Humanoid robots are envisioned for a variety of application domains including health care, domestic services, and entertainment. Humanoid robots with sensor (vision, hearing, and even tactile sensing) are under development worldwide and attract considerable public and media attention. Research on humanoid robots is particularly advanced at the MIT, www.ai.mit.edu, and in Japan, Fig.3, www.honda-p3.com, www.world.honda.com. See Bertram (2002) for more details. Humanoid robots are still in an infant stage. (Three-year old humans out-perform so far all humanoid robots in terms of walking capability and sensor capability, often also in terms of strength.) But for most tasks robot do not have to resemble humans. However, there is a general trend in robotics towards ‘seeing’ robots and ‘thinking’ robots.
At the Navy Center for Applied Research in Artificial Intelligence (NCARAI), several projects were concerned with robotics for navy applications, Meyrowitz (2000), www.aic.nrl.navy.mil. A variety of robotic behavior has been investigated: obstacle avoidance (including fields of floating mines), path planning, tracking, and cooperative mapping for flocks of robots. Robots can learn behavior in virtual worlds and improved behaviors observed in simulation carry over to improved behaviors when the software is placed on real mobile robots, Schultz et al. (1996). Movable robots do not always have to have advanced sensor capabilities as obtained in the humanoid robot research. Robots for cleaning floors, cleaning swimming pools and mowing lawns orient themselves roughly, e.g. detecting obstacles using ultra-sonic sensors.

Robots can be built in different dimension. Robots have been developed to search for survivors of earthquakes. One such model from Japan that resembles a snake and can crawl through tight spaces, www.snakerobots.com. Other robots look rather like insects. Sandia National Laboratories have built a mini-robot that “parks on a dime and turns on a nickel”. The robot moves on caterpillars and is equipped with a thermo-sensor, www.sandia.gov/media/NewsRel/NR2001/minirobot.htm. In future, the robot shall be equipped with mini-camera, microphone, and chemical sensors. Such a robot or flocks of such robots could e.g. inspect pipes, etc.

Robots with sufficient agility, sensor capability, and robustness to replace human work will not be available for some time to come and then the first such robots would probably be more expensive than humans. Mobile robots are usually weak, fragile, and need power sources. Realistically, robotics seems to be more interesting for land-based applications and for remote operating vehicles than for the operation of ships. However, research is active and the technology should be monitored and promoted. Robots may already be used for tasks like mine hunting and mine removing, or reconnaissance tasks, opening new operational aspects for navies. A practical on-board application of present technology could be e.g. "robotic arms equipped with binocular viewers will provide virtual presence in machinery spaces", Ditizio et al. (1995), for fire fighting.

3.5. Machine vision and neural nets

A decade ago, Japanese researchers failed in their attempt to employ machine vision for detecting dangers of collisions for ships. Machine vision is a field that has progressed considerable, and while the problem of collision avoidance seems to be solved by now using other sensors, machine vision offers many options in improving performance of machines.
Machine vision is interesting in combination with robotics. “Among various sensors to be used in conjunction with robot control, vision has a number of advantages: it is low cost, fast, […]”, Lamirey et al. (2000). Following this philosophy, visually guided robots to weld ship structures have been developed to prototype demonstrators, www.inrialpes.fr/VIGOR. INRIA Grenoble, www.inrialpes.fr, investigated particularly techniques where the robots carry their own cameras and alternatively techniques where the robot uses images supplied from external cameras in a room. The general background is that one may plan e.g. a robot track off-line based on a CAD model, but if there are any changes between CAD model and real world (e.g. now there is an obstacle in the path), the robot should be able to detect this change and its relevance to the initial path planning. In short: Autonomously moving robots for our applications need some sort of vision.

Autonomous robots hold a CAD map of ship and may use landmarks such as walls or stiffeners for navigation. The robot assumes a rough position and matches the landmarks of its CAD map to those detected by the vision system. “The main problems are a changing background and high computational demands. For example, a space application where the background is dark and the object consists of parts of different surface characteristics, requires dedicated hardware to run at frame rate […]. Probably the most successful system that uses vision to control a mechanism is the automatic car and air-vehicle approach using dynamic vision, Fuerst and Dickmanns (1999). It integrates the dynamic aspects of a continuously operating system and image data to update the model description of the world.”, Vincze et al. (2000).

The European research project ROBVISION has developed vision systems to allow guiding a walking robot through a ship structure, e.g. a double bottom, Fig.4, Vincze et al. (2000). A welding robot would thus be able to orient itself inside ship structures usually difficult to assess for humans, using a CAD model of the structure (‘map’) and his own vision. One of the objectives is a visual processing robust to deviations in parts and environmental conditions. To achieve this goal a technique is developed that integrates different cues of images to obtain confidence of the measurement result. The project develops an integrated vision system capable of providing adequate information to guide an advanced robotic vehicle through a complex structure. The final demonstration will see the walking robot enter and climb the vessel structure, robvision.infa.tuwien.ac.at/rvision.htm.

NCARAI has combined range-based vision with intensity-based vision using tripod operators. The system recognizes an object among 25 similar shapes in a cluttered scene, with very few false positives, Meyrowitz (1999). A typical time to find a given shape was tens of milliseconds in 1994.

Machine vision may employ neural network techniques to learn to identify patterns, Ripley (1996), Hinton (1992), www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html. This has been used for a variety of applications, both civilian and military. The pattern recognition has advanced much beyond the initial primitive applications. Commercial systems to identify faces based on video input are now available e.g. for security systems, www.miros.com.

Fig.5: Original infra-red image and extracted aircraft using NRL developed automated technique
Machines may employ radar images or infra-red images as well as the usual light wave length perceived by the human eye. Meyrowitz (1999) reports research of the Naval Research Laboratory (NRL) on pattern recognition of aircraft approaching aircraft carriers, employing neural networks to identify aircraft types from infra-red images, Fig.5: “Additional neural network research has recently yielded innovative techniques for automatically extracting objects of interest from their background (in infra-red images, for instance), and for training networks so that they are capable of rejecting deficient input data in images. This technology provides a solution to the problem of reliably but passively recognizing aircraft approaching carriers or approaching more general battle spaces. The trained networks are able to avoid processing input before aircraft are close enough for image classification, and can avoid confusing noise such as cloud formations with actual aircraft.”

3.6. Virtual Reality

Virtual reality (VR) initially referred to immersive technologies, Beier (2000), www-VRL.umich.edu. Today, the meaning of VR has broadened and includes semi-immersive and non-immersive techniques. VR models require an underlying CAD model of their world which then offers fly-through or walk-through capabilities. The VR models may be viewed in a variety of ways from CAVEs to plain PC screens. The resulting illusion of being fully immersed on an artificial world can be quite convincing. However, increased reality and model size comes at a price. Pragmatic applications have just the necessary level of detail to allow sufficiently fast responses on common hardware platforms.

There is a wide scope of VR applications, potential and implemented. VR can be used as training tool, both as a “poor-man’s” ship simulator (with underlying maneuvering model), and as a training tool to familiarize new crews without interfering with operations and to train damage control personnel, Wauchope et al. (2003). The NRL has developed InterShip, a VR tool to familiarize personnel with the layout of a ship, Fig.6. InterShip combines VR techniques with a knowledge-based route planner (shows how to get from one compartment to another) and speech control (user can e.g. open door by command; user can query system for information, e.g. invoking route planner.) Using the head-mounted display and a hand-held joystick, users could walk through portions of a ship and ask questions about compartment names, numbers and locations. (“What compartment is this?”; “Which deck is the communications center on?”)

![Fig.6: VR view of glove atavar opening door, Source: NRL Washington](image1)

![Fig.7: Solar powered chip placed on 1 cent coin, www.deafblind.com/implant.html](image2)

NRL has investigated employing this technique to improve performance of firefighters, Tate et al. (1995, 1997), www.chemistry.nrl.navy.mil/damagecontrol/vr.html. The virtual environment included a dynamically generated virtual fire made up of approximately 500 polygons. Using a mixture of physically based modeling and fractal techniques, the fire changed color and transparency levels to simulate the appearance of real flames. The density of simulated smoke varied with distance to the simulated fire and could be changed by operator control. There was a measurable improvement in the performance of firefighters that used VR training over firefighters without such training. VR trained firefigh-
ters made fewer wrong turns and reached the fire faster than untrained firefighters. Firefighters might also benefit from another VR application, not yet implemented: Firefighters might have a virtual view of the ship and fire projected on a screen in their helmets blocking out all smoke. The fire may be projected based on infrared sensors. Thus an augmented reality could be created for a firefighter allowing easier and faster fire fighting.

4. Telecommunications

Communication is an important and often underestimated topic of ship automation. Much of the currently human based standard communication could be shifted to transponders. Automatic identification systems (AIS) based on transponder technology should reduce the human communication load both for ship-to-shore and ship-to-ship communication. The Distributed Intelligent Piloting System (DIPS), Grabowski (1996,1999), may be an indication of how future civilian shipping will be based largely on communication between machines. Similar systems could also be applied for navies.

As we progress into the 21st century, crew members may have transponders on their wrist, as badges, or even fitted under their skin. These devices could carry entire medical records, security clearance, etc. In 1998, Kevin Warwick, www2.cyber.rdg.ac.uk/kevinwarwick/home.htm, had a silicon chip transponder surgically implanted in his left arm. The implant sent a signal to the computer which identified Warwick tracking his movements within the university. The system greeted him at the main entrance, opened doors, turned lights on depending where he was etc. The American company Applied Digital Solutions (ADS), www.digitalangel.net/home.asp, offered commercially in 2000 a penny-sized chip integrated in a wrist watch, called “Digital Angel”. The chip allows tracing of persons (kidnapped children or fugitive convicts) integrating wireless internet technology with GPS. The Digital Angel can also transmit selected biological functions like heart frequency and blood pressure, even a sudden fall sensor is offered. This chip could also be implanted under the skin deriving its necessary energy through natural motion of the body muscles or body heat, GEO (2000), but Digital Angels withdraw the pursuit of this offer due to public pressure in the USA. Using this commercially available technology, crew members could be traced everywhere on board automatically. Computers could automatically ascertain casualties after an attack which will be particularly useful in matters of damage control, e.g. whether to flood a room with inert gas, whether to close compartments, where medical assistance is needed, etc.

The end of miniaturization of biochips is not yet in sight. Research at MIT and Harvard Medical School started in 1989 for the retinal implant project, aiming at developing a silicon chip eye implant restoring vision in blind patients. The implants are rest on the inside of the retina and have a tiny solar power chip supplying the energy, Fig.??? By early 2002, research at the University of Southern California, Los Angeles, had progressed to enabling previously blind people to read large letters and recognizing faces.

5. Conclusion

Artificial Intelligence is about “people who research stuff that has been around for ages in the science fiction movies”, Kurzweil (1999). Artificial intelligence has also been described as a manic-depressive exercise. It appears that in several areas AI has progressed to the point of being a regular tool for engineers. The state of the art is characterized by island solutions for individual problems which already allow to reduce human work onboard ships considerably. Major advances in automated intelligent systems will result from integrating competencies now addressed individually. Merge reasoning with vision, merge diagnosis with executing, cross-reference sensors and knowledge, and you will eliminate yet more tasks now performed by humans.
References


BERTRAM, V. (2000), Knowledge-based systems for ship design and ship operation, COMPIT’2000, Potsdam, pp.63-71

BERTRAM, V. (2002), Technologies for Low-Crew/No-Crew Ships, Forum Captain Computer IV, Ed. V. Bertram, Brest


GEO (2000), Ein Engel, der zu eifrig schützt, GEO Magazine March issue


HANSEN, P.F.; PEDERSEN, P.T. (1999), Risk analysis of conventional and solo watch keeping, internal report, Danish Technical University, Dept. NAOE, pfh@mek.dtu.dk


KATAGI, T.; HASHIMOTO, T. (1990), Prospects of the diagnostic technique in the 21st century, ISME Kobe’90

KURZWEIL, R. (1999), The age of spiritual machines, Viking
See also for a short version: 130.94.24.217/specialissues/0999bionic/0999kurzweil.html


MEYROWITZ, A. (2000), NRL research in artificial intelligence, Forum on Captain Computer III, NCARAI, Washington (alanm@aic.nrl.navy.mil)


NRC (1994), Minding the helm: Marine navigation and piloting, National Research Council, National Academy Press, Washington


SCHNEIDERMAN, B. (1992), Designing the user interface: Strategies for effective human-computer interaction, 2nd ed., Addison-Wesley


SCOTT, R. (1995), Decisions, decisions, Jane’s Navy International 100/5


349
Application of Knowledge Management in Conceptual Naval Ship Design

Kees van Es, Royal Netherlands Navy, The Hague/Netherlands, c.es@mindef.nl
Martin Th. van Hees, MARIN, Wageningen/Netherlands, m.v.hees@marin.nl

Abstract

The Royal Netherlands Navy (RNIN) traditionally designs its own vessels. It has therefore a long history in Naval Ship Design; as such the typical changes, the design and engineering community encountered in the past decades also left their tracks within the RNIN. A summary is given on the evolution of design practices followed by an appreciation of the unavoidable introduction of knowledge-based conceptual design.

Already in 1992, a new approach on the use of numerical models in design was introduced at the RNIN in the form of an early version of the knowledge management system QUAESTOR. The underlying principles of this system are referred to as Knowledge-based Computational Model Assembling. KCMA allowed the designers of the RNIN to develop a new design methodology.

The contemporary methodology of conceptual design in relation to the available tools gives an insight on how knowledge management is applied. The principles of KCMA are directing the kernel of the methodology. An extensive set of engineering and analysing software tools has been added to the kernel, to produce the typical naval design and engineering products necessary to define realistic and verifiable design alternatives.

The paper will shed light on the basic concepts of KCMA. These concepts are presented in the form of intermezzos interlacing with the description of the RNIN naval ship conceptual design practise.

Design alternatives are developed to facilitate the discussion between the Naval Staff and the designers of the Directorate of Materiel. Most of the time, the demanding operational requirements for a new platform and its payload (weapons and sensors) are in contradiction with the technological and budgetary constraints of a future Naval program.

The knowledge-based design methodology as adopted by the RNIN enhances and supports the process to arrive at acceptable design solutions already in the conceptual design phase. Its possibilities leave ample room for the development of new operational strategies and technologies. In addition to this, the new design methodology allows to rapidly focus on the principal shortcomings in the knowledge available, i.e. knowledge that is instrumental to the successful development of viable solutions.

The solutions can either be reached through evolutionary or revolutionary process steps. This very much depends on the appropriate technologies used. The knowledge management impact is focused on the high level effects of the design alternatives within the decision process. This impact is mainly concerned with the balancing of the effectiveness of an alternative in the operational context against the budgetary constraints (Life Cycle Cost impact).

The paper will conclude with an appreciation of the current RNIN design methodology and a peek into the future of knowledge-based design and simulation-based design.

1. Introduction

The Royal Netherlands Navy (RNIN) designs its own naval vessels. This is considered rather unique in the international naval community. The objective of the RNIN is to design naval vessels, which fulfill the requirements of the Naval Staff within the budgetary constraints for the project. Until the end of the eighties no problems were encountered to reach good results and the RNIN belonged to the best outfitted smaller navies in the world. Although this is still true, budget cuts in more recent decades imply an increased effort to achieve similar results as before. Besides that, the navy part of the national defense budget needs to be better motivated against the needs of e.g. air force and army as well. Even so the national defense budget is weighted against the needs of other ministries controlling the budgets for education and national care. It becomes evident that a (large and costly investment) program within the navy must defend and prove its validity before it is allowed to be executed.
Within the RNIN Material directorate, in collaboration with the German navy and with the assistance of companies supporting the US navy, a design methodology was developed to support the process of finding cost-effective design concepts that fulfill the needs of the Naval Staff of the RNIN. The methodology Kenter (1997) was the basis for the development of the design tool DeSIs (Decision Support Information system) the RNIN is using now for the early stage, conceptual design of its naval vessels. In the tool DeSIs use is made of knowledge-based techniques based on the KCMA (Knowledge-based Computational Model Assembling) tool QUAESTOR, as developed by MARIN in the Netherlands, van Hees (1997).

2. Introduction of computers in the conceptual design of naval vessels

The requirements of naval vessels have changed much since the fall of the Berlin Wall in 1989. The traditional and conventionally approached “natural enemy” of the cold war seemed to have moved to other seas and within years the traditional requirements for naval vessels seemed to have lost their impact.

It is basically the uncertainty of the role naval vessels were going to play in the military scenario’s that gave the impetus to the development of “quick-reaction” design tools; an environment where the role of the computer could fruitfully be exploited.

With the evolution of the use of computers in the design also the necessary software was developed. A breakthrough came when software from heavy mainframe-computers became available on workstations and PC’s.

With the broadening of the scope of naval vessel requirements also the variety in special ship types expanded. For a specialized mission with its tasks dedicated ship types were developed (LPD) and even risk-prone Advanced Naval Vessel (ANV) concepts (like a SWATH) became options.

All in all a Walhalla for the ship designer?! Unfortunately most software developed for the widely dispersed options was director for performance analysis often for a specific scientific purpose and not so much for the development of generally applicable (broadly scoped) ship concepts, which performed well in relation to the stated operational requirements.

It became clear that the “proper” design could still not be performed by the computer, at the most by its use and the full support of the designer. The knowledge to combine the ship’s design characteristics was still an inherent capability of the designer and could not (yet) be addressed by dedicated design software. More recently dedicated ship design models were developed (e.g. Concept Exploration Models (CEM)) to support the designer, but even these were not able to express design characteristics beyond those used to develop them.

The introduction of the use of knowledge management techniques enables the designer to apply all his talent and knowledge on a certain design requirement and to develop (any) concept to enhance the performance of his design goals.

Placed in the historical context the use of computers in (naval) ship design basically developed along the following lines in the next paragraph. This will however indicate that the introduction of knowledge management techniques is unavoidable given the developments of computer science, and most of all, its perfect match with (naval) ship design as a routine.

There where traditionally the design of the latest (naval) ship was based on existing and slightly evolved concept of comparable ships, the introduction in first instance only accelerated the design process, but hardly allowed innovation and new developments.
The practice became the use of statistical techniques to capture the characteristics of designs and to produce a CEM based on this, just to provide insight in these characteristics; however all was based on the already available and consolidated knowledge of existing ship concepts.

What resulted was parameterized design knowledge, which resulted in clones of already existing designs. Possibly the insights gained from the exploration of this consolidated knowledge led to evolutionary progression in design, still basically new operational requirements could not be met and this is where the introduction of knowledge based ship design brought about the revolution in the development of new design concepts. The knowledge is not captured in statistics and parameterization anymore, but freely available to the designer in its purest physical form. Knowledge is combined by the designer to develop new concepts based on physical rules, rules based on his knowledge of ship design, mathematics, physics, and above all, technologies he may have to apply to fulfill the most demanding operational requirements to which his concept will have to perform well at acceptable cost.

3. Basic considerations for the DeSIs design methodology

Creating a new naval ship design nowadays generally is a basic capability of design experts. Depending on the uniqueness of the concept, the necessary input of new technology and the need to elaborate innovative solutions the naval ship designer usually succeeds in combining the operational requirements in a sound design. The question always remains if the most cost-effective solution was realized. On the other hand the “client” may have based his requirements on a false perception of the needs. Often the “client” already has a clearly defined concept available, which has a decisive, but not completely correct, impact on the formulation of his requirements.

The designer may have selected a parent solution without looking to deeply for other, less obvious possibilities. Also is the designer often prepared to follow the “client” in its demands for the concept.

Basically in the early nineties governments allowed navies to formulate a specification for any new naval vessel, which led to the choice by the navy from the resulting offers made by the industry against a fixed contract fitting the available budget. Too often the cost constraints were met against less performance than was required (a one-sided constraint of the solution). Requirements in those days were most of the time based on performance characteristics of the vessel itself. However, combining these performance characteristics did not always result in the required performance of the total ship concept within its operational environment.

In particular in the earlier phase of a naval program it appeared very difficult to quantify these required performance characteristics. On hindsight this approach appears to be lacking in the definition of the total context of the desired concept.

Another approach was adopted as one of the foundations for the DeSIs design methodology: the Total System Design approach (TSD) applied to early conceptual naval ship design.

4. Total System Design approach

The context of the naval ship design tool developed and used for conceptual design with the Royal Netherlands Navy is based on a Total System Design approach (TSD). The kernel of the design tool is based on the application of a knowledge based development tool, called QUAESTOR, which is a product of MARIN in Wageningen, the Netherlands. QUAESTOR is currently introduced as a general system for model assembling and knowledge management in various technology area’s to create interactive and intelligent engineering environments. Basic is the common prerequisite that the tool assists the designer or engineer to (re)use basic and specific knowledge of the metier, allowing him to spend his energy on the problem at hand and not on retracing all the relevant knowledge necessary to form the basis of the solution.
KCMA I: QUAESTOR BASICS

Although computational modeling is a common activity in science and engineering, the actual construction or assembling of computational models on the basis of domain knowledge sources has not received much R&D attention since it is still generally considered to be a programming activity. The QUAESTOR system as referred to in this paper allows an important class of computational models in design and analysis to be represented and manipulated as abstract structures that can be put together or assembled from smaller elements or model fragments, no matter the application or discipline. So, on the basis of user-provided data and a knowledge base containing computational model fragments, program interfaces, and design data (Knowledge-based), executable computational schemes (Computational Model) can be constructed (Assembling) for a wide range of design and analysis applications.

KCMA identifies knowledge coding, model fragment selection, model assembling, and execution as generic tasks. Before any calculation can be performed, a knowledge engineer captures relevant (computational) domain knowledge and stores it as model fragments in a knowledge base. The first step in the assembling of any computational model is the validation and selection model fragments, governed by a designer of an analyst if necessary or desired. The final tasks are the assembling of the selected, validated model fragments into a computational model and its subsequent execution. For this purpose, MARIN developed a computational domain and data model as well as strategies for user interaction and reasoning with numerical model fragments. The reasoning steps, heuristics and properties of model fragments were extracted from a variety of applications, such as the DeSIs system presented in this paper. The Model Assembler and the Telitab data model are the cornerstones of KCMA as implemented in the QUAESTOR shell system (see Fig. 1).

Fig. 1: Global architecture of QUAESTOR mapped on the interface

Once all the necessary knowledge for the solution is collected and arranged to solve the problem it is up to the designer or engineer to direct the knowledge to produce the solution. This collected knowledge has to be carefully managed; the process to find solutions to the overall design problem is based on experience and on the context of the problem at hand. This is where the TSD approach comes into the picture.
The TSD approach in the specific way it is applied by the RNIN has led to the development of the DeSIs conceptual design tool. This development finds its roots in the study executed by the RNIN and the German Navy within the scope of the Future Reduced Cost Combatant study, *Keizer (1997)*. This study was performed as part of NATO’s Maritime Operations 2015 study, which researched solutions for pending technology deficiencies foreseen until the year 2015, *Keizer (1997)*.

Actively involved with the development of basic assumptions for this FRCC study, Hockberger published his (further) thoughts on how to approach the conceptual design of complex naval ships in a (too narrowly distributed) publication, *Hockberger (1996)*. This publication describes why a Total Systems Engineering Design approach in naval ship design is applicable for the conceptual design practice of the RNIN. That is to say, it is still being developed for application in current and later distributions of the DeSIs conceptual design tool.

5. Application of Total System Design approach in the conceptual design of naval vessels

Following the recommendations of the FRCC-study, supported by Hockberger, an overview is given on how TSD is applied in the conceptual design.

The main principle used is the *supersystem*. This supersystem is governing all characteristics of the design concept available to appreciate its performance. This performance is expressing the operational effectiveness of the concept, given the context of this supersystem. This supersystem itself is defined as the system in which the design concept is fully integrated; this means that any aspect influencing the concept is included, or the other way around, it is the system, which includes everything that affects or is affected by the concept. This definition must be seen in the broadest possible context, so as to ensure that indeed all aspects are included. Proceeding with the development of the concept the supersystem definition can always be pruned to apply precisely enough to appreciate the effectiveness of the design concept.

To be able to appreciate the supersystem an overview must be produced of the operational requirements of the concept. This is best expressed by the definitions of the Mission Needs, i.e. the missions and tasks defined in the given scenario(s) the concept has to perform or support.

By its missions and tasks the concept plays a certain role. It is up to the environment to decide whether the performance of the concept is measurable and relevant. In other words the performance of the concept is measured against the demands of the supersystem belonging to the scenario. The scenario demands a certain performance of the components belonging to and interacting in the supersystem; this is defined in the missions and tasks. It is obvious that the concept alone cannot perform the required missions and tasks by its own and therefore all parts of the supersystem next to the concept must be taken in account to derive the (partial) contribution to the overall performance within the scenario.

It is very important to understand the application of the supersystem idea and the way to measure the performance in this context.

The performance is measured by, per definition, Measures of Effectiveness (MOEs). The total combined set of MOEs represent the overall performance of the concept given the system boundaries, the supersystem with its scenario.

A MOE is defined as the performance realized of a subset of the scenario, which is to be performed by the concept by applying its characteristics (technical contents) in relation to its environment defined by the supersystem.

An MOE can be calculated or determined by reasoning. As said, it is used to indicate the measure in which the concept fulfills a predefined set of requirements; it is important that the MOE describes the requirements of the environment to which the concept contributes with its performance. An example is
the set of staff requirements Naval Staff imposes on a concept proposal for a naval vessel. It is evident that the MOE’s are to be determined independently from the proposed solutions; this implies that MOE’s can easily, and typically, be derived in advance of the development of the concepts. Once determined, the value of MOE’s is decided by the capabilities of the proposed concept. Typically a (performance) value of the MOE is determined by the specific way a concept fulfills the predefined quality of the MOE. In other words, the MOE definition describes the required performance, where the capability of the concept for this specific MOE is used to calculate the level to which this required performance is fulfilled.

The total of all MOE’s are arranged in a hierarchy of all MOE’s, which are normalized and weighted among each other to indicate their relative importance. The resulting calculated numerical value is defined as the Figure of Merit (FOM) of the concept indicating its overall operational effectiveness.

An example is given of the meaning and application of an MOE is a hypothetical case. Suppose a navy is transporting supplies to a safeguarded beach in an enemy country with a supply-ship (Landing Platform Dock ship (LPD)). Then it is important that all supplies shipped out reach that beach over sea. Suppose that the sea lane leading to that beach is in enemy territory and a frigate is necessary to protect the supply ship. The number of fighter planes it has shot down or submarines it has successfully killed does not decide the operational effectiveness of the design concept of such a frigate. It is the amount of supplies that is landed on the beach over a given period of time that is relevant. In other words the overall MOE for the design concept of the frigate is the amount of supplies landed successfully. The subsystem, related to the overall MOE, that is relevant for the frigate contains the total of the beach, the LPD, enemy forces, other protecting forces, the sea lane and sea area and the effects the frigate may have on the performance of the LPD. It is obvious that the equipment the frigate uses to protect the LPD is essential for its success, but there are many ways to equip a frigate and there are even many design concepts for frigates or other naval vessels, who could also act successfully. The measure of success of the frigate in the defense of the LPD itself is better expressed in terms of its capabilities and each of these capabilities enable the concept to contribute to the overall MOE; however there are many combinations of capabilities to reach a certain performance, and these differ from one concept to the other. One concept is therefore more successful than the other and that is expressed by the overall MOE.

The FOM calculated does not mean so much by itself and must be balanced by the overall cost of achieving the calculated performance. A very successful concept can be very costly and therefore always the balance must be found between the performance and the cost.

In the TSD approach the FOM is balanced against the Life Cycle Cost (LCC) of the proposed concept. This gives a fairly objective assessment of the value of the concept in terms of the operational effectiveness and the affordability.

Basically all proposed concepts are given priority in order of the FOM versus LCC. Use of LCC provides insight in the effects of upfront investments against all consequences of these effects during the total lifecycle. This includes effects on the frigate concept, its cost during its lifecycle and all cost caused off its introduction in the navy (logistics, training, infrastructure).

6. Conceptual design tool DeSIs

A design concept is produced by means of concept variation. Use is made of a point design. A point design is defined as a known design of a naval vessel, where all relevant knowledge for the design exercise is available. In principle a physical description in numerical form of all characteristics is used. The point design is a relevant design with characteristics fulfilling exactly the operational requirements that are demanded. This means that a very close correlation exists between performance and characteristics. From this point design a design “space” is created by parameterization of the characteristics. This allows the naval architect to play around with the characteristics when he defines a concept for his new design.
Fig.2: Validity of the point design space

As long as this concept remains within the validity of the design space of the point design the physical representation of the concept remains valid, as well as its performance in relation to the operational requirements. The naval architect manages the design space and is supported by the Concept Variation Model (CVM), which is the principal kernel of the design tool DeSIs (figure 3). Furthermore many naval architectural tools are available in the form of analysis software to analyze the concept for its validity in stability, resistance and propulsion, sea keeping. Next to that many other tools concerning, operational performance, manning, military survivability, life cycle costing are added to DeSIs, to provide an overall conceptual design model.

Fig.3: Schema of the Decision Support Information system (DeSIs)
If a relevant design to act as point design is not available, new point design is produced, which is able to define the relevant design space necessary to perform the concept design variations in the CVM. The point design approach defining the design space has by its nature a limited validity and it is important to understand that system characteristics of a concept not fitting this design space should be provided with an other fitting design space on an other relevant point design, Fig.2.

Fig.3 gives a representation of the main components that form DeSIs. The function of the CVM is already explained; its mechanics will be explained later. To the left input is provided of operational requirements and the technology, necessary to define the concept. On top the Operational Validation Model is schematically indicated. In this part of the model the MOE’s are defined, placed in their hierarchy and used to calculate the FOM of the concept. By the MOE the following three characteristics are validated: Operations, survivability and sustainability. The first validates directly the operational requirements, the second the quality of the concept in the operational environment and the third the measure to which the concept is able to continue remaining operational.

At the bottom of Fig.3, the LCC model in found in which the LCC cost is determined. Combining the results of FOM and LCC gives the overall score of the concept, with which the concept can be compared with a benchmark design or other alternative concepts. In many cases introducing new technology produces an improvement of a design concept. The methodology of DeSIs allows technology to be assessed by including this in the CVM to determine the concept and in the end assessing the effects on the total.

![Figure 4: Presentation of the balance of the FOM against the LCC](image)

Using the DeSIs methodology also the cost effectiveness of new technology can be tested. The results of the assessment can be plotted, Fig.4. It is evident that the concepts validated in the upper-left quarter of the graph are the most successful.
KCMA II: KNOWLEDGE & DATA

KCMA assumes that any artifact or system and their properties can be described through (vectors of) attribute/value pairs. These attributes or parameters are either numerical by nature (size, mass, energy) or nominal (color, material, description). Between these numerical or nominal parameters, relationships exist that can be expressed in some explicit or implicit numerical or nominal, i.e. computational form. Parameter values are either fixed (DETERMINED) or unknown and initial (PENDING).

In the design and analysis of complex systems numerous relationships are involved: empirical, physical, spatial, legal and financial constraints and requirements, which can often be expressed in a parametric computational form. In this context any computational expression that can be applied to compute parameter values is called a ‘RELATION’ and is necessarily an equality, i.e.

$$P = f(A_1, \ldots, A_n)$$

Each RELATION should be stored in this knowledge base as an independent object containing background information, e.g. source, reliability and data, if any, its expression in parameters from the knowledge base and additional control knowledge (properties).

It is imperative to know whether a RELATION can be used as model fragment within a given context. If available, Boolean expressions called CONSTRAINTS, should be connected to RELATIONS representing their validity. A CONSTRAINT has a DETERMINED or PENDING, TRUE or FALSE Boolean value. If the CONSTRAINTS connected to a RELATION are PENDING or DETERMINED TRUE, the RELATION can be allowed into a computational model, i.e. if $\{\text{CONSTRAINTS=}\text{TRUE}\}$ THEN $\{\text{connected RELATION can be selected}\}$. The KCMA knowledge domain is a semantic network of frames containing RELATIONS, parameters and CONSTRAINTS.

A frame is a representation unit of an object and its slots contain knowledge related to the object. Each frame contains at least a type slot, which makes it possible to determine the relationship with other frames. Other slots contain properties of the frames, background information (reference), data etc.

The model assembler requires frame properties as local strategic knowledge, which are a connotation to the RELATIONS, CONSTRAINTS and parameters and affect their use in a computational model. These properties and the required inferences were extracted by reflecting on human reasoning strategies in assembling computational models. For instance, RELATIONS can be either used as equation or as function. Next to the standard arithmetic operators the expression syntax comprises a library of special functions, some borrowed from procedural languages and others to specifically deal with so-called TELITAB sets.

![Fig.5: Subset of DeSIS knowledge base](image-url)
The TELITAB set is the unifying I/O representation required by RELATIONS to form an executable computational model. The TELITAB (TExt-List-TABle) set has emerged as a logical offspring of the multi-case data structure and is the universal format in which data is stored (in the DataSet) and is exchanged within the system and with satellite applications. Next to data, TELITAB also represents the basic structure of any computational model assembled by the system, being the Solutions in the workbase, Fig.5. The TELITAB set is hierarchic in a sense that data more general to the solution or to the DataSet are located higher in the tree and are visible (inherited) in the objects below it, i.e., which is part of it. This implies that an assembled model can contain and use sub models in the workbase as special functions or as tables in numerical functions for e.g. interpolation or integration.

The TELITAB structure distinguishes three main parameter types:

**VALUE:** this type contains a (vector of) double precision value(s) and is used for numeric parameters

**STRING:** this type contains a (vector of) string(s) or document(s), the latter being a multi-line string

**OBJECT:** this type contains a (vector of) TELITAB(s). The **OBJECT** implies that TELITAB is a self-referring data structure, which is *fractal* in a sense that independent of the location or level of data, the structure is similar and has similar properties. The parametric TELITAB set unites the object-oriented modeling concept with the notion of function; as a TELITAB set can be in use a function, value and subroutine in a computational model. Figure 6 shows a TELITAB set containing model test results and analyses of test results from MARIN’s MVR system.

![Diagram of TELITAB structure](image)

**Fig.6:** TELITAB Dataset and solutions in the MVR knowledge base

The impact of KCMA in QUAESTOR on the design and analysis practice is summarized as follows:

- The development of computational models supporting engineering design and analysis is reduced to a process of acquisition and maintenance of model fragments and their properties. The model selection and assembling, traditionally a programming activity can be automated, allowing to the user to govern and control the model assembling process. Not being bothered by algorithmic issues, one can entirely focus on the quality and validity of the model fragments, being the very essence of modeling.

- KCMA requires no paradigm shift towards computational modeling. The model developer is relieved of tasks he was already performing by automating several of his own familiar inferences.
The efficiency of this automated reasoning and model-assembling process puts both developer and end user literally in control of computational knowledge and tools.

- Non-domain specialists can use systems based on this technology if the knowledge base is created and managed by an experienced knowledge engineer.
- KCMA applications can be scaled from knowledge bases containing only a few Parameters and RELATIONS up to many thousands and a large number of satellite applications, as is the case with DeSIs.
- KCMA applications can be developed and maintained in the vicinity of the daily practice by domain specialists with limited knowledge of software engineering. In small organizations with one or two users, the same person can use the knowledge base and perform its development; in larger organizations it is recommended to separate the development and end user functions.

7. Concept Variation Model: the design knowledge management of DeSIs

In the Concept Variation Model (CVM) the design knowledge of the design concept is managed. The CVM does not use its encapsulated design data in a pre-defined structure as with a CEM, but can be deliberately used and combined by the naval architect. The naval architect is managing the ship data and geometry, based on experience, and in fact is creating from the CVM data the model of the concept fulfilling the requirements and making use of the available (new) technology. The concept is generated from many aspect models, which are part of the CVM, Fig.7.

![Concept Variation Model](image)

Fig.7: Concept Variation Model (CVM)

Aspect models in the CVM are manning, geometry, propulsion, stability, signatures, vulnerability, seakeeping, construction and many others. These aspect models are as much as possible physical representations base on valid ship concepts, thus assuring that the results of the concept variation is a producible result. As design knowledge is not applied in a pre-defined way, the amount of viable combinations of knowledge must be infinite; however the application of the QUAESTOR tool manages the assembling of the knowledge to find a viable concept in a rational way based on the experience of the naval architect. The
knowledge management is guided by the pre-defined rules (based on the physics of ship design) of QUAESTOR and combined by the heuristics applied by the naval architect. Figure 7 shows schematically how the CVM is combining knowledge and the interaction of the naval architect to come to realistic ship concepts.

8. Conclusions and recommendations

The RNIN has now gained some four years experience with DeSIs. The tool has, among other projects, been used to define the concepts for the replacements of the L-frigates of the RNIN. A whole range of concepts, from corvettes (light frigates) to OPVs (Offshore Patrol Vessels) have been analyzed and costed. The discussion between the Naval Staff and the design community was supported by DeSIs. The tool appeared adequate to support the discussion and allowed Naval Staff to prepare the necessary documents to start the procedures to obtain the approval for the program. Using the tool DeSIs a lot of experience was obtained and inevitably the weaker points surfaced. During the coming years DeSIs will be improved and extended with more tools (external analysis software), interfaces, new knowledge concerning conceptual design and more efficient ways of handling data and knowledge. QUAESTOR will continue to govern the management of knowledge and maybe will open new ways to use the DeSIs.

The RNIN and the knowledge infrastructure in the Netherlands, where MARIN is supporting the RNIN, are still doing research for better ways to improve naval ship design and performance. From Hockberger’s publication and the experience gained in recent years it is obvious that Simulation Based Design and Virtual Prototyping may in the future be supporting the increasing insight, which is gained by using tools, like DeSIs. The RNIN is prepared to invest in these developments and hopes, based on these developments, to design better and more efficient naval vessels.

References

HEES, M.Th. van (1997), QUAESTOR: Expert governed parametric model assembling (PhD thesis), Wageningen, Netherlands

KEIZER, E.W.H. (1997), Future Reduced Cost Combatant study(FRCC), MO2015 final report, MARIN Wageningen, Netherlands


List of abbreviations:

ANV Advanced Naval Vessel
CEM Concept Evaluation Model
CVM Concept Variation Model
DeSIs Decision Support Information system
KCMA Knowledge-based Computational Model Assembling
LCC Life Cycle Cost model
LPD Landing Platform Dock ship
MARIN MARitime Research Institute Netherlands, Wageningen
MO2015 Maritime Operations 2015 (NATO study)
MOE Measure of Effectiveness
OPV Offshore Patrol Vessel
OVM Operational Validation Model
RNIN Royal Netherlands Navy
SWATH Small Waterplane Area Twin hull
TSD Total System Design approach
Application of Factory Simulation to the Shipyard

Yuichi Sasaki, Mitsubishi Heavy Industries LTD., Nagasaki/Japan, yuichi_sasaki@mhi.co.jp

Abstract

Recent years have increasingly seen the practical application of shipyard CIM systems, featuring integrated information management from upstream design processes through to downstream production processes. These systems make effective use of information that has been defined in the initial design phase. In the mean time, simulation-based design is an evolving technology that is widely used in mass production industries. The use of this technology facilitates design and production optimization by selecting from several potential design options. This simulation system may be very useful to the shipyard. A three-component simulation system for the shipyard (Line simulation system, Process planning system and Virtual Factory) is developed and verified.

1. Introduction

The use of simulation-based design and virtual reality technology based on 3-D digital mockup in manufacturing industries facilitates higher efficiency in terms of work strategy planning, thus resulting in major productivity gains. This technology is widely used in mass production industries to reduce costs by selecting from several potential design options and production strategies.

In the meantime, Computer Integrated Manufacturing (CIM) is starting to be applied in shipbuilding to achieve integrated information management from upstream design through to downstream production processes. Fig.1 shows an overview of MHI's shipbuilding CIM system, Yoshimura et al. (1997). Current rapid progress of information technology enables enhancement of CIM. The simulation system is expected to be useful for verification of design results and optimization of production method.

![Diagram of MHI CIM System]

Fig.1: Overview of MHI CIM

Fig.2, Sasaki et al. (2001), shows the general procedure for ship production. Generally, at the point of contract, principal dimensions and basic hull structure are essentially fixed. Thus, the optimisation of construction work strategy is crucial for the total cost reduction.
The research reported here considers a simulation system for work strategy planning, aimed at shipbuilding application. Accordingly, this paper discusses the following three components:

1) Line simulation system
2) Process planning system
3) Virtual factory for production stage evaluation

2. Line simulation system

Line simulation systems are used in the automotive and other industries for the consideration of manufacturing methods, and the application of such systems is also being considered for shipbuilding. Although it would be difficult to apply detailed work step consideration in shipbuilding, since ship is built as ordered, the system is useful for the verification of line balance among workshops during long-term planning when each workshop is viewed as a line. This kind of system can also be applied for the consideration of the introduction of new equipment in shipyards. Accordingly, consideration has been given to the realization of appropriate systems in this context.

2.1 Overview of line simulation system

Fig. 3 shows the application of a line simulation system. Long term planning must account for a period of several months or even one year ahead. It is therefore desirable to undertake simulation for ships prior to contract closing or in the design stage, but, because the design data is lacking for such ships, production planning information is unclear, and simulation cannot be carried out. In order to realize then an appropriate system, a function is needed for the estimation of unconfirmed production planning information. Also, factory load leveling is required during actual planning, while rough leveling is needed in the consideration of long term production plans. Because the numbers of processes involved in such leveling work are vast, automatic leveling functions are required. Here, the expansion of a commercially available line simulation system for the development of these functions is discussed.
2.2 Estimation function of production planning information

Although various methods have been proposed for the estimation of production planning information, the research reported here makes use of data as-is from a vessel that has previously been built. Fig. 4 shows an outline of the estimation logic. The main flow of this function presented below.

1. Actual Past Steps/Processes Used for Template
2. Modification of Schedule Template
3. Total Materials Quantity Applied to Ship Main Particulars and Estimated, Followed by Distribution to Template
4. Output of Estimated Step/Process Data

Fig. 4: Outline of the estimation logic
2.3 Leveling function

The shipbuilding leveling function was realized by incorporating factory-specific leveling conditions into an ordinary line simulator. An outline of these factory conditions is presented below.

(1) PS schedule synchronisation — In process planning, almost all of the PS blocks are assigned the same schedule. Accordingly, a function was added to make the PS schedule the same for all blocks except for those that are specifically designated.

(2) Expression of conditions before and after processes — In leveling, schedules are moved ahead or behind, although the beginning day of plate cutting and the day of block erection are not generally subject to change. An approach was therefore developed for fix these points in the context of movement.

(3) Expression of conditions for distribution — The number of blocks that can be started in a single day is determined by crane conditions/limitations at the delivery point of the workplace. Adjustments were accordingly made to enable the expression of these conditions.

(4) Expression of wrap processes — Because the number of processes in constructing a ship is so large, the simulation reported here combined multiple processes into single ones. This led to the occurrence of wrap processes, where work on a subsequent process could begin prior to the conclusion of the previous process. The line simulation system was therefore modified to express this type of wrap process.

2.4 Trial of Line simulation system

In the present project, simulation was carried out of production planning accompanying the reduction of the time at dock. Fig.5 shows the verification results of the estimation function for production planning information. Here, "Built" represents data for six ships that have actually been constructed, while "Estimated" represents data for five ships already constructed plus one estimated, such that comparison is conducted on a cumulative basis. There is good agreement overall, with sufficient precision for long term planning.

![Fig.5: An result of Estimation function of production planning information](image)

The leveling function was verified with respect to a new method in which the number of blocks is increased in the highest productivity large block section. Accordingly, based on uncoordinated process tables, work at each location was leveled by means of a leveling function, and the load for each section was verified together with the simulated schedule. Fig.6 presents an example of leveling conducted at the section level, on the left prior to leveling, on the right afterwards. In this manner, leveling was verified so as to remain within the capacity of the facility.

Evaluation of the simulation results by site managers confirmed that the simulation was sufficiently practical. The effectiveness of the simulation system was also confirmed, in that the conventional 2~3 week verification period was reduced to just one day.
3. Process planning system

Process planning for a shipyard involves to define the work strategy of the ship. The designer defines then how to built a ship from its many hull or outfitting parts. This operation is very important because work strategy defines production cost, and it is useful to compare some work strategies by using simulation, IMS (2001, 2002). Recently, the use of simulation-based design and virtual reality technology based on 3-D digital mockup in manufacturing industries facilitates higher efficiency in terms of work strategy planning, thus resulting in major productivity gains. This technology is widely used in mass production industries to reduce costs by selecting from several potential design options and production strategies. It should be useful also for shipyards, and a computer-aided process planning system is developed to evaluate the availability to the work of the process planning for shipyard.

3.1 Overview of computer aided process planning system

A computer-aided process planning system was developed for use as a 3-D digital mockup system to enable study during the design stage of construction methods, with the objective of achieving cost reductions. Fig.7 shows a basic operating screen of this system, Sasaki et al. (2002). The left-hand side of the screen is a dialogue-type network editor for the intermediate product, and the right-hand side is a 3-D representation of the intermediate product and its parts, showing shape and factory facility information. This allows to study the assembly procedure, while viewing the shape of the intermediate product.

Fig.7: Computer aided process planning system

Fig.8 shows the flow of the computer aided process planning system. Based on the hull structural data
and weld line data defined by the MATES shipbuilding CAD, the system defines the construction method in terms of the assembly sequence and production facility to be applied. The defined construction method is then evaluated from the standpoints of workability and cost, and, upon return to MATES, detailed hull drawings which express construction method is generated.

![CAD for shipbuilding](image)

**Computer aided process planning system**

**3.2 Computer aided process planning system functions**

The functions of the computer aided process planning system are a) to define and b) to evaluate the assembly procedure.

**3.2.1 Assembly procedure definition function**

In order to define the construction method, it is necessary to define the assembly sequence for all parts in the block, which in turn requires a substantial amount of time. Therefore, a knowledge-based system was developed to enable the automatic definition of the assembly sequence based on production practices and engineer’s know-how. Fig.9 shows the flow of this knowledge-based system.: First the pre-assembly base plate of hull block is selected, then assembly tree of the block is defined automatically, using production rules accumulated in the knowledge base.
Fig.9: Flow Assembly procedure definition function

This knowledge-base stores knowledge specific to MHI with respect to assembly tree generation, and incorporates current practice obtained from the comments and observations of actual production engineer. Nevertheless, as it is difficult to obtain practical assembly tree full-automatically for every kind of block, the system has been equipped with an assembly tree editor that helps for the production engineer to modify it efficiently.

3.2.2 Assembly procedure evaluation function

A calculation function, cost estimation function and assembly simulation function is developed as assembly procedure evaluation function.

(1) Calculation function
A calculation function of values used for production management has been developed for evaluation of the assembly procedure generated by the system. The values generated by the system are as follows.
1. Weight of intermediate products
2. Size of intermediate products
3. Welding length by posture for each assembly stage, Fig.10.

Fig.10: Welding length calculation
(2) Cost estimation function
Multiplication of the weld length by posture for each of intermediate product by the work difficulty rating gives the indices for evaluation of production cost. The corresponding formulae are given by (1) and (2).

\[ C_{\text{production}} = \sum (W_{\text{conversion}} \times C_{\text{unitconst}}) \]
\[ W_{\text{conversion}} = W_{\text{real}} \times K \]

Where \( C_{\text{production}} \) is the cost for production of one block, \( W_{\text{conversion}} \) is converted welding length, \( C_{\text{unitconst}} \) is the cost to weld 1m for each posture, \( W_{\text{real}} \) is the actual welding length, and \( K \) is a coefficient to express the difficulty of the welding work. This function allows to compare production cost when a construction method is changed, and can be used for study of optimal construction methods.

(3) Assembly simulation function
Assembly simulation is useful to avoid the interference in the block assembly. Fig.11 shows an example of an interference check during the design stage. In this case, the design is changed because of the interference when putting two hull blocks together. If the designer would not be aware of this interference, factory workers would have to cut the blocks, thus incurring costs. The digital mockup system is also useful, then, in helping to prevent assembly cost increases. There are many practical software applications for digital mockup, and a data exchange function from the Computer aided process planning system to digital mockup system has been therefore developed.

3.3 Trial of computer aided process planning system
The trial results for the computer aided process planning system are discussed in the following section.

3.3.1 Assembly procedure definition
An execution example is given for the assembly tree automatic generation function. In the example, structural information is taken from MATES for block 2D5P in engine room. This information is then used in the computer aided process planning system for the automatic generation of the assembly tree, and the assembly procedure information is then transferred back to MATES. Fig.11 shows the automatic generation results for the assembly tree from the computer aided process planning system. This block consists of 269 parts, and the number of intermediate product generated is 21 (only the larger ones are shown here). The intermediate products generated by system and their assembly sequences are appropriate one from practical view point. The only point to be modified was the sub-assembly blocks circled in the figure. In the actual ship, one of these two sub-assembly blocks are to belong to the lower pre-assembly blocks. Since the parts tree can be easily modified using the tree editor, these automatic generation results are considered to be fully applicable for the actual production planning. Also, it is found that the facility to be applied (in this case, flat panel line) has been correctly defined by the system.

![Fig.11: Assembly procedure generated by automatic assembly tree definition function](image-url)
Fig. 12 shows the results upon the return of the generated assembly procedure information to MATES. The screen shows assembly procedure for assembly block to sub-assembly blocks as well as the facility information. The assembly procedure is thereby stored in the MATES database, and can be used for parts development, etc. With respect to the facility display screen, the facility information defined in the computer aided process planning system is indicated by the letter K (representing flat panel line), and can be effectively utilized by MATES. In the past, assembly procedure information has generally been defined by dialogue commands, but the system described here achieves major labour savings through almost completely automatic generation.

Fig. 12: Assembly procedure generated by automatic assembly tree definition function

3.3.2 Assembly procedure evaluation

Fig. 13 shows a process plan evaluation. The work strategy for engine room block 2D31 is evaluated, with comparison of two strategies: deck based and shell-based. In this case, the deck-based strategy is better because it requires fewer upward welding operations than the shell-based strategy.

Fig. 13: Process plan evaluation trial
Use of the assembly procedure evaluation function thus enables consideration of optimal production methods.

Fig.14 shows trial result for assembly simulation performed based on a block defined by the computer aided process planning system. The assembly simulation system is realised by using commercial digital mockup system. Thus, Assembly procedure without interference can be defined by using this system prior to actual works. Also, the shipyard worker can easily image the construction method.

![Fig.14: Trial of hull block assembly simulation system](image)

### 3.3.3 Trial results

In addition to allowing 3-D visually-based testing of the assembly procedures that workers typically envision, the assembly procedure evaluation function of the new system enables more efficient study of optimal production methods. The system also helps eliminate simple mistakes and problems that have been easily overlooked in the past because of the lack of a 3-D approach, thereby allowing qualitative improvement in the definition of construction methods.

### 4. Virtual Factory

Fig.15 shows the system image of a virtual factory for production stage evaluation. Although various "kaizen" (literally, "improvement") plans for better productivity are typically suggested in the production stage, it remains difficult to actually evaluate these plans. Accordingly, a virtual factory for production stage evaluation offers the opportunity to simulate work properties and to evaluate productivity by comparison between conventional methods and "kaizen" plans.

![Fig.15: System image of a virtual factory for production stage evaluation](image)

In this system, with the basic motions of shipbuilding-related activities being converted and transferred into the database beforehand, the resulting work data is used for simulation, together with product data.
4.1 Overview of Virtual factory for production stage

Fig.16 shows the system configuration of virtual factory for production stage. The overall system described here was constructed based on ENVISION, from DELMIA. The data required for the simulation consists of the workshop arrangement, assembly tree data, parts form data, and work sequence data. The assembly tree data and parts form data were taken as output from the computer aided process planning system noted in Section 3. In order to easily define the other data, a workshop definition function and a work sequence definition function were developed, and the definitional data was converted to ENVISION.

The workshop definition function is a system for the definition of the workshop arrangement; the work sequence definition function serves to define the work sequence for each worker, based on the assembly data from the computer aided process planning system. The data defined by these systems can be simulated after conversion to an ENVISION model using ENVISION macro programs.

![Virtual factory for production stage evaluation](image)

Fig.16: System configuration of a virtual factory for production stage evaluation

5.2 Workshop definition function

In realizing computer-based assembly simulation, it is necessary to model the locations of materials storage and work performance, as well as the transit paths taken by workers, cranes, and other moving elements. Then, workshop definition function was developed to definition these data easily. Workshop arrangement is defined by using this function. Here, workshop definition model and workshop definition function is stated.

5.2.1 Workshop definition model

Fig.17 shows a example model for the assembly of a sub-assembly block, created based on the results of an investigation of the hull workshop.
Fig. 17: Workshop model for a sub-assembly block

Here, a worker transports Part A from the Parts storage area to the production area, with the flow along the transit path indicated by Fig. 18.

\[ 1 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 5 \]

Fig. 18: Flow along the transit path for part A

By defining routes and arranging nodes, the entire transit path required to produce the sub-assembly block is expressed. By applying the database information on worker positions/postures, cranes, etc. to these routes, one can "observe" workers in the simulation as they perform their tasks. By creating the entire parts storage area in advance (Node 2), materials or parts are easily added even when there are multiple types, simply by connecting a route and a node (e.g. Route D and Node 4) to Node 2. The same holds true of welding and other operations, such that by connecting a route and a node to the general work location (Node 3), the various operations required for the production of a sub-assembly block can be added. Since all the operations are taken as parts, they can be easily added or removed.

5.2.2 Workshop definition system

In order to facilitate easy definition of the above-noted workshop arrangement, a workshop definition system was created. Fig. 19 shows an example output of this system. Based on a workshop arrangement data created in Visio, this system outputs the data for the workshop definition model. The output data is converted to an ENVISION model by using ENVISION macro programs.

Fig. 19: Workshop definition system
5.3 Work sequence definition function

In order to conduct the simulation, worker movements must first be modeled. In the case of shipbuilding, these movements can be viewed as repetitions of standard actions such as welding and the transport of materials. In the research presented here, then, the standard movements in the shipbuilding context were first defined, with the simulation data created by combining these standard movements. The standard movements were defined so as to be compatible with the human model and work sequence. Here, the definition of the human model, the work sequence definition, and the work sequence definition system are discussed.

5.3.1 Definition of human model

Simulations involving human operations require the modelling of human work positions and postures. Accordingly, a human model basic position definition function was developed, allowing human activity positions to be input into a computer. Fig.20 shows a trial to define the activity positions for the human model. The human model is defined from a 3D measurement system by measuring the locations of tags placed on human articulated joints. This trial verified the realization of a human model position definition function that measures the positions of workers.

![Fig.20: Trial to define human model activity positions](image)

In order to make the work associated with constructing the human model more efficient, a database was established in the current study along the lines of each basic position or posture. By treating each work position or posture as a distinct part, the various operations performed by workers could be realized in the assembly simulation by combining these postures. As examples, work postures for a small assembly were modelled, with the basic postures required listed in Table I.

<table>
<thead>
<tr>
<th>No.</th>
<th>Basic work posture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Movement</td>
</tr>
<tr>
<td>2</td>
<td>Determining positions</td>
</tr>
<tr>
<td>3</td>
<td>Tentatively fixing positions</td>
</tr>
<tr>
<td>4</td>
<td>Adjustment (using hammer)</td>
</tr>
<tr>
<td>5</td>
<td>Confirming right angles</td>
</tr>
<tr>
<td>6</td>
<td>Welding</td>
</tr>
</tbody>
</table>

Fig.21 shows examples of the basic posture results realized on the computer. Combinations of these basic postures express the worker operations required for the production of a sub-assembly block.
5.3.2 Work sequence definition system

Fig. 22 shows the work sequence definition system. In this system, work procedures linked with the human model presented above can be defined. The work content is displayed as squares on the screen, and the work sequence is defined by aligning these squares. In addition to the work sequence, workers, tools, and working time are also defined. The data defined in this system is exported to Envision by means of the workshop definition macro.

5.4 Trial of Virtual factory for production stage

Using the method described above, a prototype hull workshop was created for a sub-assembly block, and advance study was conducted in order to boost the efficiency of workshop. This exercise was performed for the cases of one and two workers, studying working time and efficiency.

Fig. 23 shows simulation results. The left-hand side of the figure is for one worker, and the right-hand side is for two. In the case of one worker, that worker performs everything from materials transport to tentatively fixing the positions, and welding. In the case of two workers, Worker A performs the tasks from materials transport to tentative fixing the positions, while Worker B does the welding.
The simulation showed that waiting time occurred in the case of two workers, confirming the use of one worker to be more efficient. This demonstrated the possibility of using the shipbuilding virtual factory for advance study so as to improve work efficiency.

6. Conclusion

(1) A line simulation system was developed for application to long term planning. In order to enable specialization for the shipbuilding context, production planning information estimation functions and leveling functions were added to a commercially available line simulation system. Application of the system to an actual vessel resulted in the reduction of the 2~3 week time period conventionally required for consideration to just one day.

(2) Process planning system with 3-D visualization function was developed. The system enables the designer to semi-automatically define the hull block assembly sequence and to calculate the evaluation value for each process design candidate. Visualization of assembly sequence using 3-D digital mockup is useful for evaluation, and thus enables identification of optimized production work strategies.

(3) As a simulation system for the improvement of work operations in factories and workshops, a virtual factory for production stage was developed. This system includes a factory definition function and a work sequence function in order to simplify data creation, and was found to effective in the consideration of improved operations in shipyard.

References

YOSHIMURA, T., ET AL. (1997), Overview of the CIM for Shipbuilding at Mitsubishi Heavy Industries, ICCAS’97, Yokohama.

SASAKI, Y., ET AL. (2001), Research on Total Cost Evaluation System for Shipyard, 7th Int. Symp. of the Japan Welding Society

IMS (2001), IMS Promotion Center, IMS0030 Industrial Process Control, Design and Training using Virtual Reality

IMS (2002), IMS Promotion Center, IMS0030 Industrial Process Control, Design and Training using Virtual Reality

SASAKI, Y., ET AL., A study on 3-D digital mockup systems for work strategy planning, ICCAS’02, Malmo.
Market-Driven Control in Container Terminal Management

Larry Henesy, Fredrik Wernstedt, Paul Davidsson, Blekinge Institute of Technology
Ronneby, Sweden, {lhe@bth.se, fwe@bth.se, pdv@bth.se}

Abstract

The steady, global increase in number of containers and the size of vessels able to carry containers is adding pressure to seaports and terminals to increase capacity. The alternative solution to increasing capacity other than physical expansion is via increased terminal performance so that containers are loaded, discharged, stored, and dispatched efficiently whilst optimizing available resources. The automatic planning of the operations of a container terminal via market-based allocation of resources may greatly benefit the container terminal in satisfying its objectives and meeting its goals. The proposal is that a Multi-Agent System approach would offer port or terminal managers a suitable tool to plan, coordinate, and manage the container terminal domain. There exists a variety of inputs and outputs, actors, intrinsic characteristics and a large number of combinations of factors influencing the output that makes it quite difficult to conduct analysis. In the suggested approach, the Multi-Agent System will plan and co-ordinate the processes within the terminal by mapping the objects and resources that are used in the terminal. The agents will be searching, coordinating, communicating, and negotiating with other agents via a market-based mechanism, a series of auctions, in order to complete their specified goal.

1. Introduction

Seaports are important nodes in international shipping. The transfer of goods from one mode of transport to another model has been the primary function of seaports and more specifically, terminals. It is important to note that terminals are parts of a port where specialized cargoes are handled, e.g. passengers, autos, containers and oil. Ports are more than just piers. More than 90% of international cargo is moving between ports, Winklemans (2002). Of this increasingly growing trend, containerization has become the dominant method of moving unitized cargo in the world with many adverse effects such as the requirement for increasing space and causing congestion. This paper will pay particular attention to container seaports and container terminals. The needs for higher operational productivity, faster exchange of information, and speedier vessel turn-around times are just a few of many critical factors that are currently pressing port’s nodal position within logistics systems and supply chains. Logistics chains are stretching across continents where production may be in one continent and the market in another. Cargoes and shipments from all over the world have been increasing exponentially. However, seaports have not kept with the pace that economic development has been growing. In fact, many seaports are experiencing difficulties. There exist many bottlenecks in terms of information and physical status of the cargo leading to low productivity within the terminal. There are many obstacles in increasing terminal capacity through expansion, Notteboom and Winklemans (2002).

In container terminals, the management of container terminal systems (CTS) is a decentralized, poorly structured, complex, and changeable problem domain, Gambardella et al. (1998), Rebello et al. (2000). It is important that the definition of terminal operation system be explained in that it is an operating system managing the flow of cargo through the terminal, ensuring that the cargo all go the right places and that the cargo movements are handled in the most efficient manner. Unfortunately, the few “off the shelf” programs that are available (i.e. NAVIS, based in Oakland, California and COSMOS NV. of Antwerp, Belgium) are designed for specific functions and not covering the total terminal operating system. The proposal to use Market Driven control implemented as a Multi-Agent System (MAS) in container management would provide control over the various sub systems found in a CT by decentralizing the problem solving tasks to the local area agents.
The MAS approach is considered as a viable approach to CT management due to the complexity in finding a solution, because performance of terminals are determined by a variety of inputs, outputs, actors, intrinsic characteristics and external influences, Persyn (1999). Both for the CT operators and the vessel operators it is paramount to minimize “turn-around time”, i.e. the loading and discharging of containers should be done as quickly as possible. An average container liner spends 60% of its time in port and has a cost of $1000 per hour or more, Rebollo et al. (2000). To shorten the time spent by vessels, terminal operators need to spend special emphasis in resource allocation. Receipt of information before vessel berths in order to reduce the $45000 stay of a third generation containership or $65000 of a large vessel at port is important in the planning of terminal operations, Kia et al (1999). The terminal operators are obliged to provide a service that involves much more than crane moves per hour. In the CTs there exist four main subsystems and several processes that have a direct effect on each other and on the system as a whole. The MAS approach to the management of containers would allow each agent to find the container destinations through the array of subsystems that make up the CTS. By introducing auctions, agents will bid based on criteria and goals set before each auction, the agents would negotiate and bid their way through the series of subsystems found in CTS.

The use of Artificial Intelligence (AI) techniques to support port or terminal management has already taken root in some parts of the world. For instance, a family of 10 expert systems assists the port of Singapore to plan the optimal use of the port resources, which serve 800 vessels daily, reduces the stay in the port from days to hours, Turban and Aronson (1998). A number of uses exist where agents have been applied to related areas as air traffic control, Ljungberg and Lucas (1992) and recently to SouthWest Air Cargo operations, Wakefield (2001).

In the next section we describe briefly the principles of container port terminals. This is followed by an overview of related research and then a section presenting the suggested approach. Finally, we provide conclusion and pointers to future work.

2. Problem Description

Currently, there exists an estimated 15 million containers and this figure is projected to continue increasing for the next 10 years at 8.5%, Containerisation (2002). Ship lines are aware of this growth as can be seen by the huge investments in yard construction of very large container ships that can transverse the oceans at 25 knots, whilst laden with 6000 or more containers. Ports and terminal operators are also cognizant of the coming changes and perhaps threats if they do not keep up with the pace of change. Ports such as Antwerp, Rotterdam, and Hamburg are expanding their terminals or creating new terminals to accommodate the projected rise in number of containers. The planned CT investment in Europe (1999-2001) is approximately 208 million Euros, Weigmans et al. (2002). Due to increases in speed and volume, the operations of a CT requires a better regulating systems approach. One area where terminal operators are experiencing problems is reducing the unproductive and expensive container moves in a terminal. Technology such as agents may be able to assist terminals in increasing capacity and performance without spending large investments on terminal expansion and equipment. The “software” rather than the “hardware” of port development will be the determining factor in future trends in port competition vis-à-vis terminal management, Winklemans (2002). The CT is viewed not as a passive point of interface between sea and land transport, used by ships and cargo as the natural point of intermodal interchange. They have become logistic centers acting as ‘nodal points’ in a global transport system.

Congestion and increasing cargo dwell times is a common scene in many of the world’s ports. Government authorities such as customs and health may delay containers from reaching their destinations due to inspections. Shipping lines are unconcerned if there is a poor terminal productivity, as long as their vessel sails on time. Terminal operators are trying to reduce or stabilize the cost per TON/TEU (twenty-foot equivalent unit: container) handled and thus maximize profit. The aim is to efficiently use the resources available during the operating time that the vessel is occupying the berth. Complications in port systems arise in having the various computer systems work together. Currently, ports are seeking better ways in improving their productivity and offering logistical solutions to
shippers of cargo. No longer are ports handling just cargo, but more and more they are becoming “information handlers”, Henesey, (2002).

We will consider CTs that are at least handling over 50,000 TEU per annum. It has been researched that after 50,000 TEU per annum a terminal requires an Information System to help manage, Jeffery (1999). In building a model of the system, a set of operations is taken from the various sub-systems that exist within the terminal domain. In Figure 1, the four main subsystems/operations in a CTS are illustrated; (1) ship-to-shore, (2) transfer cycle, (3) storage, and (4) delivery/receipt. The two subsystems that are constantly plagued with congestion and bottlenecks are the (2) transfer cycle and the (4) delivery and receipt area (also known as the “gate”). The optimization of the vessel turn around (time spent in port) is viewed by much research as being paramount to a port’s performance and competitive advantage. We propose that a Market Driven control would provide faster discharge and loading of containers and increased productivity through faster turnaround of containers through the CTS are the primary goals.

Fig. 1: A container terminal system and the four main subsystems

2.1. Ship to Shore System

Also synonymously used as the maritime interface in that this area is where cranes handle vessels. One area where terminal operators are experiencing problems is reducing the unproductive and expensive container moves in a terminal. The number of cranes used to perform the operation varies depending on the size of the containership and the volume of containers to be handled. Usually, every gantry crane will be served with a fixed number of transport machinery, which transfer the containers in the terminal and can stack them to a certain height depending on the type of transport machinery employed. The vessel planning is typically executed 24 hours before a vessel call and produces a manifest, list of containers to be loaded or discharged is provided by the ship line.

2.2. Berth Planning System

The objective of berth planning by evaluation of congestion and cost as suggested by Nicolaou is to arrive at an optimum port capacity while incurring minimum capital cost Frankel, (1987). Each containership that arrives at a terminal will be assigned a berth, a location where a vessel can dock in the terminal. The characteristics of a container berth are the length, depth, equipment (i.e. cranes), handling capacity, and service facilities.

2.3. Transfer System

Containers are moved from berth to the storage area to be stacked or placed in an area for dispatch or containers from the stack are delivered to the gantry crane at the berth to be loaded on a vessel. The import container information such as its number, weight, seal number, and other information are recorded along with the location identification to a central database, such as a yard system in the terminal. Depending on the operations either yard tractors, front loaders, or straddle carriers are employed as transport in this operation. The type of transport employed has a direct relation to the
layout of the yard, operations of the terminal, and how the stacking is executed. The export containers are transferred from a location in a stack, thus notifying a yard system that the location is free and will be given to a gantry crane to be loaded on a vessel.

2.4. Container Storage System

There exist three main types of storage systems: short term, long term, and specialized, Frankel, (1987). The short-term storage system is for containers that may be transshipped onto another containership. Long-term storage is for containers awaiting customs release or inspection. Specialized storage is reserved for the following containers: refrigerated (called reefers), empty, liquid bulk, hazardous materials, or are out of gauge. Transtainers (either RTG-rubber tired gantry cranes or RMG-rail mounted gantry cranes) are usually employed in the sorting and management of containers in the terminal. The container storage system uses stacking algorithms in assigning a space for the container till it is loaded or dispatched.

2.5. Delivery and Receipt System

The interface to other modes of transport lies in this system. The managing of the gate is to obtain information of containers coming into the terminal so as to be properly physically handled before ship arrival and to release import containers before the arrival of trucks or rail. Controlling this access to the terminal is important that it affects other parts of the container terminal system. The data collected for example are; container number, weight, port of destination, IMO number if hazardous, reefer, shipper, ship line, and seal number are used in deciding where to place containers for storage and later for loading.

3. Related Work in Agent Oriented Approaches to Container Terminals:

The planning for port optimization and control has been traditionally been dominated by researchers in the field of Econometrics and Operations Research. In the field of Artificial Intelligence, recently there have been several papers written that incorporate the use of agent-oriented technology (AOT) such as MAS in the CT domain.

Buchheit et al (1992) have modeled a multi-agent scenario that considers parts of a terminal by using a developed platform called MARS for several shipping companies where the transportation firms carry out transportation orders dynamically and the complexity of orders may exceed capacities of a single company. Cooperation between firms is required in order to achieve goal(s) in satisfactory means. The common use of shared resources, e.g. ships and trains requires coordination between many firms. Only a partial container terminal system is viewed.

Degano and Pellegrino (2002) apply agents in operating cycles called export, import, and transshipment in an intermodal container terminal. The dispatching of containers and the stacking or storage of containers is touched upon in the research. Petri nets are used to assist in fault diagnosis and recovery. Their monitoring system uses agents that detect disturbances to a Daily Process Plan. The agents are able to perform diagnosis, and decisions in a simulation that has been validated with historical data from Voltri Terminal Europa in Genoa, Italy.

Gambardella L. et al. (1998) investigated the intermodal container terminal in a number of papers where a combination of OR techniques with simulation using agents in a hierarchical order is applied. The problems focused are the scheduling, loading, and unloading operations. The models of the intermodal terminal are based upon complex mixed integer linear program. Decision support for terminal management is divided into three modules: forecasting, planning, and simulation. The last
module, simulation, uses agents that act as an agent simulator test bed to check for validity and robustness of policy.

Rebollo, et al. (2000) have suggested the multi agent system paradigm in a few papers in order to solve the port container terminal management problem and specifically the automatic container allocation in order to minimize the time a ship is berthed. Various resources and entities such as trainstainers, yard planners, and ship planners are mapped as agents. The use of wrapper agents is suggested for legacy systems in order to provide access to the database, along with communication with external software. A prototype is still being developed.

Thurston and Hu (2002) have developed an agent simulation written in Java or the loading and unloading of containers onto vessels, also known in this paper as the ship-to-shore system. The authors focus on the quay cranes as being paramount to the total performance of a terminal. It is assumed that first all containers should be unloaded are unloaded first and those container to be loaded would be loaded after unloading has been completed. However, in reality containers are loaded and unloaded simultaneously, rarely are vessels unloaded and then loaded with containers due to time. The authors provide insight on the job assignments for the straddle carriers and how their routing may be plotted. The system has been evaluated in a simulation with randomly generated data.

Lee et al., (2002) analyze the port operations via agent-based simulation for the planning and management of the CT. As with Thurston and Hu (2002), they have focused on the berth allocation and the crane policies. The researchers simulated the PECT terminal in Busan, Korea by testing various policies with physical and logical agents. The agent based simulator results indicated that the stronger the partnership relationships between shipper agents and CT operator agents, the faster the handling of containers. The study was primarily focused on the ship-to-shore system and the transfer system.

The Market Driven control to container management is viewed as a possible holistic solution to the container terminal system through decentralized problem solving within the sub systems of the CT leading to a global solution. In the next chapter the subsystems of the CT are defined and the conceptual model that is currently being developed is discussed.

4. Market-Based Control

In the next chapter, we will describe a market-based approach to CT management. The motivation of using market-based control is formulated from auction theory in economics where system wide costs are minimized, bidding agents will bid according to their true values, and auctions offer a specifically short-term contract that ignores long-term implications. Much interest has been garnered in the use of market mechanisms in AI. Perhaps the interest in the Internet has swelled such interest in the form of electronic markets and even auctions, Sandholm, (1999). A large informal body of knowledge on auctions has been in existence for centuries, and a more formal, game theoretic analysis of auctions began in the 1960’s with the pioneering work of Vickery, Vickery (1961). Market-based control is viewed as a paradigm for controlling complex systems that are difficult to control or maintain. In this paper, we consider the port terminal domain to be a complex system and difficult to be structured quantitatively. The fundamental properties of such complex systems consist of the following notions, Gosh and Lee (2000):

1. Entity: characterized in the CT domain as resources, such as gantry cranes, straddle carriers, lorries, and ships having consistent behavior that does not deviate, i.e. straddle carrier will not change roles with gantry crane.
2. Asynchronous behavior of the entities: various entities on the CT, such as gantry cranes, straddle carriers, lorries, and, ships are encapsulated with unique behavior described by functionality and timing.
3. Asynchronous interactions between the entities: not all the resources in the CT have the knowledge to execute a task, thus the sharing of information is necessary to carry out jobs, i.e. the straddle carrier can not load container in the vessel only the crane can and the crane can not travel to the yard similar to a straddle carrier.

4. Concurrent execution of the entities: simultaneous occurrences of lorries, trains, and vessels entering and leaving a CT with varying number of containers.

5. Connectivity between the entities: the sharing of data, information amongst the resources in the CT constitutes connectivity.

Market-Based control has been proved to be a suitable tool for complex resource and task allocation applications, Bredin at al. (1998). It is interesting that markets are not initially perceived as a means to control a system. In the market-based system, the agents are provided with individual goals and through their interactions with other agents in an auction, a control of the CT system is achieved. Since the CT “owns” the agents, there is no security threat from agents acting selfish or behaving greedy. For the market to function we assume that agents will not bid more than they can and that agents will honor agreements. The view is that agents should act benevolently in that agents will not cheat or lie, but will buy or sell when they can. The agents in the CT system view resources, i.e. time and containers as assets that can be bought and sold. The auctions protocols currently being considered for the prototype for the various resources within the CT are proposed to be a Market-Driven Contract Net, Clearwater (1996). Where a task would be generated as request for bid (RFB) and broadcasted to all resource agents. The resource agents would make bids according to their cost (based on position, time and operating cost) to carry out or execute the RFB (task).

5. Multi-Agent System for Container Port Terminal Planning

In this section we present our suggested approach to a market based system for allocation and dispatch of containers within a CT. The system is primarily used for creation of work orders, container yard allocation and berth planning. The system uses the agent and multi-agent system metaphors in that the mapping of functionality in the container port terminal is made in terms of agents. The system will make use of auctions where agents are free to bid and raise their bid until no other agent is willing to bid any longer. The auction setting depends on the value that each agent places on an activity. A setting that could be utilized is the correlated value auction, Weiss (1999), each agent bidding is dependent on its preference and the value that other agents may have for handling the task. In Figure 2, we show the main flow of resources traversing the system as well as the four different types of global agents inhabiting the system:

The ship agent is instantiated upon the planning of an arriving vessel. The agent will, before the final decision of the berth location, interact with the berth agents to decide where the most cost beneficial berthing can be achieved. The agent gains revenue when discharging/selling containers to the terminal and has expenses for the loading/buying of containers.

The berth agent is responsible for the allocation of resources at a dynamically changing part of the quay. It will upon request calculate the current price for the berthing of a ship with an indicated loading manifest (list of containers). The berth agent calculates the price by issuing requests for crane resources, container transportation and container storage.

The yard agent is responsible for a dynamically changing storage space in the terminal. The agent will on requests for container storage, respond with a bid by calculating the value of the specific container, e.g., is there already containers in the dedicated storage with similar destination data, is there any space available and is it allowed to store the container at that space? Other impacts on the agent bids are the expenses related to transportation of the container and the subsequent need for transtainers to lift the container into place. The agent will during loading sequences of ships demand revenue for the dispatch of containers from the storage area. The agent will also request revenue for the dispatch of containers to the gate.

The gate agent is a logical wrapper to the physical gate. The gate agent allocates containers to the terminal storage by awarding the containers to yard agents and requests stored containers when dispatching containers to land transportation.
Fig. 2: The direction of the revenue flow during loading and discharge of a ship.

In addition to the agents mentioned above there are three other types of agents that are used by the global agents as utility agents:

The crane agent is a mapping of a crane (typically a gantry crane) for the loading and discharging of a ship. The agent is concerned with the optimal usage of the crane in that it will try to minimize the number of location shifts in relation to the maximum utilization of time. This agent is one-sided in the auctions in that it will always sell its service and its costs will be based on its operating running cost.

The transtainer agent is a mapping of a crane used for the movement of containers within a yard. The agent is mainly concerned with optimization of the allocation of containers within a designated space. Typically it will make use of queuing theory, stacking algorithms and other existing techniques for positioning the containers so that a minimum of subsequent handling is necessary.

The transport agent is a mapping of a transportation vehicle. The main goal is to utilize the vehicle as optimal as possible both for allocation as well as for dispatch of containers. The utility function for the transportation agent is the degree of occupancy in relation to the distance to travel. The transport agent is one-sided in that they are always selling their service and not buying in the auctions.

The system architecture mainly supports the following activities:

Allocation of incoming containers to the terminal yard (see Fig. 3). A gate agent will on receipt of a container initialize an auction and request the yard agents to bid on the specific container. The yard agent has to take into consideration the cost and availability of transtainers and transportation as well as the likelihood that it later can sell this particular container at a higher price. The gate agent awards the container to the yard agent presenting the best bid.

Fig. 3: The yard agent has a cost for receiving a container

Dispatch of containers from the terminal yard to ships (see Fig. 4). A ship agent will make a request for a price for the loading of a set of containers at a specific berth. The corresponding berth agent will calculate the price by issuing a request for the containers to the yard agents. The yard agents will indicate a price as well as availability (depending on the transtainers) if the container is stored within its area. The berth agent then requests a price for transportation and cranes. Depending on the availability of a load order list,
the exact sequence of containers is used when calculating the price, otherwise the availability, distance and occupancy determines the price. The final decision on which berth the ship will use depends on the lowest price presented by a berth.

**Fig. 4:** The berth agent has cost for making the containers available for loading onto a ship.

**Allocation of the yard with containers discharged from a ship** (see Fig. 5). A ship agent will make a request for discharging a container to the berth agent. The berth agent then initializes an auction and requests the yard agents to bid on the specific container. The berth agent will also have to request operations from the crane agents to lift the container off from the ship. The crane agents will sell their service to the berth based upon their operating costs, thus crane will acquire income.

**Fig. 5:** The yard agent has a cost for receiving a container

**Dispatch of containers from the terminal yard to land transportation** (see Fig. 6). The gate agent will make a request to the yard agents for a container upon demand from a land transportation source. The gate agent will also have to initialize an auction to receive bids for transportation from the yard to the gate.

**Fig. 6:** The gate agent has expenses for making a container available for dispatch from the terminal.

**Reallocation of containers after final decision of berth.** The ship agent will continuously interact with berth agents to determine the current price for an actual berthing. After the final decision of berth, the yard agents can start buying and selling containers among them if the cost or price of shifting is beneficial to the yard agents. Optimally, the containers are already stacked close to the awarded berth but the shifting of one or two containers may improve turn-around time for the ships.
6. Conclusion and Future Work

Research in applying MAS approaches in container terminal planning and management issues has been gaining popularity due to the complexities in solving the problems. Researchers have proposed varying methods in applying MAS in CT. We have suggested a multi-agent architecture based on a market-based approach. This is our initial approach towards a holistic solution to a very complex domain. The MAS approach to the automatic planning will generate several work schemes. Furthermore, the planning will assist terminal management when executing decisions from the work schemes.

The system is to provide dynamic yard allocation, dynamic berth allocation, and will reduce idle time of transport vehicles. Furthermore, the main goal is to optimize the capacity of the terminal, which is measured by four main performance indicators: measures of production (e.g. traffic or throughput); measures of productivity (e.g. crane moves /hour); measures of utilization (berth occupancy) and measures of level of service (ship turnaround time). Some questions that concern CT performance are length of time to move equipment and supplies through the CT, what and where are the potential bottlenecks and limited resources to movement through the CT, why are operations not completed by the required time, what are the implications if certain seaport resources are constrained or available? What are the port throughput capability given explicit assumptions on assets, resources and scenarios?

We are currently developing a CT simulator that will be used to evaluate the market-based approach. The simulator will run scenarios where the interactions between the agents within the system will follow the information patterns that are generated and executed by physical moves, i.e., the system will map the flow of an actual container terminal.

The suggested approach needs to be concretized in several aspects, e.g., which auction protocols should be used, and how is the update of information to be achieved. Furthermore, the system needs to be validated and evaluated.

References


SURF- An In-House Development for a Ship Hull Design Software

Virgil Mutu, Vuyk Ship Design Ltd, Galati/Romania, sdbg@ vuyk-galati.ro
Ovidiu Ionas, University of Galati/Romania, o.ionas@vuyk-galati.ro
Ionel Gavrilescu, University of Galati/Romania, igavril@xnet.ro

Abstract

In shipbuilding sector, commercial computer applications for design and manufacturing of ship hull have been under a continuous development process during last decades. Some of the most important achievements in this area were, and still are, the integrated systems (CAD/CAM) and the applications for numerical hull shape definition. Besides these commercial applications, the design offices and shipyards have developed their own software, easier to use and more flexible. SURF is one of these software. Made by Vuyk Ship Design, SURF is an in-house application for ship hull shape design. The software package SURF contains three main modules, each for a special task (numerical definition and fairing of the hull shape; obtaining the cross section through shape; shell plate development and bending information) and some small applications for links and interfaces with other programs. This paper will present the SURF application and will describe the functions of each module and the methods used during the hull shape definition and shell plate development.

1. Introduction

These days the shipbuilding industry could not be imaginable without software and integrated systems for design and manufacturing. Due to the high expectations and requirements of this sector, this kind of product must have some important qualities: high accuracy of output information, friendly interface, easy to use and learn, capacity to solve a wide variety of tasks and not least, capacity to assure links and interfaces with other systems.

In particular case of hull design, each design stage is solved by specialized software. On the ship design market there are several software products that solve the computer aided hull design issues in a more or less integrated manner. Integrated systems, already imposed on the market, have some disadvantages; among them being the difficulty to adjust the system to particular situations (repairs and conversions, special-purpose vessels, etc.). This kind of disadvantage disappears in case of in-house software, due to the capacity of adjusting to different customer’s demands, with lower resources consuming. It seems that a mix of integrated systems: specialized task software and in-house software, with the possibility of information transferring between them, is the best way to use the naval applications. In this context our company – VSDG – uses specialized task software for initial design stage and integrated systems for engineering and detail design, some of this software being in-house applications. Furthermore VSDG has developed link and interfaces between these applications and has managed to obtain a new and efficient “integrated software package”.

Vuyk Ship Design Galati (VSDG) is a design and engineering company, founded in 1991, offering a complete set of services, starting with the earlier design stages up to workshop information. A main activity of VSDG is the development of new software for ship design and research, among them the most important being SURF - a CAD-CAM applications for numerical hull shape definition and shell plates calculation.

SURF is a powerful software developed in VSDG, having three main modules:
- SHAPE – module for numerical definition and fairing of the hull shape, using advanced numerical methods (Coons, B-spline and NURBS combination), mathematical and graphic methods for checking of surface quality, powerful 3D graphic representation.
- LINE – module for cross sections in hull shape using any 2D or 3D section plane, optimization of cross line description and line export to other software.
- PLATE – module for numerical definition of seam lines, shell plates expansion, bending information, remarking of stiffness on shell, seam lines offset table.

This paper presents the SURF software package and its contributions to parametric hull shape definition and double curved surface development methods. Besides introduction and conclusions the paper includes 3 main chapters. In the second chapter are presented the numerical methods used for smoothed hull shape definition and shell plate calculation. Chapter 3 presents the SURF application with its components and the means used for integration with other software and systems. Some numerical aspects of implementation and examples on real ships are presented in Chapter 4.

Over 50 ships were built up based on SURF-made information, in different shipyards. SURF proved to be a powerfully and efficiency application, due to accuracy of the output information and high productivity. The possibility of transferring its data to other software used in shipbuilding activities (NUPAS, NESTIX, AutoCAD) should also be mentioned.

2. Theory

2.1. Philosophy

The definition of the ship hull surface is the base of the whole system. The system must admit the definition of a new shape, but also the “identical” reproduction of an actual shape. To achieve this, three major concepts are defined: nodes - points on the surface, curves - who pass through the nodes and surface elements (patches) - supported by curves. The nodes are defined by the user by their coordinates (x, y, z). The nodes must be defined in a way that permit maximum of accuracy with minimum of information. The curves are defined by an ordered succession of points. There are two types of curves: main and ordinary. The main curves are the important curves of a shape: boundary, knuckles and tangent lines (CL, deck, flat side, flat bottom, etc.). The ordinary curves help the user to define the shape and to generate the “eyes” for surface patches. The patch is the elementary surface element supported by curves and defined by the corner nodes. The curve continuity must assure the surface continuity without any other user intervention. The idea is: the points define the curves, the curves define the surface elements. In this way the user will check up only the nodes co-ordinates and, sometime, the end conditions of the curves. The checking of the patches defined surface is made implicitly by means of the curve control. Thus the user will focus only at points and curves, the surface being generated automatically.

The surface definition is not only a purpose, but also, above all, a manner to achieve the engineering objectives, such as: the surface lines computation, the shell development and the bending information computation, the last being the re-folding of the surface.

To achieve these objectives four primary procedures are necessary (Fig. 1):
- shape definition;
- sections through surface;
- curves generating;
- interposing planes generating.

2.2. Conditions

The mathematical models that will be used must accomplish with the following conditions:
- to admit 3D shaping of the hull surface, including the “identical” reproduction of an actual shape;
- to be numerical stable;
- to have a reasonable amount of input data;
- the number of checking parameters to be as few as possible, and these parameters to have an engineering meaning;
- the necessary CPU time to be reasonable.
2.3. Shape definitions models

2.3.1. Curves definition

We use for this the 3 dimension cubic spline curves in parameter equation \( h = h(u) \), where \( h = x, y \) or \( z \). The spline curve \( C \) is composed from an assembly of \( n \) curve segments \( C_i \) (\( i = 1..n \)), that pass through \( n+1 \) definition points \( P_i \) (\( i=0..n \)). In internal points are imposed continuity conditions – type \( C^0, C^1 \) and \( C^2 \). The \( C^0 \) and \( C^1 \) type conditions are implicit, but the imposing of \( C^2 \) type condition leads to a linear equation system \((n-1) \times (n+1)\) double undetermined.

\[
h_i(u) = \{u\}^T [M_i] [G_i] \quad i = 1..n
\]

where:
- \( \{u\}^T \) – parameters vector;
- \([M_i]\) – Hermite matrix;
- \( \{G_i\} \) – conditions vector.

To solve the curve definition two more conditions are necessary. These additional conditions, imposed in nodes \( 0 \) and \( n \), could be known slope or curvature type. The solution of the equation system consist in slopes \( m_i \) (\( i=0..n \)) in definition points. With the points and slopes the curves is completely defined.

2.3.2. Surface definition

The user can choose two different models for surface definition. The S surface is composed from an assembly of elementary surface elements (patches) \( S_i \) – parameter defined. The patches are in Coons or B-spline formulation, in accordance with the model chosen. The patch is defined by the generic function \( h(u,v) \), that can be \( x(u,v) \), \( y(u,v) \) or \( z(u,v) \). For quadrilateral surface elements (four corners) and with cubic variation of the parameters the generic function is:

\[
h(u,v) = \{u\}^T [C]\{v\}
\]

where \( \{u\}, \{v\} \) are the parameters vectors and \([C]\) is the curvature coefficients matrix.

The coefficient matrix and the continuity conditions impinge upon the surface definition.
The Hermite surfaces in Coons formulation have the following curvature coefficient matrix:

$$[C] = [M_h]^T [Q_h] [M_h]$$

where $[M_h]$ is the Hermite matrix and $[Q_h]$ is the condition matrix.

In Coons formulation of the Hermite surface model, the condition matrix contains the co-ordinates of the corner points and the derivatives in these points, but also information concerning the marginal curves of the patch. If you can assure the continuity of the marginal curves segments and the continuity of the corners derivatives, then the continuity of the surface composed from Coon patches is implicitly. The continuity of the marginal curves assures the $C^0$ and $C^1$ continuity of the surface. The problem occurs at the estimation of the corner derivatives, especially the mixed derivatives.

B-spline surfaces. The defining conditions for the B-spline bicubic patch are the co-ordinates of 16 control points. The surface is generated for the internal “eye” of this 4x4-point subnetwork.

The coefficient matrix is $[C] = [M_b]^T [Q_b] [M_b]$. where $[M_b]$ is the B-spline matrix and $[Q_b]$ is the condition matrix composed from the co-ordinates of the control points. The uniqueness of the control points network leads implicitly to the smooth coupling of the B-spline patches.

2.3.3. Improvements

Some improvements of this mathematical methods were necessary in order to comply with the above mentioned principles and conditions:

a) Tangent type end conditions of the curves were defined using a straight segment $P(0)P(1)$ attached to curves end. The end condition will be $\alpha = \frac{\partial}{\partial t} P(t) \bigg|_{t=j}$, where $\alpha$ is the slope of straight line $P(0)P(1)$. The spline approximation begins from second definition point, first segment being straight. This allows the easy tangent control using co-ordinates of a point instead using the 3D tangent vector. Furthermore, using this procedure is very easy to define o tangent curves to a flat surface, for instance frame line tangent to flat bottom. This kind of curve can be defined and checked using the number and co-ordinates of the definition points and the end condition.

b) The “classical” Hermite conditions $(1)_{i} = (0)_{i+1}$ were substituted with more “relaxed” conditions $(1)_{i} = m \cdot (0)_{i+1}$, during the coupling of the curve segments process. The curvature factor $m > 0$ is generated automatically using the local curvature. This allow the usage of a small number of definition points for curves, and leads to curves that look like the usual ship lines.

c) The simple derivatives from conditions matrix of Coons patch were automatically calculated using information obtained from boundary curves of the corner point with the following relations

$$\left( u \right)_{i} = \frac{\partial}{\partial u} P(u,v) \bigg|_{v=j} , \left( v \right)_{i} = \frac{\partial}{\partial v} P(u,v) \bigg|_{u=i}.$$ These derivatives are not input data and have any intuitive purport. The double derivatives are calculated using finite difference method or using macro-patches, described at point f). This assure the automatically $C^2$ continuity of the surface patches.

d) The imposition of some “relaxed” continuity conditions for patches using a curvature factor similar with thus from point b).

e) The B-spline patch was defined using only the corners points (4 nodes), the other 12 control points being generated automatically based on some extrapolates of the outside patch surface. This procedure allows also to assure the $C^1$ continuity for the coupling of independent B-spline patch, except when on this direction are knuckle line where the continuity is $C^0$ type.

f) Using the macro-patches. When you decide to use the Coons patches and expect to obtain an excellent surface fairing, you have the option macro-patches. The macro-patch is a patch disposed over a rectangular network of ordinary patches. The program generates automatically a B-spline...
patch over the network of internal points and curves, and from this macro-patch will get the information concerning the derivatives necessary for the internal Coons patches. This allows to reduce the number of input data and to obtain an impeccable fair surface.

2.4. Surface usage

The surface defined with these improved methods will be used for section lines, unfolding of the shell plate and for the bending information. The usage of the non-linear and unfinished mathematical shape that is defined on different surface elements (patches) will lead to a succession of difficulties:
- non-linear equation system with multiple solutions;
- a lot of CPU time consuming;
- complicated management of information;
- difficulty to convert the parametrical co-ordinates into the ship cartesian co-ordinates.

To avoid this kind of troubles a new system was implemented: through every surface patch an orthogonal net of \((u,v)\) points is generated. These points are obtained in cartesian co-ordinates of the global reference system. The points are joined using line. The result: a “mesh” with quadrilateral eyes and straight sides.

The points are generated using an iterative algorithm that assures a sufficient accuracy for the reproduction of the surface. This accuracy is adjust by the user (about 0.1 mm) and represent the distance from the mesh to the analytic surface. The mesh can be temporarily or permanently stored and despite the big dimensions (1...50 Mb) is facile to use.

2.4.1. Section lines

*Plane sections.* Every section through surface represent a section of the mesh with a plane, this means the 3D intersection between a plane and some straight segments (mesh lines), a fast and easy operation. The resulting points will be ordered using a simple algorithm. The section curve will result like a polyline that has the deviation from the surface smaller then an imposed error.

*Definition curves.* During the processing of the information it is necessary to extract from the surface some spatial curves that were used to define the surface (for instance: boundary curve, knuckle, tangent lines, etc.). These 3D curves can be projected on a main plane or can be used like a spatial curve.

These curves will result like an ordered table of points defined by their co-ordinates. These co-ordinates can be local co-ordinates (of the section plane) or global co-ordinates (of the ship). The export to other CAD-CAM applications can be done immediately without any other post-processing.

2.4.2. Surface unfolding (shell plate development)

The main steps of this operation are:
- defining the development area (shell plate);
- the properly development;
- checking of the development.

The development area is defined using boundary curves. These curves may be definition curves or section curves. The section curves result from the intersection between the surface and the section plane, which is defined by two points and a parallelism condition. The boundary curves must define on the surface a closed curve quadrilateral.

The unfolding of the limited area is made using the slices methods and following the next steps:
- the median curve for the surface is determined;
- the median curve is divided into segments, the density of the segments is in accordance with the curvature of the surface;
- the quadrilateral surface is automatically divided into parallel slices using section planes that pass through the median curve segments. The direction for these planes is imposed by the user;
- starting from a reference axis (the straight line on the unfolded surface), each slice is divided into triangles and is unfolded independently by the condition to keep constant the lengths of the triangle sides (the length of the side of the curbed triangle on the surface is equal to that of the plane triangle on the unfolded surface);
- the unfolded slices (independently) are placed on the reference axis. Because the common curve of two consecutive slices is no identical after the unfolding, the final placement on the unfolded surface can be made in many ways. In extreme situations the positioning is made without overlapping of the slices (supposed that the bending will be made by compression of the plate) or without gap between slices (supposed that the bending will be made by stretching of the plate).
- after the unfolding of the outline defined by the boundary curves, on the unfolded surface are identified and placed the welded joints, the marks, the rolling lines and the tangent lines.
- in the outside of the welded joints the margins (over-length material) for cutting are placed.

The development of the double curved surfaces is an indeterminate problem; this leading to different results in accordance with the development method used. To measure and validate the quality of the development the following methods are used:
- the comparison of the 3D surface lines lengths with the unfolded surface lines lengths;
- the estimation of the plastic deformation for unfolded areas.

2.4.3. Bending information

The purpose of the bending information is to offer the necessary information for the “flexion” of the unfolded surface to recompose the curved surface. The bending information is in accordance with the technology that is chosen for the process and can be with templates and model (mock-up) or with PIN-JIG.

A general issue for every computational methods of the bending information is to find the reference plane for bending. This plane is made automatically by regression of the surface nodes and has the following attributes:
- can be a random plane or parallel with a main plane;
- must interpose the surface, that means the surface must be “relative” close to every point of the plane;
- have a minimum imposed distance to the surface;
- is situated on an imposed side of the surface.

The template is the “negative” of a section through surface and the mock-up is an ensemble of templates. The template is a closed profile composed from a curve and a straight line. The curve is determined from intersection between the surface and the template plane and the straight line is determined from intersection between the template plane and the reference plane.

The template information includes also:
- the offset table of the template;
- the position of the fire line (reference line);
- the position of the rolling lines;
- the mutual intersection between the templates;
- the angles of the template with the surface and the reference plane;
- the template orientation.

The PIN-JIG information includes the dimensions for the arrangement of the pins in connection with the surface and the distances from the reference plane to the surface (the heights of the pins). These heights take into consideration the thickness of the shell plate and the inclining effect. The PIN-JIG information can be also used for numerical bending.
3. SURF application and its components

3.1. The SURF system

The SURF system is a Windows application, developed by Vuyk Ship Design Galati under Delphi language. The SURF system (Fig.2) is composed from three main modules (SHAPE, LINE and PLATE), each of this modules having internal applications that allow the integration and transfer of data with other systems.

Fig.2: The block diagram of the SURF system

The functions and methods of each module and also the import-export functions of the system are presented bellow.

3.2. SHAPE – Surface generation

The SHAPE module is intended for numerical definition and fairing of the surfaces. These surfaces are generally areas of the main hull shape but the software is suitable also for definition of special surfaces such as skegs, bulwarks, rudder, anchor pockets, propeller blades, etc.
The surfaces are defined with points, curves and patches (Fig.4) using the methods presented in Chapter 2. The control of the curves is made through the checking of the curvature (Fig.6) and the control of the surface is made using the section lines (Fig.5).

The input data for this module differs in accordance with the design stage. For the initial shape design the user needs only some general information concerning the hull shape: main dimensions, frames type (U, V, straight), bilge type (rounded or knuckle), dimensions and type of the keel and sheerstrake, transom type, etc. For detail design the user must reproduce the initial shape. In this case he will need the lines plan and the offset table.

The output data of this module is a file (named *.zon) that contain the surface of a smooth area (zone) of the ship hull. Therewith, from this module the export of the hull shape to CARENA application is made. CARENA, other in-house naval software of VSDG, is used for computation of hydrostatic characteristics, tank capacity, sounding tables, ship's stability and longitudinal strength.

3.3. LINE – Section lines

In this module information about section curves and definition curves are obtained. The user just define the intersection plane or choose the definition curves and these are exported in 2D or 3D coordinates. Also the user define the planes for the intersection of the curves in order to obtain the offset table.

The input data for this module are the files *.zon from SHAPE. The LINE module makes the coupling of these zones and generates a mesh on the hull surface (Fig.8). This mesh is intersected with the curve plane. The output data are the offset table and the 2D or 3D lines exported in local or global coordinates (Fig.8). Imported in AutoCAD these curves are used for the 2D and 3D lines plan or for construction of the internal parts. The LINE module can export the hull shape to NUPASCADMATIC system (Fig.7).

3.4. PLATE – Shell plates

Is the module for numerical definition of seam lines, shell plates development, bending information (templates and PIN-JIG) and offset table of seam lines. The shell plate are defined through:
- boundary curves;
- seam and butts (Fig.10 and Fig.11);
- reference axis and the median curve;
- the marking lines;
- the margin (over-length of the material).
The shell plate is unfolded using the algorithms described in Chapter 2. On the final unfolded plate the margins, the marking lines and the rolling lines are placed (Fig.9). The plate is exported to AutoCAD where is processed and saved under dxf format ready for nesting. Some additional informations are also calculated: the plate surface, weight and centre of gravity, the necessary material format, etc.
The templates (Fig.12) and PIN-JIG compose the bending information. These informations are calculated using the methods described in Chapter 2.

For this computation the following input data are necessary: the *.zon files from SHAPE, the preliminary seam lines plan and the block division. The output data presented above are obtained directly or using AutoCAD.

3.5. Import-Export functions

To assure the integration of the SURF application with other software systems some import-export functions were necessary (Fig.3). These functions are developed like internal applications of the SURF modules and are presented below.
**BMP Image export**
- application included in all modules for images export to other WINDOWS programs;
- output - a BMP file with selected image.

**Shape export to CARENA**
- application included in SHAPE for conversion and export of numerical shape definition from SURF to CARENA;
- output - files with nodes and patches in CARENA specific format.

**Shape export to NUPAS**
- application included in LINE for conversion and export of numerical shape definition from SURF to NUPAS;
- the selection and density of the lines exported are users choice;
- output - non-standard dxf files with NUPAS specific format lines.

**2D&3D line export to AutoCAD**
- application included in LINE and PLATE for conversion and export of 2D and 3D lines and seam lines obtained from sections through hull shape, shape definition or shell plate definition;
- The user can select the definition points density according to the error control balance between the polyline and the surface curve;
- output - a drawing with 2D or 3D polyline in AutoCAD specific format, reported to active UCS.

**3D line export to SURF**
- external application developed under Visual Basic extract definition points from a 3D AutoCAD polyline;
- The user can specify the density of points extracted from the polyline;
- output – point’s coordinates in SHAPE format.

**Plate export to AutoCAD**
- application included in PLATE for conversion and exports of the expanded shell plates information, including edges, marking lines, margins, main bending line;
- output - a drawing with a group of 2D polylines in AutoCAD specific format, reported to active UCS.
Template export to AutoCAD
- application included in PLATE for conversion and exports of the template information, including edges, reference line and mutual intersection between templates;
- output - a drawing with a closed 2D polyline in AutoCAD specific format.

PIN-JIG information export to AutoCAD
- application included in PLATE for conversion and exports of the PIN-JIG information, including seam lines, depth to pins, position of plate's corners, angles of structural elements;
- output - a drawing with pin grid, seam lines with dimensions and numerical notations.

Remarking information export to AutoCAD
- application included in PLATE for conversion and exports of the specific information for plates remarking, including seam lines, new lines for structural elements, dimensions and lengths;
- output - drawing of plate with lines for structural elements, dimensions and numerical notations.

4. Applications and examples

In order to clarify how the SURF system works and look, in this chapter some applications and examples are showed.

Fig.4: 44m Voith Schneider Escort Tug
- Points and lines for surface definition-

Fig.5: 24m Tractor Tug
- Section lines for surface control-

Fig.6: 120m River Barge
- Curvature evolution along frames -

Fig.7: 3880 m³ Hopper Dredge
- 3D Section lines ready for export -
Fig. 8: 85m Inland Tanker - 3D and 2D Section lines

Fig. 9: 28m ASD Tug - Shell plate development process –
5. Conclusions

The software is based on modern mathematical methods and algorithms. These basic models have been improved with many original ideas to increase the speed of computation, reliability and to obtain a user-friendly tool.

The large generality of algorithms and structure of input data make this software applicable not only for ships but also for any kind of 3D surfaces.

A special consideration was given to the user interface. The simplicity and intuitively of input data, the small number of “tuning” parameters, the large number of checking options, make from this system a “stressless” tool.

The SURF system can be very easy connected with many “standard” CAD-CAM software due to the large number of data import-export functions.

The development of this application is a real success. Using this software VSDG was able to enter, without major investment, in the naval market of the shape definition, detail design and workshop
information. Using the important advantage of in-house software the SURF system is permanently improved and adapted to the specific request of the shipyards, without supplementary costs and in very short time.

Based on this software VSDG design company has made the initial design (lines and shape), hull detailed design and production information for over 50 ships in the last years, for customers from West and East Europe. During the time, the software has proved his accuracy, flexibility and reliability.

References


IONAS, O. (1999), Contribution in naval architecture based on numerical definition of hull shape, Ph.D. Thesis, Galati University


NAPA System Supporting Ship Design

Tommi Toivonen, Napa Oy, Helsinki/Finland, tommi.toivonen@napa.fi

Abstract

The NAPA system has a wide variety of well-established functions for the early stages of ship design such as hull form design and naval architectural calculations. Since the introduction of the NAPA Steel package, the applicability area of the NAPA system has expanded increasingly towards basic design. New functionality is being added making it possible to perform complete basic design in NAPA including outfit design. Furthermore, interfacing with classification and production systems is improving rapidly enabling continuous data flow through the whole design process.

1 Introduction

The origin of the NAPA system is in the initial ship design. Hull form design, compartmentation and naval architectural calculations have been the main applications from the very beginning of NAPA system development. The NAPA system has been in production use for two decades, and it has reached the leading position especially in ship hull form design and demanding damage stability calculations. New applications and functions are incorporated into the system each year in order to cover a growing number of ship design disciplines. Today, the NAPA system is a comprehensive, versatile software package covering most important needs encountered in initial and basic ship design.

Napa Oy has revised its focus area for the years to come. The NAPA system shall be the market leader in the ship initial and basic design phases, and in the ship operation and life cycle management. Keys to achieving this goal are the full exploitation of the 3D NAPA ship model concept and close cooperation with leading shipyards, consulting companies and maritime authorities.

This paper presents the existing functionality of the NAPA system in comparison with the ship design process. The present coverage is illustrated by examples, the functionality yet to be realised is discussed, and some further ideas are taken up for review.

2 Ship design phases and the role of NAPA

The ship design process is usually split into initial design, basic design and production design, Fig.1. The filled areas are already supported by NAPA today at least to some extent.

Fig.1: Ship design phases supported by the NAPA system
**SHIP INFORMATION FLOW**

**OWNER’S INITIAL PARAMETERS**
- **Linked Technical Tasks**
  - Resistance, speed, power capacities
  - Compartment weight feasibility
- **Model tests**
  - Strength analyses
  - Classification simulations, prototyping

**Initial Design**
- **Spaces**
  - Geometry general arrangement
- **Hull**
  - Construction principles
  - Hull materials
- **Systems**
  - Initial system schemes
  - Main components

**Contract**
- **Coordination**
  - Spaces
  - Hull main constructions classification
  - Systems complete schemes main routings placement of main components

**Basic Design**
- **Production of outfitting areas**
  - Detailed outfitting drawings
- **Production of hull blocks**
  - Detailed hull drawings
  - Fabrication and installation drawings
- **Production of the systems**
  - Prefabrication, unit assembling and installation of systems

**Commissioning**
- **Spaces**
- **Hull**
- **Systems**

**Delivery**
- Feed back

**Operation**
- Ship operation supported by Onboard-NAPA

**Linked Administrative Tasks**
- **Planning**
  - Building strategy
  - Workload schedule
- **Procurement**
  - Indicative enquiries
- **Cost**
  - Initial budgeting
  - Offer calculation
- **Target budgets**

**SHIP DESIGN SUPPORTED BY NAPA**
- **Final building strategy**
- **Coarse planning**
- **Turn-key contracts follow-up**
- **High value material orders follow-up**
- **Long deliveries follow-up**
- **Turn-key orders follow-up**

---

Fig.2: Different phases and tasks in ship design

Fig.2 is based on the experience gained in the MOBISHIP project (Model Based Initial and Basic Ship Design). It presents the most important phases and tasks during ship design, commissioning, delivery and operation. The role of NAPA in ship design concentrates on the initial and basic design phases, while the role of Onboard-NAPA focuses on the ship operation.
Some shipyards also separate a conceptual design phase in which the ship project is handled mainly in the form of a parametric model without actual geometry. As to a conventional vessel or ship types that the shipyard is very familiar with, this rough statistical design approach is sufficient to provide the first indication of the ship’s price. However, for more demanding ship types the commercial commitments in the form of a quotation require more thorough studies, which are made in the initial design phase. Therefore, the activities during the conceptual design phase are incorporated into the initial design phase in this presentation.

2.1 Initial design before signing the contract

Initial design aims at signing the contract with the ship owner. The result is mainly documented in the ship specification, general arrangement drawing (GA) and engine room layout and in the contract itself.

The contract binds the ship designers thus limiting the freedom to do major changes. More than 50% of the final building costs are already fixed in the contract material, although initial design as such costs from one per mille to one per cent of the total costs of the shipyard. Consequently, from the point of view of fixed costs and lost freedom, initial design is the most important phase in ship design!

2.2 Basic design after the contract

The main result of the basic design phase is the approval from the class and owner, but the main purpose of the basic design phase, from the shipyard’s point of view, is to coordinate different design disciplines and space management, to avoid collisions and to minimise changes during the detail design and production phases.

Main difficulties at the basic design phase are related to avoiding collisions of spaces, systems and structures, weight control and scantling of structures. Most of the available freedom to improve the design and reduce costs is spent during the basic design phase, as the approved design material cannot be changed easily in later stages.

The basic design phase is actually the last one during which the ship is handled as one integrated entity, since detail design is normally made per system, per machinery, per outfitting area or per building block for structures.

The information created during the basic design phase is detailed enough for ship life cycle management; hence the ship life cycle management can be based on the information produced during the basic design phase. The information created for production purposes is often too detailed and contains too much irrelevant information from the point of view of operation.

The basic design phase amounts to about 2–5% of the ship’s total costs, but after the basic design phase more than ¼ of the ship’s total costs have been fixed.

2.3 Production design

Production design must follow the approved basic design material, thus there is very little freedom left to do any major changes. Even though most design hours are spent during detail design, it is not considered as critical as basic design from the shipyard’s point of view. It has become very common in the European shipyards to outsource detail design to engineering companies.

2.4 On the competition and role of NAPA

There are numerous adequate systems for ship detail design and production. Systems such as Tribon, Nupas-Cadmatic, Foran, Hicadec, GSCad, Mates, Nuovo Scafo, ProShip, Catia, Intergraph, Microstation and AutoCAD are being successfully used in production design.
Tough competition between the systems will keep the price level reasonably low while the vendors fight for market shares. Pressure from general-purpose CAD vendors is increasing, as the tools are growing bigger and more flexible. For these reasons, the penetration to detail design market is not considered attractive to Napa Oy.

As to the early design stages the situation is quite different. General-purpose CAD systems are limited in their capability to handle ship-specific studies, and only a few systems have been developed for ship initial and basic design. NAPA has been the first totally integrated ship design package based on a uniform 3D model and a consistent use of databases. In order to maintain and strengthen our market leader position, we are directing our development especially towards the area of basic design as 3D ship models are finally being exploited to improve the quality of design.

3 Design input

3.1 Owner requirements

Owner requirements set the target for the design including:

General requirements:

- Ship type
- Operation area
- Standard of vessel
- Reference vessels
- Class notation

Operational requirements:

- Mission or business idea
- Range and Endurance
- Payload
- Deadweight
- Speed
- Capacity
- Special

Special requirements:

- Maximum main dimensions
- Noise and vibration limits, etc.

Key owner requirements can be stored in NAPA into the reference system as additional parameters to be used as reference values. At the later stages, the resulting Figs derived from the project should be compared with the owner requirements. Each organisation can maintain their own standards as to the model reference system, which forms the basis of new projects. In addition, any new project-specific parameters may be added into the reference system.

3.2 Experience and statistical data

Knowledge gained from built vessels is stored into the statistical databases. Naval Architects have always been keen on relying on statistical data, which facilitates designing familiar vessel types quickly.
The NAPA system offers the following tools for handling statistical data:

- Table calculation function to create, manage and use statistical (numerical) data
- Collecting reference pictures into archives
- Collecting standard geometric definitions
- Connection to commercial relational databases
- Regression analysis tools to test formulas and correlation
- Graphics for visualisation
- Databases for archiving
- System database to store tables of interest for direct re-use
- Hull form libraries
- Hydrodynamic reference database to store

3.3 Reference vessels

An existing comparable vessel can be used both as source of input and for comparing measures of merit.

4 Ship definitions

The main geometric components in the NAPA Ship Model are the hull and other curved surfaces, reference surfaces, surface objects, rooms, structures and outfitting components. The relations of these are illustrated in the following figure.

![Diagram of ship components](image)

Fig. 3: Modelling process in the NAPA system
4.1 Surfaces

4.1.1 Hull Surface

NAPA offers state-of-the-art tools for hull form design with seamless definition tools from the first sketches through CFD studies and model testing to final fairing. Various alternative design methodologies for deriving the hull form are supported:

- Numerous transformations
- Parametric definitions
- Quick and easy from scratch
- Global and local modifications
- Patch, grid, wire frame and lofting table representation

NAPA has been involved in hull form optimisation using the modern CFD tools. The design process is now running smoothly. Several hull form variations can be designed and analysed per day by using a combination of NAPA geometry, new panel task, interfaces to CFD programs and with proper CFD tools.

The following figures illustrate examples of NAPA hull forms with panel grid defined.

![Fig.4: Hull surface panelisation](image)

4.1.2 Reference surfaces

Reference surfaces form the backbone of ship design by tying together geometric objects, aligning surface objects and compartments and by keeping the general arrangement consistent during the changes.

![Fig.5: Reference surfaces](image)
4.1.3 Surface objects

NAPA’s surface objects are topologically defined surfaces for modelling of structures. Surface objects are totally integrated with the other geometric objects in NAPA including room definitions. Surface objects support both the design of ship structures and general arrangement as well as various design analysis calculations.

Fig.6: Surface objects

4.2 Spaces

Spaces, or rooms as called in NAPA, are also topologically defined objects used for all kinds of volume-oriented purposes. Rooms support both the design of a ship’s general arrangement and various design analysis calculations.

Fig.7: Rooms are defined with the aid of reference surfaces

Ship design depends on a complex set of geometric, numerical and rule constraints. Management of these constraints is the key issue in designing the general arrangement.

The design is originally based on a space budget representing the required total area or volume per space type. The actual set of spaces is often documented in a space catalogue. The figures in the space budget should be constantly compared with the actual figures presented in the space catalogue.

The NAPA Geometry subsystem is used to model the actual spaces (rooms) whereas the Ship Model subsystem handles all information related to the spaces with the efficient purpose mechanism.

Fig.8 illustrates an example of a complicated ship general arrangement. This is to demonstrate that there are no limits to the complexity of the modelled vessel in NAPA.
4.3 Structures

Initial and basic ship design of structures has traditionally been based on 2D drafting because of lack of suitable software. Advent of NAPA Steel changed the situation dramatically, as now comprehensive 3D models can be created in a short period of time.

Fig. 9 illustrates the accumulated steel weight as a function of time as presented in the NAPA User Meeting 2000 by Nordvestconsult (today known as Rolls-Royce Marine AS, Department Ship Technology – Nordvestconsult). Modelling time has shortened even more since year 2000 thanks to the more efficient use of topology and the more versatile modelling functions.
NAPA Steel provides a 3D product model of ship structures. Unique modelling techniques ensure that the model can be created early enough to meet the needs of project and basic design. The following objects are added to the main structures: stiffeners, brackets, openings, holes, seams and plates. Objects are defined by taking advantage of topological relationships and repetition of components. In this way only a small amount of primary information is required to give the model excellent adaptability to changes.

Attributes can be defined for each object or by defining object types having their own attributes. For example, opening types can be windows, manholes or lightening holes, having properties such as unit cost, man-hours or anything the user wishes to define. Output can be graphical or numerical, applied
to the whole ship, blocks or any group of structures required. 
Numeric output can be used for instance for weight calculation, production planning and cost 
estimation. Various interfaces, including Web formats, are available for the transfer of information to 
other systems.

The NAPA Steel model can be converted to an FE model by using the NAPA FEM interface. 
Surfaces can be output as shell elements and pillars, web frames and other stiffeners can be output as 
beam elements. All attributes from the NAPA Steel model are available to be transferred to any FE 
solver.

NAPA Steel drawings can be completed as classification drawings by exporting the basic drawings to 
a drafting system by using for instance the DXF or IGES interfaces. However, a complete drawing 
system is under development, which will enable efficient classification drawing generation directly 
from NAPA Steel.

4.4 Outfit

The NAPA Outfit module extends the modelling functions of NAPA to arbitrary components such as 
machinery, lifeboats, pipes, ducts, masts, accommodation, and so on. The geometric modelling of 
these components can be carried out with NAPA, but it is more preferable to rely on externally 
provided models imported via system-specific formats, e.g. as Nupas-Cadmatic models or by using 
neutral exchange formats such as IGES, DXF or STEP. According to the openness policy of NAPA, 
the existing information should directly be re-usable. The primary source of information can be 
outside NAPA, e.g. the material system or the electric design system. Data can be shared, for 
e.g. example, by using relational databases. The role of NAPA is to help in managing the whole, and in 
generating output where information from different sources is combined.

A unique feature of NAPA Outfit is the placement of components within the main model by 
topological relationships to steel structures or to other components. The purpose is to have a model 
that easily adapts to changes made to the main geometry. The equipment can be used for visualization 
in a 3D form or included in arrangement plans and similar drawings. The information can be used for 
producing reports of various kinds, including weight and cost calculation.

Fig.11: NAPA Outfit model
5 Calculations

5.1 Weight and cost

5.1.1 General principles

The weight and cost calculation system helps to manage the weight and cost data and calculation rules, and performs calculations of the centre of gravity and the weight distribution. The results can be output by means of versatile numerical and graphical output functions. The organisation of weight and cost groups and the calculation rules are given as input data, and are stored in tables forming the so-called weight and cost model. In the design of the model, one can rely on the ship model for providing measures such as volumes, areas, and locations used for estimating or calculating weights and costs, the centre of gravity and the weight distribution.

The weight calculation is based on the geometry of the ship model, and the link between the weight tables and the ship model is automatic. Consequently, any changes in the ship’s geometry automatically lead to updating of the weight. Parallel with the weight calculation, attributes such as cost can be handled. The cost calculation can be based on the weight of structures, the distribution of interior areas, single components etc. In order both to limit the access to the weight and cost calculation and to prevent their unintentional editing, the weight tables can be stored in a separate database, called protected database.

An accurate lightweight distribution, to be used for strength calculation in the loading task, is generated by a simple command.

5.1.2 Calculation

The weight calculation system is useful for various degrees of accuracy: from the initial estimate to a complete detailed calculation.

In the different stages of ship design, the accuracy with which the ship is described varies, for which reason the accuracy with which the weight calculation can be done also varies. Adaptation of the weight model to different degrees of accuracy should preferably be done within the same weight hierarchy, in order to preserve the connection between the different stages. This can be accomplished by adjusting the degree of detail or by gradually refining the individual weight estimates.

In the following, different calculation methods for use in different phases of the design process are described:

1) **Quick weight estimation based on the main dimensions and the central ship-related parameters**

   A ship model is not necessarily needed for the first weight estimate. The weight estimation can be done utilising only the table calculation provided in NAPA. At this stage, input typically consists of statistical data. The weight calculation is initiated by copying a weight model from a reference ship. All primary ship-related data - such as the main dimensions, the power of the main engines, the number of cabins, etc. - are stored in a table of parameters. By updating the ship parameters and the unit weights, a first weight estimate can be obtained very quickly.

2) **Calculation based on surface areas and volumes derived from the ship model**

   Utilisation of the ship model substantially increases the accuracy of the weight calculation. The weight of steel structures and interior constructions is calculated on the basis of areas and volumes derived from the ship model. The weight calculation is automatically updated
when changes are made to the geometry of the vessel. A higher accuracy in calculating the centre of gravity is achieved by linking the weight calculation to the ship model. The ship model can also be utilised in defining the location of single weight components, e.g. engines and other equipment. Components can be attached to surfaces or rooms in the ship model dynamically, by means of advanced tools for defining the placement of the weight component.

3) Detailed weight calculation based on the NAPA Steel model, equipment tables etc.

When the structural model is constructed in NAPA Steel, the weight of the steel structures can easily be transferred to the weight calculation task. The calculation based on statistical data and the calculation based on actual steel structures can preferably be made within the same tables, as this enables efficient comparison of the results obtained with the two methods. Equipment such as engines, lifeboats, mooring equipment etc. can be managed in the equipment tables in the Ship Model task. Detailed information on, e.g. the dimensions of the equipment can be given. The weight and the centre of gravity of pipes and canals are calculated accurately by routing the pipes and canals in the ship model. Using the same method as for surfaces and rooms, attributes such as unit weight are given in the weight tables.

Fig.12: Graphical user interface for the weight and cost calculation
5.2 Other calculations

NAPA contains a comprehensive set of calculation functions to support all essential studies during the initial and basic design phases. In addition, many of these functions are useful also in the detail design phase. The calculation functions included:

- Speed and power performance including vibration excitation
- Loading conditions
- Hydrostatics
- Capacities
- Tonnage
- Damage stability
- Container loading
- Grain stability
- Offshore structure stability
- Manoeuvring
- Seakeeping
- Finite element pre-processing

It is assumed that the reader is familiar with these calculations; hence they will not be presented in more detail in this conjunction. Anyway, once the model is ready, the actual design studies can be made in a very quick manner, because most input data is already available in the model. The new NAPA Manager concept will automate the performing of standard calculations preparing output presentations.

6 Data sharing and exchange

6.1 General

No matter how integrated the design systems, there will always be a need to exchange and share design information with other designers, design systems, other design companies, authorities, and so forth. There can be several hundreds of different software systems in use at one shipyard. It must be possible to efficiently share the key design information with all relevant tools, thus the sharing and interfacing of design information is of crucial importance.

The NAPA system supports more than 40 interfaces of different kind, including:

- Neutral formats such as IGES, DXF, STEP, VDAFS, IDF, etc.
- Proprietary formats such as Nupas-Cadmatic, Tribon, Autokon, etc.
- Direct interfaces to Tribon, Foran, Nupas-Cadmatic

6.2 STEP interface

The STEP interface has been a standard feature in NAPA since Release 2001.1. Today, the STEP interface is the most comprehensive alternative for transferring complete product models. NAPA Steel models have been exported for instance to DNV Nauticus and GL Poseidon systems in STEP format.

6.3 Application Programming Interface (API)

A completely new alternative to interface with NAPA has been introduced recently. This is based on the Application Programming Interface (API), which makes both the NAPA project databases and the NAPA system functions accessible to other systems. Already now API covers a huge amount of possibilities to import and export data between NAPA and external applications.
6.4 NAPA Steel - Nupas-Cadmatic interface

The NAPA Steel - Nupas-Cadmatic interface is currently under development by Numeriek Centrum Groningen. The interface is based on the NAPA API and it enables the transfer of the complete constructional topology of the NAPA Steel model to the Nupas-Cadmatic 3D topologic model. According to NCG, the interface offers the Nupas-Cadmatic users flexibility during the initial stage of detailed design where a substantial amount of man-hours can be saved.

6.5 NAPA Steel – Tribon interface

The new interface generates Tribon input data from the NAPA Steel model. The generation of the Tribon model is based on the topology of the Steel model, which means that panels and stiffeners are defined referencing to other panels when possible. The interface completes the generated model by adding some additional details not yet supported by NAPA Steel e.g. cutouts and notches. Project or company specific design rules and naming conventions can be applied using control tables. The interface offers Tribon users the following advantages:

- Uniform structures for all building blocks
- Number of human made errors can be reduced
- Tribon modelling can be started at a later stage
- The amount of working hours can be reduced
- The style does not vary between scheme files

![Fig.13: A Tribon view to details generated by the NAPA Steel – Tribon interface](image)

6.6 ABS SafeHull Interface

The next generation of ABS SafeHull builds the model in NAPA by design templates. SafeHull provides the interface to the appropriate ABS Steel Vessel Structural Rules. Furthermore, the SafeHull model will be directly available in NAPA Steel to be used for various purposes. The same model is also used for 3D finite element analysis.
7 Conclusion

The NAPA system covers well all the central needs both in ship initial and basic design. The actual extent of use varies very much among the NAPA user organisations, and today only a few leading shipyards take the full benefit of the NAPA ship model concept.

Keeping in mind Napa Oy’s revised business strategy, the functionality of the NAPA system will be extended especially in the field of basic design. We are confident that the NAPA core technology suits very well to the modern needs of ship design, and the business perspectives look promising for a small and flexible company like Napa Oy. Therefore, we anticipate seeing increasing numbers of NAPA users also in the basic design departments.

The NAPA system will be further developed together with our key customers. We have already a good set of shipyards, ship engineering companies and maritime authorities with direct relations to guide the development. However, we would like to invite even more companies interested in cooperation to contribute to the development of the NAPA system.
Linking Knowledge to Product Data

Robert Bronsart, Ulf Cantow, Venizeslav Petkov, University of Rostock, Germany
{robert.bronsart,ulf.cantow,venzi.petkov}@mbst.uni-rostock.de

Abstract

A method for modeling knowledge is presented. The implementation can be utilized in the design process of ships. The data definition language is used to model rules and regulations as objects themselves. This approach allows for a direct relation of rules, expressing knowledge, to relevant product model data. A semi-automatic translation of the rule as well as the product model schema to java class definitions forms the basis for an operating system independent implementation. The developed system architecture to be used in a distributed working environment is described to some detail. Two fundamental use-cases: knowledge definition and knowledge utilization are discussed. An example from the ship design process serves to demonstrate the implementation strategy and shows functions of a prototype system.

1. Introduction

Despite the increasing utilization of first-principle design methods, ships are still built according to rules and regulations, which are developed and published by many organizations. These rules address different aspects in design and construction as well as in operation of ships. They represent knowledge to be applied in the different processes. The degree of complexity of the objects as well as the corresponding relevance to society, e.g. safety and environmental aspects, result in an increasing number of nationally and internationally agreed upon regulations to be fulfilled. In this context, ships operating in worldwide trade have to be regarded as complex products.

Many different tools are used throughout the whole life cycle, starting from the very early design to the operation including maintenance and survey. All classification societies offer CAE-tools, which are to be used in the detailed design process. The most elaborated functions are offered for deriving the required scantlings of the structural steelwork. An integration into the design process is only partly given by “simple” data exchange methods.

Parallel to these tools, rules and regulations to be observed in the process steps are published in books, either traditionally printed and distributed via mail or published via electronic media, sometimes adding extra functionality like searching, checking for relevance etc., BRONSART, R. (1999). For the design of the product itself, high sophisticated CAE-tools are used at all shipyards and design agents. Apart from the mentioned class tools, there is almost no functionality offered which help the designer to (automatically) observe rules in the design process: a major gap still exists between product definition (CAD) and product design (engineering work). Even first principle design procedures, utilizing direct calculations like FEA-Methods require data to start the calculations with. In many cases rules based scantlings are used for this.

2. Analysis of Technical Rules Representing Knowledge

Rules to be observed in the design and construction of ships can be categorized under different criteria. Either flag state or international rules, rules addressing design, construction or operational aspects, regulations addressing different “technical” requirements like Load Line Convention, SOLAS, MARPOL etc. Furthermore, rules, which are released by the classification societies, have to be paid attention to in the design and construction of ships. Whereas the first ones are agreed upon on IMO (= international) level, the latter ones are to a large extend specific to the different class societies. The common approach is that rules and regulations are related to a product: ship or part thereof. Products can be defined by product data, which means that technical rules have to be related to product data.
2.1. General Model of Rules Expressing Knowledge

A rule for a product can be considered as a logical expression, which is TRUE if according to the rule and FALSE in the opposite case. Therefore a rule is a logical function

\[ f: \{\text{Object}\} \rightarrow \{0, 1\} \]  

with values TRUE or FALSE. Each function has a definition set, which in this case consists of objects for which the rule can be checked. For convenience this definition set could be extended in the following way: we assume that a function representing a rule is always satisfied for objects to which the rule is not applicable. This assumption allows an arbitrary rule to be checked for any object.

Technical rules, for instance the rules of the classifying societies such as Lloyds Register, *Lloyd’s Register* (2002), Germanischer Lloyd, *Germanischer Lloyd* (2002), and others consist of a large set of rules. The conformity of a product, for example a ship, to such rule sets means that all logical functions introduced above have to result in the value TRUE. Thus conformity of a product according to defined rule sets results in the check of a single boolean function to be of value TRUE. This function is the conjunction of all single rule logical functions:

\[ F(P) = f_1(P) \land f_2(P) \land f_3(P) \land \ldots \land f_n(P). \]  

In the “classical” rule books of the classification societies different types of rules can be found. Some rules define variables equal to some mathematical expressions, other require just an answer of type “yes” or “no”, or simply give an advice. The most of the rules in these books are of type: “ IF ‘condition’ THEN ‘requirement’…”, JENSEN, H. (1992).

This construction is well known in the theory of knowledge based systems as a “production rule”. “Production Systems” also called “Rule Based Systems” are based on the idea of the mathematician E. Post in 1943 who tried to solve problems introducing production rules. In the artificial intelligence theory there are three important knowledge representation methods: frames, neuronal networks and production rules (IF-THEN rules). Today IF-THEN rules are a popular method used in artificial intelligent applications.

A knowledge based system is a software system, where the solving of a problem will be separated outside the standard control structures of the program languages. Such systems consist in general of two main parts known as knowledge base and inference machine. The common problem is the knowledge acquisition in the knowledge base and then using the inference machine to produce new knowledge and add it to the knowledge base. Considering the problem of modeling and utilizing the technical rules in such knowledge based system we state the necessity of a common rule model, which is independent of a concrete rule. The inference machine that is responsible to produce new knowledge we reduce to the task of checking a product against the rule, and not to create new rules using the existing ones. This approach will not be considered here, although the problem is very interesting for further research.

As a consequence, the problem of implementing technical rules and regulation in a computer based environment could be treated as a task of applying knowledge based systems.

2.2. Structure of technical rules

2.2.1. Document Structure Representation

In many cases, the document structure of rules and regulations is given by an alphanumeric coding principle, rules are categorized into different sections. These categories represent subsets of rules named rule modules. A rule module consists of rule modules and/or rules. This type of structure can be represented as a tree structure. The elements of the tree structure are rule modules and rules.

2.2.2. Logical structure

A different method to represent rules is their structuring according to the logical content and applicability. In this approach, two main aspects have to be considered:
- The applicability to a product or parts thereof. For this, the product structure has to be uniquely defined. A product can consist of underlying products, defining any number of product levels. The product qualities are described by product parameter, STEP.
- The functionality of the corresponding rule. Here, basically three different types have to be distinguished:

  **Definition:** This rule serves to define parameter. It is used in logical or arithmetic assignments. A parameter has a relationship to a product, a rule or a rule module. In most case such parameters are related to a concrete rule. The relationship defines the scope of the parameter.

  **Advice:** Rules might have an advising character. Their function is to provide additional information to the designer.

  **Criterion:** This type of rules is also called “IF-THEN rule” introduced above. A criterion supplies a result of a rule call (rule evaluation). The criterion has the form: IF precondition is fulfilled THEN condition must be fulfilled. Precondition and condition are logical expressions. The linkage of both expressions with logical IMPLICATION is the result of the criterion. In a rule, several criteria might exist. The result of a rule is the linkage of all criteria. Some important applications are the comparing of a value against a predefined range or an exactly given value.

All types or subset can be represented in a single rule.

2.3. Computer Based System of Technical Rules

For a computer based processing, technical rules representing knowledge must be transformed in an adequate form. At least the following functions have to be provided:
- reliable storage with efficient access,
- effective maintenance including quality control,
- navigation and presentation capabilities,
- check for consistency.

An interface will serve for the integration with external software tools. The system architecture will have to provide for a distributed, network based working environment.

3. Rule Modelling

The modelling of data necessary for the representation of technical rules is performed using the implementation independent, object oriented data definition language EXPRESS, ISO 10303-11 (1994), the diagrammatic representation of the data model is done with EXPRESS-G.

Different schemas are used to clearly separate distinct aspects, thus allowing to formulate models with precisely defined interfaces.

Some key objects of the model are introduced in the following.

A **Rule_object** represents a data structure (entity) to model rules and their relationships. The attributes of this object are:
- **name_**: which serves to identify a Rule_object
- **structure_**: represents the document structuring part of the Rule_object e.g. paragraph, section, ... By these two attributes, the traditional book structure of complex rules and regulation documents are expressed.
- **representation_**: points to a location where a detailed representation of the rule itself as well as its content can be found, e.g. position in a classical rule book. The document link is modelled in detail in form of a uniform resource locator (URL).
- **container**: references to an **Engineering_rules_module**; it provides the tree structure of the set of rule objects, which is not necessarily the same structure as given in the “book” view.

The **Engineering_rules_module** is the model construct of a rule module (see also Rule_object). It may consist of an arbitrary number of rules and rule modules, see figure 1.

![Diagram](image)

**Fig.1: Entity RuleObject**

The **Engineering_rule** entity represents a rule for a technical object, e.g. ship or part thereof. It is described by the following attributes:

- **valid_for**: defines the validity of a rule which respect to a (set of) product type definition, by this, the scope of the rule is defined.
- **definitions**: definition of parameters which are used in the rule expression internally.
- **criteria**: serves for the modelling of IF - THEN type of rule. With help of this most important construct, the applicability of a rule can be judged upon as well as parameter of a product description can be calculated and/or checked against predefined values. Whereas the other parts of the model are mainly being used for searching and navigation based on a textual representation of rules, this attribute, a set consisting of any number of different criteria, will serve for checking of conformity based on given product data (see also under **Expression**).
advices_: serves to identify additional information to be observed, this is basically a link to a document.

The model construct Expression represents an expression to be evaluated in the context of a rule. It is used for the precondition of a criterion as well as for the modelling of the condition itself. Apart from attributes to identify or describe the expression (name_ and description_), the expression object is modelled by the following attributes:

- **interpretable_script**: is a text string which can be evaluated by an interpreter, in our implementation the intention is python syntax (Python [5]) to be used.
- **statements_:** a set of statements which are referred to in **interpretable_script_.** A statement either represents the result of another rule evaluation, a parameter or an expression. It has to be noted that this can build up, networks of referenced rules of any complexity, which are evaluated as a single script. An example for this is the “typical” definition of rules in terms of intermediate values which are given by other rules, which in turn refer to other rules …
- **valuetype_:** indicates the value type of the expression

---

**Fig.2: Entity Expression**

4. Implementation Methods

The basic functions of a system implementing technical rule in a computer based environment have to allow reliable storage and accessing of the data representing rules, to provide search and filtering methods for an efficient work, to support different exchange formats (e.g. XML , EXTENSIBLE MARKUP LANGUAGE, STEP ) and to be flexible and scalable according to changing requirements. This could be achieved by an open distributed software architecture. The global network Internet makes it possible for a client computer to communicate with a server using different technologies. The advantage of this architecture is a central storage, assuring the most actual information. Continuous development of services extending the functionality supplied by the server will be accessible by all clients.

A Rule Server System under development at the Institute of Maritime Systems and Fluid Engineering is implemented in Java , JAVA™. Java offers an excellent support for distributed applications (CORBA, Java RMI), exchangeable component structure (Enterprise Java Beans) and DBMS integration (JDBC).

One important part in the overall system architecture therefore is the database management system (DBMS). The main task of the DBMS is to provide reliable persistence mechanism for the server.

For an efficient support of the design process steps, the following functions have to be supplied by a computer based Rule Server System. Principally two use cases have to be distinguished: rule definition and maintenance by rule publishing organizations and rule usage by e.g. shipyards and design agents. In the first case, emphasis is put on efficient functions to define rules while e.g.
observing consistency with already existing ones. Furthermore, rules have to be maintained in versions to allow for using appropriate rules according to certain key dates. For use case “engineer/designer”, the necessary functions are discussed.

![System Architecture Diagram](image)

**Fig.3: System Architecture**

- **Searching of relevant rules**
  Different search strategies are required, which provide for the navigation in complex rule and regulation systems. A simple approach is the mapping of existing rule structures according to the printed layout in rulebooks. By this, the contents of rules can be navigated by predefined links and viewed similar to the traditional way. Functions to navigate the rules according to the product structure represent an approach, which is more suitable for the design process. Rules are found based on a structured catalogue of product data in terms of parts, systems, subsystems etc. specific to the product, the rules are to be applied to. By this, rules can be found and thus taken care for which in the first case might be overlooked as the document structure not necessarily presents all relevant rules at the same location.

  Additionally, key words can be used to find rules relevant to specific aspects, parts, components or systems. For this, rules have to include applicability information, which is part of their meta information.

- **Checking of a design against a set of rules**
  An existing design has to be checked against all rules applicable. In this case, the relevance of rules is the important issue. In a second step, design values can be checked against those derived from the relevant rules.

- **Dimensioning according to rules**
  In the design process, the rule server is to be used to derive design parameter, e.g. scantlings of the structural steel work or design parameter values of machinery components, according to relevant rules. In this use case, the product is not completely defined and the functions serve to finalize the product data definition.

5. **Integration Strategies**

Different integration strategies can be followed for a workbench in form of a CAE-system environment: integration into existing software tools (process-process communication) or supply of applications used by the designer (man-machine communication).
5.1. Integration with CAE-Systems

The open, platform independent system architecture and implementation strategies followed allow for the integration with CAE-systems in different ways. Two scenarios are proposed below:

- An external application communicates with the rule server and CAE-System utilizing existing interface functions and serving as a proxy. A special interface for each different CAE-Systems is necessary which is to be built on top of the offered application programming interface. For this software layer, third party tools exist for major CAE-Systems, though providing a standardized access to the systems. For CAE-Systems specially designed for and used in the shipbuilding industry, this type of common interface is not known.

- A second approach is to integrate directly the use of the rule server interface as extension or plug-in in the CAE-System, providing the functionality of the rule server in the CAE environment.

5.2. Application Services (Rule Server System)

Client applications are used to query the server application and, if appropriate, directly perform rule checking or dimensioning of parts or components. In this case, results out of these processes have to be taken care for on a manual basis. This type of application obviously is less complex to be implemented but will also increase productivity in the design process considerably. The figures below display some screen shots of prototype implementations of different clients. The client applications communicate with the server application “Rule Server” to supply the information asked for as depicted in figure 3.

Fig.4: Prototype Application “Rule Editor”
5.2.1. Rule Server

This is the main part of the system responsible for the storage, maintenance, and accessibility of the rules and their data. Provides main functionality such as navigation, search, check, and other through an interface. A python interpreter implemented in java (JYTHON) is used for evaluating expressions.

5.2.2. Rule Server Mapper

The Rule Server Mapper is a client application that uses the rule server interface to map external data in the rule server.

Product data created in PLIB, PLIB and stored in STEP format are used by the Rule Mapper as a source for creation product information on the rule server. This data are of primary importance for the logical content of the rules in the system.

The creation of the objects representing the rules on the rule server (Engineering_rule and Engineering_rules_module) is a result of a semiautomatic process starting from the textual source. A script converts it into a representation that confirms STEP AP21. In a second step, the Rule Server Mapper creates the corresponding objects. Unfortunately the logical content of the created rules remains empty and requires human intervention.

Specific product data, represented in conformance to AP 226 and by this linked to the stored PLIB data model can be uploaded on the RuleServer for evaluation processes.

5.2.3. Rule Editor

As mentioned above adding of knowledge (logic) to rules can be made only manually. The Rule Editor is a tool, which makes possible to set up the applicability of a rule, to define values introduced by the rules as well as expressions that have to be evaluated. So it makes possible to define the logical statement of the rule.

5.2.4. Rule Navigator

![Rule Navigator](image)

Fig. 5: Prototype Application “Rule Navigator”

Fig.5 shows the screen shot of a client “Rule Navigator”, which offers access to the rule base for searching and navigation purposes. Here, the rule server is configured to work with class rules. The upper left window displays the book structure, highlighting the specific rule which content is shown
in the right window, in this case, the minimum diameter of a shaft being part of the main shaft line.

Fig. 6 shows the predefined product structure which can be used to filter the rule set according to specified product components. This view is generated automatically based on the rule data according to the above described rule data model, which in turn refers to the underlying product model (valid_for_). Filtering can be applied to all product parameter values. Additionally, filters on rule date and versions as well as on book structure can be developed and applied in form of a condition like AND or OR.

![Image of Product Type Filter in “Rule Navigator” Client](image1)

**Fig. 6: Product Type Filter in “Rule Navigator” Client**

![Image of Prototype Application “Rule Checker”](image2)

**Fig. 7: Prototype Application “Rule Checker”**

### 5.2.5. Rule Checker

Fig. 7 shows the screen shot of the prototype “Rule Checker” application. Here, the intermediate shaft of a shaft line is checked against the minimum diameter. The log (right window) displays the rules found to be applicable to the given product data and therefore used in the checking process. The input values as given in the product description (used Parameter), the required values according to rules (calculated Parameter) and the overall check of conformity are shown. In the checking process, rules are classified into different categories: those which are used for checking, which are fulfilled and those which are not fulfilled rules. Furthermore, advice type rules are identified and can be visualised, navigating the corresponding button. In this example, rules are found in the underlying database that
can not be evaluated based on the given input and are classified as advice (Advice Rules). All rules can be directly viewed by the links to the client “Rule Navigator” which in these cases is triggered to display the corresponding subset of rules.

Acknowledgement

The work presented in this paper is supported by the German Federal Ministry of Education, Science, Research and Technology under grant 03SX133A.

References


GERMANISCHER LLOYD (2002), Rules and Programs, Hamburg

JENSEN, H. (1992), Überprüfung der Anwendbarkeit und Einhaltung technischer Regelwerke sowie Dimensionierung an Hand von Regelwerken mit Hilfe von Rechnern, Bericht 528, Institut für Schiffbau der Universität Hamburg, in German


EXTENSIBLE MARKUP LANGUAGE, (XML) 1.0 (Second Edition), http://www.w3.org/TR/REC-xml

JAVA™ technology homepage; The Source of Java™ technology: http://java.sun.com

BIGUS, JENNIFER; BIGUS JOSEPH (2001), Constructing Intelligent Agents using Java 2; John Wiley and Sons, New York


PYTHON, http://www.python.org

JYTHON, http://www.jython.org


TOPARLAK, I.A. (1995), Knowledge based processing of technical rules and regulation at the example of the shipbuilding rules, Bericht 552, Institut für Schiffbau der Universität Hamburg, in German
Intervention of Robots in Shipbuilding and Maintenance

Peter Weiss, Cybernetix, Marseilles/France, peter.weiss@cybernetix.fr
Frédéric Schom, Cybernetix, Marseilles/France, frederic.schom@cybernetix.fr
Yves Chardard, Cybernetix, Marseilles/France, yves.chardard@cybernetix.fr

Abstract

The application of robotic systems became a standard in many industrial fields during the last two decades. These systems prevent the human worker from health damaging work and have proven to increase efficiency and precision significantly. Ship building and maintenance include many activities which are known to be hard and sometimes health damaging for the personnel working on site. Despite of this fact few automated systems have found its application in this area. The current paper will present some first approaches of the introduction of robotic systems on European Shipyards. The obstacles and drawbacks for these systems will be illustrated by presenting two recent European projects on robotics for shipbuilding and maintenance. Furthermore an outlook will be given to future approaches in this fast developing field.

1. Introduction: A need of automation?

The last decades have witnessed the emergence of robotics in many industrial branches. Automobile industry, microelectronics or food industry are some examples for the relieve of human workers by robots. The factors that promote the automation of industrial processes can be divided into three fields: economic factors, technological factors and ergonomic factors: The economic factors are obvious: Process automation decreases the fabrication times and thus the costs of the units produced. Technological factors can be the precision or quality needed in production that can not be achieved by manual production means like for example in the fabrication and assembly of micro-technological devices. And ergonomic factors are the relieve of the human worker of health damaging or tiring work. These aspects help to evaluate if an industrial process is suited for the application of robots or not.

In the field of ship production and maintenance several processes are thinkable to be automated by robots. Fig.1 shows the different phases during the lifetime of a ship. This paper will deal with the activities starting with the assembly of the ship hulls parts even if it is well understood that other activities precede to this assembly. The assembly of the ships parts is followed by the welding of the different parts and finally the painting of the hull and its interior.

These activities are then followed by the construction of the ship's interior– the outfitting, Andritos and Perez-Prat (2000). This last step marks the main difference between ship building and other industries where robots are largely integrated in the fabrication: A ship is a customized product. Low-volume production calls for flexible automated systems that can be easily adapted to each fabrication case. Long programming times to adapt the system present thus the principal drawback in such a case, Andritos (2000).

During the operation of the ship, maintenance appears on a regularly scale. It includes the inspection of the ships parts like the hull to control corrosion, damages or vegetation. This can then be followed by the cleaning of the hull to remove algae, shells, oil or rust. Another operation that is included in the maintenance is the repainting of the hull, either to replace recently removed coating or to add an additional one.
2. First Approaches

Several areas in the domain of shipbuilding and maintenance offer possibilities for the application of robots. Following chapter will state some examples of projects in this field and present two projects in detail that are and were developed with the participation of CYBERNETIX in the frame of European projects.

2.1. Shipbuilding

2.1.1 Robotic systems in Shipbuilding

In the area of ship building welding activities on the dockyards offer possibilities for the application of robots. Robotic welding systems have proven its feasibility and efficiency compared to a human welder in many other fields like the automotive industry. In ship construction the pre-assembly of the hull blocks and the final assembly of the blocks host a potential for automated welding systems. An example project that is aiming at the assembly of large steel structures, not particularly for ship building, is the NOMAD project, Peters et al. (2002), NOMAD homepage under www.cordis.lu. NOMAD presents a novel approach on the large-scale and low-volume steel fabrication. The goal is the automated assembly of customized steel structures in by automated vehicles in a construction cell equipped with a vision system capable of determining the localization and orientation of a structure. Thus this project could be an example for the automated pre-assembly event when it is not directly intended for the ship industry.

2.1.2 Welding of assembled blocks: The DockWelder project

DockWelder, http://www.mip.sdu.dk/dockwelder/, is a GROWTH project coordinated by APS (D) with the participation of ODENSE STEEL SHIPYARD (DK), FINCANTIERI (I), MIP (DK) and CYBERNETIX (F), Its objective is to develop an automated and moveable platform equipped with a manipulator arm and a welding robot intended to weld the joints between the different hull sections during the assembly phase of a ship. This project is also a feasibility study to test new technologies and their possible use in modern ship erection. Much importance is given to the application on the shipyard which implies the environmental constrains and the usability by the personal working in this field.

The DockWelder robot is intended to be used at the later stage of the shipbuilding process where the ship sections are closed and ready to be welded. The system is a vehicle, moveable by the operator, that can be positioned inside the ship structure. Once positioned, it is able to execute the welding jobs
automatically by finding its own position in the working area and by following the welding line. Fig.3 shows the assembly of a vessel by the different blocs on the Odense Shipyards.

![Assembly of the ship hull.](image1)

![The system components of DockWelder.](image2)

The global system can be separated into three main components:

- **The Placer** is the systems base. It ensures the transport of the robot and the stability during the welding process.
- **The DT-VGT** is a hydraulic manipulator consisting of two modules. It serves to position the welding robot in the working area while the Placer is stabilized.
- **The Welding robot** carries the welding torch and the sensors necessary to follow the welding line. A standard industrial robot will be used for this task.

The vehicle carrying the manipulators is called Placer. The main functions of the Placer are to ensure the transport of the vehicle and to stabilize it during the welding procedures. The transport onto the ship will be done via a crane. But once inside the working area it can be moved by its motorization at the rear wheel. A electric driven motor that is controlled by the operator was chosen. The project refrained from an automatic displacement of the vehicle seen the various obstacles inside those welding areas and due to security reasons. The system will be stabilized via three hydraulic cylinders that are lowered to the ground during the welding procedure.

![The DT-VGT.](image3)

![3D Model of the DT-VGT.](image4)

Once stabilized, the path following will be executed by three different manipulators: Two similar DT-VGT modules and one industrial welding robot. The DT-VGTs enlarge the workspace of the system. Together both manipulators have a reach of approximately two meters. Thus they are used to position the robot before the welding phase. During the welding process itself the DT-VGTs are not moving. The welding robot is a 6 axis Motoman SV3X. It carries the welding torch for the MIG welding and the welding sensor that is used to follow the gap between parts to be welded. The wire feeder is
The system will be controlled via a control station that includes the PCs to command the different manipulators, to receive and treat the data coming from the various sensors and a motion planner to calculate the welding jobs. The Man-Machine-Interface foreseen for the robot will be simple to use to facilitate the use for the operator. It will be similar to MMIs that are known for industrial robots nowadays. Simplicity will be a primordial factor if DockWelder shall be accepted by the personal working at the face.

Several sensors ensure the location of the robot in the working area and the path following of the welding. After the positioning of the vehicle a laser scanner will be used to scan the environment by measuring the distances to the walls under several degrees. The position of the vehicle inside the room is calculated by comparing these measurements with the 2D structure of the working area. Fig.7 shows data results during development work of this device. A rectangular room is represented by the different dots. The vehicle's position is the center of the circle. In the case the system is not able to calculate a position due to obstacles obstructing the view, a control loop was introduced enabling the operator to exclude an angular field of the measurements.

Fig.7: Measurements by the positioning sensor.  

Fig.8: The main components of Octopus.

The second position sensor of the system is the welding sensor which is an optical device detecting the gap between the welding parts. During the welding process the adaptive control mechanism is comparing the geometrical data of the joint area picked up by the laser sensor with the parameter sets pre-calculated for this job. The acquired data is used to adapt the parameters of the welding robot and its integrated components like the welding torch and the wire feeder.

A typical welding procedure of DockWelder can be divided into different steps:
- **Positioning of the vehicle**: The vehicle is positioned by the operator inside the working area. To facilitate this procedure, visual markings on the floor can be foreseen.
- **Localization**: The laser scanner is activated to take measurements of the room. The systems position is calculated by using this data. In the case the position is not correct and the manipulators are not able to correct this error, the vehicle has to be adjusted manually.
- **Stabilization** via the three stabilizing legs. The hydraulic and electric connections are established after this stage. The fact that the connections are only established after this stage give the system a maximum of flexibility.
- **Pre-positioning** of the manipulator by a further rotary hydraulic actuator and the DT-VGT.
- **Welding procedure** by the Motoman SV3X including the adaptive process control by the welding sensor.

The different welding jobs are preprogrammed and can be retrieved by the operator.
The DockWelder project reached its mid-term in January 2003. The different components have been chosen and a overall design study of the system was done. The challenging part to come until Summer 2003 will be the integration of the modules into the entire system.

2.2 Maintenance

2.2.1 Examples of robotic systems in ships maintenance

In maritime industry, maintenance and repair of ships are operations that have to be performed for several reasons: Safety, efficiency, reliability and aesthetic. Some of these activities offer applications for robotic systems like e.g. the inspection of the ships hull, the blasting or even cutting of metal sheets and the painting. Robots present major advantages in these fields in comparison to the human worker not only in terms of costs and efficiency but also in terms of the necessary procedures: These activities are done nowadays mainly in the dry-dock or with divers (in the case of the inspection). In particular the ship's hull and topside cleaning are performed manually by specialized teams in unsafe and tedious conditions. The standing times in the dry-dock furthermore present significant costs for the companies. Robots could be able to replace the human worker in these fields while even doing the tasks when the ship is actually in the water. This is particularly of interest for large floating structures like for example FPSOs in the oil industry where maintenance in a dry dock is practically infeasible.


The objective is the control of the ballast tanks of double hull tankers. The solution that was developed by the consortium is a small size ROV equipped with vision and NDT sensors for inspection and material testing purposes.

Another project as example in the field of ship hull cleaning is AURORA of the IAI (SP), Algosystems SA (GR), Lund University (S), Suministros y Aplicaciones Industrials S.A. (SP), T. Kalogeridis & Co Inc (GR), Union Naval de Barcelona S.A. (SP) and the Riga Technical University (LA). Its goal is the development of a underwater climbing robot for the cleaning and inspection of the ship hulls.

2.1.2 HP Water-jet Blasting and Painting of Ship hulls: The Octopus project

A similar project to the problem of the hull cleaning is the Octopus project. It represents a consortium between CYBERNETIX (F), LCO.S.R.L (I), R.G.I Resource Group Integrator (I), LISNAVE Shiprepair (P) and UNINOVA (P). A remote controlled maintenance system was developed in the frame of this project. It consists of a carrier vehicle equipped with permanent magnets enabling it to crawl along vertical surfaces. A high pressure water blasting tool is fixed at the front of Octopus, that serves to wash or blast the surface of the hull. The major advantages of the permanent magnet are the high magnetic force per volume and the fact that the vehicle does not fall down in the case of a power breakdown.

The basic functions of the Octopus is the washing and blasting of large surfaces like the ship's hull. The surfaces that can be treated can be either vertical, inclined or horizontal. The capabilities of the vehicle allow a treatment of approximately 80% of the complete surface of a ships hull (problems arise with small angles like for example at the bow and the stern parts). For the washing of the surfaces a efficiency of approximately 150 m²/h were reached and for the blasting 90 m²/h with a linear speed of 0.15 m/s. For movements without blasting, Octopus reaches speeds up to 0.30 m/s. The blasting is executed with Ultra High Pressure water jetting that reaches a pressure up to 2500bar. One of the
advantages of the system is the fact that the used waters and thus the waste, is recovered by the system. This makes Octopus a very environment friendly tool compared to the techniques used nowadays that imply pollution and overspray. The same can be stated for the noise pollution: Traditional hand-carried water guns emit noise levels up to 115 dB, Octopus came in at 65 dB which also represents a significant ergonomic and ecological aspect for the workers and the environment.

Two different operation modes are implemented in Octopus: An automatic mode able to generate the necessary trajectory on a given surface or a remote operated mode controlled by a joystick. At the beginning of the Octopus project a totally autonomous vehicle was aimed. But tests on shipyards with the personnel there have shown that the remote controlled mode via joystick is often more appreciated. This is why previous versions of the vehicle dispose of this mode of control. The time seems yet to early for automatic Octopusses crawling along the ship.

![Fig.9: The first Octopus with its Control Station.](image)

![Fig.10: The second version – remotely controlled.](image)

![Fig.11: Octopus working underwater](image)

Octopus represents a modular concept. It consists basically of a carrier vehicle on which different tools can be fitted. New developments were made since the first version of this vehicle. As stated above a major advantage of a robotic system is its use in areas where human intervention is difficult. A successor of the on-land version of Octopus is a underwater vehicle based on the same technology able to clean the ship's hull while the vessel is in water. Thus the time and money consuming dry-dock intervention can be avoided.

Another development was on the painting of ship hulls. A new tool is in development that can be fitted to the Octopus carrier enabling it to paint the ship hull. First results with this tool were promising and even the painting of surfaces under water seems possible due to a new kind of paint.
3. Conclusion

Robots will have a entry in shipbuilding and maintenance as in other industrial sectors before. Nevertheless this approach will differ from the systems known in industry nowadays. In view of the complexity of ship production the only possible solutions are either systems with a high degree of intelligence able to react fast on different interventions, or remote operated vehicles that relieve the workers from fuzzy work but that still need a high interaction with the human.

Acknowledgements

DockWelder and Octopus are projects under the GROWTH programme financed by the European Commission.

The authors want to thank Fivos ANDRITSOS of the JRC and Keith HERMAN of CATERPILLAR for their helpful input and discussions that helped writing this paper.

References


Simulation of Fluid Dissemination in a Virtual Reality Environment Onboard the Ship

José Miguel Varela, Tiago Santos, Carlos Guedes Soares, Instituto Superior Técnico, Lisboa/Portugal, varela@mar.ist.utl.pt, tsantos@mar.ist.utl.pt, guedess@mar.ist.utl.pt

Abstract

The paper addresses the simulation of fluid dissemination in a ship, using a 3D environment of visualization and control coupled to a simulation algorithm, originating a virtual reality application. The virtual reality environment was developed apart from the simulation algorithm using a generic architecture based on the object-oriented paradigm. The architecture allows adapting the model data structure in order to make the connection to the algorithm. The object-oriented approach is currently the most used way to achieve an efficient production of high quality reusable software. The abstraction offered by this technique hides subsets of existing complex processes and decreases the difficulty of dealing with highly complex systems. The algorithm was also developed apart from the virtual reality environment. It loads the data provided by the virtual reality application in order to run each cycle. The data is then updated and displayed in the 3D environment. The object in study is a virtual world composed by a vessel with several watertight compartments connected by doors, valves, hatches, pipes and ventilation ducts. The purpose of this virtual model is the study, visualization and control of fluid dissemination along the vessel in a casualty situation.

1. Introduction

The use of Virtual Reality techniques in the industry in general and in the maritime industry in particular has increased significantly during the last years. Most of the cases virtual reality is used to visualize data produced by simulation algorithms in a more understandable form for humans. In fact, simulation algorithms normally produce lists of numbers and tables that are difficult to interpret and sometimes even to understand when read for the first time. Virtual reality techniques provide a more understandable way of reading and visualizing this data, allowing the user to take conclusions in less time. Concerning ship operation, Virtual Reality can be very useful to visualize compartimentation and equipment of ships, namely of complex ones like military vessels. When applied in a real time simulation, Virtual Reality has the potential to significantly reduce the training costs and the risk to the users, Burdea (2000).

The present study focuses on the design and development of a Decision Support Tool, which uses Virtual Reality techniques to display the condition in the various ship compartments, which is to be used in a Damage Control Centre. This work follows-up the studies presented in Varela and Guedes Soares (2002).

The Decision Support Tool provides visualization and detailed information about the ship compartments, and the relevant equipment inside them, helping the user to take the most correct decision in an emergency situation. The tool is also able to simulate flooding in real time if connected to a specific algorithm that performs the calculation of the effect of gradual flooding. This feature provides a high potential for crew training purposes. As described in Varela and Guedes Soares (2002b), the user is also able to control some of the equipment.

The Decision Support Tool that is described here builds upon the creation of the Virtual Reality Environment described in Varela and Guedes Soares (2002a), and on the real-time development of the Simulation Algorithm reported by Santos et al (2002). It is the assembly of both the Virtual Reality
Environment and the flooding Simulation Algorithm allowing the exchange of data between them that will generate the Decision Support Tool (Virtual Reality Application) as shown in figure 1.

![Decision Support Tool Diagram](image)

Fig. 1 – Decision Support Tool diagram

This paper describes in a brief form the architecture of the Virtual Reality Environment since a detailed description may be found in Varela and Guedes Soares (2002b). The Simulation Algorithm is also briefly described. Being distinct programs, developed separately and in different periods of time, the Virtual Reality Environment and the Simulation Algorithm have particular features concerning input/output data, running cycles and even program languages that may lead to incompatible linking. Linking both programs required total compatibility between them. Some results are presented showing the evolution of the flooding process of river passenger vessel in the virtual reality environment.

2. The Virtual Reality Environment

The Virtual Reality Environment provides the interface to visualize and navigate through the model of the ship. Typically a 3D model is displayed and the user is able to navigate to any position with a desired point of view. Selection of existent relevant objects is also allowed and information about the existent objects may be displayed in dialogue boxes, which are 2D objects displayed in front of the world and typically contain required or useful information to the user.

The model of the ship includes a geometric part composed by sets of triangles that are displayed in the output device and a topologic part that describes the type of objects that each set of triangles represents, the relations between them, their state and behaviour.

2.1. Taxonomy

The object taxonomy used in the Virtual Reality Environment was kept as simple and generic as possible in order to allow adding new object types in the future. In fact, the tool is intended to load different models of ships with different type of equipment.

Since the situation to which the tool was meant to support includes fluid dissemination through the compartments of the ship, two situations were taken into consideration: the storage of fluid inside the ship and the dissemination of fluid itself. These two situations generated the first two main types of objects. The first one originated the Fluid Container Object type (FCO), which are objects that store fluid. The second situation originated the Connector Object type (CO) that allows the connection between Fluid Container Objects.

Concerning the fluid dissemination, CO are objects that may allow fluid transfer between their
adjacent \textit{FCO} (holes) or may retain the fluid inside those \textit{FCO} (bulkheads, closed doors or valves, etc.). This fluid transfer is related with the difference of fluid pressure on both sides of the \textit{CO}, i.e. on their adjacent \textit{FCO}.

The application was tested on the forward section of a passenger vessel shown in figure 2, where these types of objects can be observed.

![Diagram](image)

**Fig. 2 –** Forward zone of a vessel - Fluid Container Objects, Connecting Objects and Static Objects

![Diagram](image)

**Fig. 3 –** Connecting scheme between \textit{CO} and \textit{FCO} concerning fluid dissemination

The scheme in figure 3 shows the relations between the objects of figure 2 concerning the fluid dissemination aspects.

Although a detailed description of these types of objects may be found in Varela and Guedes Soares (2002b), the fundamental aspects will be mentioned in the following paragraphs.

\textbf{2.1.1. Fluid Container Objects (FCO)}

Fluid Container Objects are inner spaces that store fluid. Examples of this type of objects are inner spaces of compartments, tanks, ventilation ducts or pipes.

Two types of \textit{FCO} were considered: compartments and ducts. This distinction results from the difference of volume that exists between them. When considering the calculation of fluid...
dissemination onboard, ducts do not store fluid since their volume is negligible when compared to the volume of compartments. Ducts or pipes simply work as fluid transporters between their openings (that may be positioned in different compartments). This simplification reduces largely the computational work needed.

Main characteristics and attributes of these objects are the following:

- Unique identifier
- Volume (V) and geometric centre (GC)
- Two or more adjacent CO that bound the FCO
- Ability to store fluid which results in a level of water inside them

FCO do not have a geometric representation by triangles like CO do. Hence, these objects are visually represented by their adjacent CO. Concerning fluid dissemination FCO have information about their characteristics and may obtain information about their adjacent CO.

2.1.2. Connecting Objects (CO)

Connecting Objects or simply Connectors are objects that bound FCO and at the same time connect them. Examples of these objects are bulkheads, doors, valves, hatches, etc.

Main characteristics and attributes of these objects are:

- Unique identifier
- Two adjacent FCO which are connected by the CO
- Geometric representation (composed by a set of triangles)

Similarly to FCO, every CO may obtain information about their adjacent FCO.

CO are divided into Controllable CO and Non-Controllable CO. Controllable CO allows the changing of state while Non-Controllable CO do not. Every CO are watertight objects, i.e. do not allow fluid transfer between their adjacent FCO with the exception of the openings. Openings, that also connect two FCO, are generated by holes in the hull or in the bulkheads, or when a Controllable CO changes its state such as when a door or a valve is opened. This last situation generates an opening with the same geometry and location as the Controllable CO when it was closed.

2.1.3. Static Objects

Static objects exist only to provide more realism to the environment. These objects may be helpful to identify a specific zone or compartment of the ship. Although they are not considered on any flooding calculation, static objects increase the size of the model and so they must be chosen carefully. Only the relevant static objects should be modelled such as the engine in the Machine Room or the stove in the Galley.

2.2. Architecture

The architecture of the model defines the existent objects, their relationships and behaviour. It reflects on the efficiency of the final application and on the possibility of improving it with the minimum effort. Figure 4 shows in UML diagram the generic architecture used.

Two main aspects were considered when defining the architecture of the model:

- Simplicity, concerning only the essential types of objects, relations and behaviours
- Possibility of including new types of objects as sub-classes of the existing ones using inheritance

The world is composed by one instance of the class Ship. The ship contains an undefined number of objects of three main types: the Static Objects, the Connecting Objects and Fluid Container Objects.
Static Objects and Connecting Objects have an undefined number of geometric representations that provide them different Levels Of Detail (LOD).

Connecting Objects may be controllable or non-controllable depending on the possibility of changing its state. Doors, hatches, valves or pumps that have pre-defined states are Controllable Objects. A door may be open or closed; a pump may be turned-on or turned-off. Non-Controllable Objects do not change their state. Openings may be created or destroyed, but the never change state.

Doors and Hatches are always associated to an object of type Bulkhead. A door does not exist without a bulkhead or a hatch without a deck. By the same reason, Valves and Pumps are always associated to ducts or pipes.

Fluid Container Objects are bounded by at least two Connecting Objects. They may be compartments or ducts. A compartment may be crossed by an undefined number of ducts.

Fig. 4 – Model Architecture - UML diagram

2.3. Visualization

3D Visualization is one of the most important issues of Virtual Reality Applications. It is the feature that allows understanding easily and faster the model of the ship or the data produced by the simulation algorithm. An effort was made to provide a realistic 3D model of the vessel using textures,
lights and shadows. Although 3D Visualization provides an efficient and easy form to understand the model and its characteristics, it’s also useful to have the possibility of visualize the model in 2D layers corresponding to decks, namely when complex models are concerned. Actually, in the 3D space near objects hide the ones behind them. In ship models, normally upper decks hide lower decks and so a functionality that allows showing or hiding the decks was added. The consequence of this feature is that every object must have a flag pointing to its deck.

2.4. Functionalities

The defined architecture allowed a set of functionalities that might be useful in an emergency situation: Check compartments in risk, Compartment isolation and Path finding.

‘Check compartments in risk’ functionality informs the user, which compartments are in risk of flooding or contamination if the selected compartment is flooded or contaminated. The algorithm generates a list of compartments in risk using a recursive method. Since the compartments have a list of adjacent CO from which may obtain information, the algorithm checks which of those are openings. If they exist, the algorithm gets for each opening its adjacent FCO, which is in risk of flooding or contamination. The method is applied for each compartment in risk.

‘Compartment isolation’ functionality outputs a list of actions that must be performed to isolate a selected compartment. The algorithm checks for adjacent CO of type Opening to the selected compartment. If they exist, then for each of these objects, the algorithm check if there is a Controllable CO associated. If there is then the Controllable CO must be closed to isolate the compartment. Else, the object of type Opening has another adjacent compartment besides the selected one and the same verification must be performed to the holes of that compartment.

Path finding is a simplified version based on the algorithm described in Smith et al. (2002) and it returns a list of CO that define a path between two selected compartments. The path is defined considering flooded or contaminated compartments and human size accesses.

3. The algorithm for real time simulation of ship motion

The fluid dissemination algorithm, which allows the simulation of the ship behaviour and flooding status of the compartments, consists of two main modules: the equations of motion module and the flooding assessment module.

3.1. Equations of motion

The simulation algorithm is based on a theoretical model comprehensively described in Santos et al. (2002). The co-ordinate system shown in figure 5 and based at the ship centre of gravity, is used to compute the ship motions using a set of six non-linear coupled equations of motion with time dependent coefficients. These equations are:

\[
[M(t) + A(t)] \ddot{X}(t) + B(t) \dot{X}(t) + C(\phi, \theta, t) = F(t)
\]  

(1)

where \(M(t)\) is the mass matrix, \(A(t)\) is the added mass matrix, \(B(t)\) is the damping matrix, \(C(\phi, \theta, t)\) is the stiffness or restoring matrix, \(F(t)\) is the excitation force vector and \(X(t)\) is the motions vector. In these equations of motion with constant and variable coefficients, the non-linearities arise from the stiffness term, which is calculated directly and stored into the matrix.
The mass matrix $M(t)$, which is diagonal, represents the mass properties of the ship and depends on the ship’s mass $M$, and the principal moments of inertia, $I_{xx}$, $I_{yy}$ and $I_{zz}$. The moments of inertia are calculated using the formulae (3):

$$
I_{xx} = M k_{\theta}^2 \\
I_{yy} = M k_{\phi}^2 \\
I_{zz} = M k_{\psi}^2
$$

(3)

where the k factors represent the radius of gyration (roll, pitch and yaw, respectively). It is well known that the mass matrix is almost diagonal for most ship types, except for an off-diagonal term $I_{zx}$ and its symmetric, which are zero if there is fore and aft symmetry. This was taken to be the case, in this work, as a first approximation.

The motion-induced forces (added mass and damping terms, $A(t)$ and $B(t)$) are computed using a 2-D diffraction program. The sectional results are then integrated longitudinally to yield the total forces and moments. The viscous effects, especially important in relation to roll motion, are taken in consideration using an empirical method. This method provides a simple way to estimate the correct magnitude of the roll-damping coefficient.

The stiffness or restoring matrix, $C(\phi, \theta, t)$, accounts for the contributions of the heave restoring force, roll restoring moment and pitch restoring moment. Therefore, this matrix is only populated on the entries of the heave, roll and pitch modes. Furthermore, the stiffness matrix does not multiply a displacement vector because of its non-linear nature. Thus, each entry is calculated independently and then stored in its position.

The ship’s righting arms are obtained using the pressure integration technique, reported in Santos and Guedes Soares (2001), which integrates the hydrostatic pressure field over small panels defining the surfaces of the hull and internal subdivision. As some compartments will be flooded during the time domain simulation of flooding, the internal pressure of the floodwater has also to be integrated in order to calculate the properties of the vessel taking into consideration the floodwater effects. The floodwater is assumed to behave quasi-statically, i.e. it settles down to a regular volume with a horizontal free surface parallel to the mean sea level. This free surface can be ignored within the lost buoyancy method.

In this work, no excitation forces will be considered since the flooding and capsizing phenomena are being studied only in calm waters.

3.2. Flooding model

In the current work any water flowing from the sea to the damaged compartment(s) and between damaged compartments is modelled using well-established classical hydraulics formulae. Such floodwater is assumed to settle down instantaneously with a flat parallel surface to the waterline. This assumption is considered good if, within the flooded compartment, no significant obstructions to the
free flow of water exist. If such major obstructions do exist within a given flooded compartment, it is assumed possible to subdivide that compartment into several separate smaller compartments and cross-flood these compartments. The connections between compartments are modelled as polygonal panels. This modelling of crowded compartments allows the simulation of the asymmetric flooding within these compartments.

The flooding model consists of two separate components. The first component calculates the amounts of water that flows from the sea into the damaged compartment(s) and between compartments, during each time step. This is done using the pressure integration technique applied to the hydrostatic pressures that act across the damage openings and cross-flooding openings. There are, however, several cases to be considered in relation to the type of flooding, which are shown in figure 6. These cases arise because of the relative positions of the water level in each side of the opening.

The mean water velocity through any opening is obtained using the following hydraulic formula, based on the Bernoulli theorem:

$$U = k \sqrt{2.0(p_2 - p_1)} / \gamma$$

where $U$ is the mean water velocity, $k$ is the flow coefficient, $\gamma$ is the specific weight of the water and $p_2$ and $p_1$ are the pressures in the left and right sides of the panel. This formula only gives an approximation to the true flow velocity, which, when multiplied by a suitable cross-section area, yields the rate of flow. It is acknowledged that the true flow is extremely complex, both turbulent and chaotic. However, as a simplification of the true physical phenomenon, it has been assumed that the flow can be adequately modelled by this formula with carefully chosen values of $k$.

The second component of the flooding model consists of calculating the effects of floodwater on ship stability. This is done using the pressure integration technique applied to the internal wetted surfaces, yielding the geometric centre of the floodwater. This floodwater is then taken as an amount of lost buoyancy within the concerned compartment. This procedure is repeated for all flooded ship compartments and, after the summation of the values of intact buoyancy for all compartments, the total amount of buoyancy is found. The centre of buoyancy can be found in an analogous way. Knowing the centre of buoyancy and, given the centre of gravity, the righting arms can be easily found and used to obtain the stiffness matrix.

4. **Connection between the decision support tool and the time simulation algorithm**

The connection between both tools required data and run-cycle compatibility. The models used by the tools were different since they had different objectives. Some data had to be rearranged to provide input for the simulation algorithm. For identification proposes, the model used by the Virtual Reality tool was called the *Virtual Model* while the model used by the simulation tool was called the *Simulation Model*. Diagram of Fig. 7 presents the main cycle for the Decision Support Tool during the flooding simulation.
As shown, the model of the ship is loaded into the Virtual Environment. If no event occurs, the state of the Virtual Model remains unchanged. An event occurrence starts the flooding cycle and the model is consecutively updated until the final state is achieved. The event that starts the flooding cycle must always include the creation or destruction of an opening between two FCO.

Once the cycle is started the Virtual Model is updated (an opening is added or deleted) and the data corresponding to the updated model is exported. This data is loaded and updated (or not) by the Simulation Tool. The new data is then exported to the Virtual Environment Tool that checks if it was changed or not. If the new input data is different of the previous cycle output data, then a new cycle starts with the new input data. Else, if the new input data is equal to the previous output data then one of two situations has occurred: the event did not have influence on an eventual flooding and the cycle does not start or the model as achieved its final state and the cycle terminates.

Every time a flooding cycle initiates, the simulation algorithm loads the Simulation Model, corresponding to the initial state of the model and applies the transformations according to the output data. This Simulation Model is generated in the Application Initialisation phase.

4.1. Data compatibility

The input for the Simulation Algorithm that is generated by the Virtual Reality Tool consists in the geometry of the vessel for simulation purposes, i.e. the Simulation Model, and the Hole Geometry (existent holes).

The Simulation Model is automatically generated during the initialization of the Virtual Reality Application. This model is only generated once and remains unchanged during the application runtime. It consists of a structure containing the following information:
- Main dimensions of the vessel
- The compartments of the vessel and their corresponding geometries

The Virtual Model provides all the necessary information to generate the Simulation Model. Main dimensions as well as compartments’ identifications, locations, polygons and corresponding 3D points in space were extracted from the geometric and topological database of the Virtual Model. Since the compartments for the Simulation Model are defined as closed polyhedrons, it was not necessary any data arrangement.
A new data structure had to be created and updated in run time to provide the existent holes, their geometry and adjacent compartments to the Simulation Algorithm. This structure is mentioned in Fig. 7 as Hole Geometry. The Hole Geometry is also generated during the initialization of the Virtual Reality Tool and updated when an object of type Opening is created or destroyed. The structure contains the list of holes, each one including the name, the adjacent FCO and a list of 3D points to define the polygon that represents the hole. While for the Virtual Environment, the order of these 3D points was not relevant, for the simulation algorithm they had to be clock wised or anti-clock wised. Hence, the holes’ points collected from the Virtual Model had to be rearranged using the algorithm described in Popovice (1998). The result is shown in Fig. 8 for a hole generated by a door.

![Fig. 8 – Watertight Door with corresponding Hole](image)

The exchange of remaining data between tools such as, draft, trim, heel or water high in compartments was unproblematic since they had exactly the same meaning in both models.

4.2. Run-cycle compatibility

The simulation algorithm was created to run during a certain period of time, calculating the stability data for consecutively discrete points in time depending on a defined time-step. The result would be a table containing stability data for the consecutive times. The input for the simulation algorithm was initially assumed static in time, i.e., no new holes were generated or destroyed during the simulation. The Virtual Environment Tool may use this original output of the simulation tool for post-processing purposes. However, for simulation in real-time and flooding control by the user, two modifications had to be done:

- The results produced by the Simulation Algorithm had to be constantly updated in the Virtual Environment
- The input data for the Simulation Algorithm could also change during the simulation

Hence, instead of running the simulation algorithm for a previously defined number of cycles with a specific time-step, the algorithm had to be called once on every Virtual Environment cycle during the simulation. The time step was the duration of the previous cycle before the next Simulation Algorithm call.

This provided the possibility of updating the Virtual Model according to the input data every cycle and update this input data (generating the output data) for the Simulation Algorithm when relevant changes occurred in the Virtual Model. It also allowed the possibility of changing the Hole Geometry in real-time.

5. Results

Assuming that the simulation algorithm has already been tested and its results are correct, evaluation of the Virtual Reality Application concerned the following:
• Correct data exchange between *Virtual Environment* and *Simulation Algorithm*
• Frame rate
• Visualization of the model

Data exchange was preformed correctly between tools even when the *Hole Geometry* changed during the simulation.

Using a virtual model with 25 compartments, 184 objects and 3268 polygons (triangles), a frame rate of 15 fps was achieved with a non-optimised application using an Intel Pentium 4, 1.6Gb with 512 Mb RAM. The frame rate is not brilliant namely because the number of polygons is not very high. Fig.9 presents some screen captures obtained during the simulation in different time and for different *Hole Geometry*.

On the first capture there are no holes and so the compartments remain waterless. On the second capture, a door (used only to test the model) to the exterior is opened and the adjacent compartment is flooded. Next captures show that other doors are opened and compartments are flooded. The trim, draft and hell of the vessel are also changed due to the flooding process. Each of the captures shows final states of the model for specific *Hole Geometry* that was changed in real-time.

![Fig. 9 – Screen captures during the simulation](image)

6. **Conclusions**

The architecture of the Virtual Environment was able to couple a simulation algorithm performing data exchange in real time, and showing the results from each cycle in a 3D environment also in real time. The Virtual Environment can also do post-processing of the simulation algorithm results. The data compatibility did not require much effort since the tools had a similar conception to the model.

The problem of the frame rate is mainly due to the complex calculation of the Simulation Algorithm. Solution may be the use of other simulation algorithm (not likely) or optimisation of the process. This
optimisation may include code optimisation or decrease the Simulation Algorithm calls increasing as much as possible the time step. The relation Virtual Environment Cycles/Simulation algorithm Cycles should be the highest possible. Note that increasing the time step may originate inaccuracies in the calculation. To avoid visual discontinuities between two Simulation Algorithm calls, linear interpolation can be done when updating the Virtual Model.

The prototype of the application as proved to be an efficient tool to visualize in real time the model of the ship including its compartments’ disposal and equipment. It’s also a good tool to visualize the simulation results in real time providing understandable information of what is happening during the flooding process.

References


From Redesign to Optimal Hull Lines by means of Parametric Modeling

Claus Abt, FRIENDSHIP-Systems, Berlin, Germany, Abt@friendship-systems.com
Stefan Harries, Technical University Berlin, Germany, Stefan.Harries@ism.tu-berlin.de
Justus Heimann, Technical University Berlin, Germany, heimann@ism.tu-berlin.de
Henning Winter, Technical University Berlin, Germany, winter@kandinok.de

Abstract

The efficient development of optimal solutions to design problems of high complexity has become imperative to increase competitiveness and success in shipbuilding. Information technology integration is a key factor in speeding-up the design process and in improving the product evolving from it. This holds in particular when one-of-a-kind systems need to be engineered and delivered in a short time.

The paper intends to contribute to the field of ship design by presenting a parametric approach to the modeling and hydrodynamic optimization of ship hull forms. Focus is given to the redesign and parameterization of complex shapes featuring hulls with bulbous bows. A sophisticated modeling technique – integrated in the novel system FRIENDSHIP-Modeler – is discussed with regard to efficient and effective form generation and variation.

Based on a RoPax ferry typical of fast short-sea shipping, all stages from redesign to hydrodynamic optimization will be described. The paper therefore presents an overview on the practicalities of shape import, identifying suitable parameters for shape variation and applying parametric principles to optimization. For the task of optimization and IT integration the state-of-the-art system called modeFRONTIER was utilized while the numerical flow simulation was performed with the well-established SHIPFLOW code. Additional wave pattern analysis via SWASH – the new wave analysis tool developed at the Technical University Berlin – was undertaken to strengthen the applicability of the flow simulation in the context of formal optimization.

All tools combined establish an advanced hydrodynamic design process implemented as a multi-layer optimization problem of non-linear programming. The elaborate example given will serve to illustrate objectives, methods and results.

1. Introduction

Can we still squeeze out a few percent? Shall I increase the bulb’s volume and is it advantageous to shift the forward shoulder? Would we not like to know more about the design space we find ourselves in?

In naval architecture we often face the predicament of coming up with a good hull shape within the tight corset of many constraints from many different fields – safety and comfort, cargo capacity and handling to mention just a few. Once a feasible solution is identified the design team often lacks the resources in time and budget to undertake an extensive search for further improvement. It would therefore be quite nice to let the computer do the busy-work, gain the freedom to lean back and wait and, in the end, simply decide which favorable result we are inclined to accept. Unfortunately, such a tool is not (yet) available. But even a long journey commences with the first step and a promising course adopted to accomplish the task of investigating many designs in a reasonably short time is automated optimization by means of parametric modeling as discussed for instance by Birk and Harries (2000).

Simply put, automated optimization is the formal process of finding a good (the best) solution from a set of feasible alternatives. It requires a complete mathematical problem formulation in terms of objective functions (what is to be improved), free variables (what shall be consciously changed) and constraints (what restricts the feasibility). Within this paper we will follow this line of thought and focus on hydrodynamic design. We assume the somewhat idealized vista point of optimizing an initial hull by minimizing its wave resistance component in calm water. The example ship chosen is that of a
fast ferry called FantaRoRo that was devised as an elaborate test case within the European R&D project FANTASTIC, see Maisonuneve et al. (2003) for an overview on FANTASTIC. The FantaRoRo test case was set up by the consortium to study the potential of the tools developed or improved throughout the project. Several alternative optimization schemes were applied by various partners. The purpose of this paper is to present the details of the approach developed at the Technical University Berlin.

A comprehensive optimization task from redesign to optimal hull lines will be discussed. Figure 1 shows the lines plan of the example ferry. The main particulars of the design are summarized in table I. Key constraints for the optimization are given in table II. As can be seen from table II we allowed ourselves a tangible freedom in shape modification so as to better see merits (and shortcomings).

![Fig. 1: FantaRoRo initial shape](image)

**Table I: Main particulars of FantaRoRo**

| L<sub>pp</sub> | 122.74 m | Length between perpendiculars |
| B   | 19.2 m   | Maximum breadth molded       |
| T   | 5 m      | Design draft                 |
| V<sub>l</sub> | 6584 m<sup>3</sup> | Displacement volume          |
| C<sub>B</sub> | 0.549 | Block coefficient            |
| x<sub>CB</sub> | 57.8 m | Longitudinal center of buoyancy from AP |
| A<sub>WP</sub> | 1836 m<sup>2</sup> | Waterplane area              |
| V<sub>S</sub> | 21 kn   | Design speed                 |
| F<sub>n</sub> | 0.311   | Froude number                |

**Table II: Key optimization constraints**

<table>
<thead>
<tr>
<th></th>
<th>No changes aft of the maximum section</th>
</tr>
</thead>
<tbody>
<tr>
<td>L&lt;sub&gt;max&lt;/sub&gt;</td>
<td>128 m</td>
</tr>
<tr>
<td>ΔV&lt;sub&gt;l&lt;/sub&gt;</td>
<td>±10 % V&lt;sub&gt;l&lt;/sub&gt;</td>
</tr>
<tr>
<td>Δx&lt;sub&gt;CB&lt;/sub&gt;</td>
<td>0.6 m</td>
</tr>
<tr>
<td>Δx&lt;sub&gt;Ch&lt;/sub&gt;</td>
<td>0.7 m</td>
</tr>
</tbody>
</table>
2. Design Scenarios in Geometric Modeling

Independent of the modeling approach, two design scenarios can be generally distinguished:

-Redesign of an existing shape and
-Design from scratch.

In shipbuilding the redesign of an existing shape and its subsequent modification to meet new criteria is widespread. A shipyard normally utilizes its data base of previous projects to set up an initial hull which then serves as the starting point for another vessel. A model basin also faces the task of remodeling a given shape regularly. It typically receives a hull form for tests and further perfection in either a proprietary or a common exchange format. Unless the model basin happens to run the same Computer Aided Design (CAD) software, however, a geometry import and certain post-processing is needed.

Design from scratch is undertaken more often in the field of yacht design in which the entire hull shape is established virtually from an empty piece of paper (or screen). Due to the simpler geometry of a yacht’s bare hull and appreciably higher influence of aesthetics, many designers prefer this line of work. Nevertheless, when undertaking an optimization even in yacht design one may encounter the remodeling of a parent hull, see e.g. Hochkirch et al. (2002).

In the following we will concentrate on redesign rather than on design from scratch. Once the initial shape is established the successive steps of hydrodynamic design are equal in any case, see figure 4.

2.1. Parametric Modeling

In parametric modeling the design problem is formulated partly or even fully in terms of parameters. Parameters are the descriptors of the product to be developed. In geometric modeling the parameters are more precisely called form parameters. Instead of simply replacing numbers by variables and then again assigning values to them, parameters feature relationships. The value of a parameter might thus be computed from a formula including higher ranked parameters, it might also depend on certain conditions or it might be computed from a set of equations etc. In this way a product is represented at a much higher level then by point data alone as it is still done very often.

For example: A cube is completely defined by the three Cartesian components of all eight corner points. Conventionally you would enter the point data to each corner, i.e., you would specify 24 real numbers in total. In a parametric model you could reduce the definition to one single parameter if you wanted, say a characteristic length. The breadth and height of the cube could be computed as a fixed percentage of the characteristic length. If you needed more freedom you could use three parameters, namely the cube’s length, breadth and height independently. As can be readily deduced, a parametric description comprises all instances of a product and a specific product can be generated with less effort. However, to some extent the parametric model also imposes a limitation due to its build-in relationships. For further discussions see Abt et al. (2001).

Today’s CAD software can be generally subdivided into three categories:

- Conventional tools without parametric support,
- Tools with certain parametric capability and
- Fully-parametric design tools.

In the context of optimization a conventional non-parametric approach has serious disadvantages. While the number of free variables – e.g. point data – that govern the problem rapidly becomes prohibitively high, the influence of each individual variable also decreases. Moreover, the fine tuning of variables to achieve a desirable result gets very demanding.

Consequently, a fully-parametric design tool has been developed for the hydrodynamic design of ships and yachts which is now available on the market: The FRIENDSHIP-Modeler. The
FRIENDSHIP-Modeler originates in research performed at the Technical University Berlin, see Harries (1998). A substantially extended and enhanced version is available from FRIENDSHIP-Systems, see website in reference list.

2.2. FRIENDSHIP-Modeler

The FRIENDSHIP-Modeler’s key feature is a fully-parametric shape description. Each hull – be it a sailing yacht with keel and rudder or a ship with bulbous bow – is generated in a three stage process. The entire model is based on B-splines curves and surfaces which are optimized for fairness.

In the first step a flexible set of longitudinal curves (basic curves) is produced from form parameter input. An excerpt of the human readable form parameter file for the example FantaRoRo can be seen in figure 2. In the second step a skeleton of transverse curves is created according to the parametric information contained in the basic curves. In the third step a set of fair surfaces is built that interpolate this skeleton of transverse curves. Figure 2 also shows the graphical user interface (GUI) of the FRIENDSHIP-Modeler, featuring an improved FantaRoRo (TUB 882*modi) in perspective view.

![GUI of FRIENDSHIP-Modeler with excerpt of form parameter file and perspective view of TUB 882*modi](image)

2.3. Parametric Redesign of Existing Shapes

The FRIENDSHIP-Modeler supports both the redesign of existing shapes and the design from scratch. The original hull shape for the FantaRoRo test case was developed within NAPA, see website in reference list. Figure 3 presents a comparison between the original geometry and the redesigned geometry. Considering NAPA’s different modeling concept, fair agreement can be observed, in particular in the underwater portion of the hull. No considerable attempt was made to further reduce
the deviations in the forebody since the entire hull was free to change from the maximum section forward in the optimization.

Hull shapes can currently be imported into the FRIENDSHIP-Modeler either via IGES files or via offset data. In the former case the user fills up a parameter file interactively and iteratively compares the hull form produced with the geometry imported until satisfaction. In the latter case – which was pursued here – the offset data is analyzed with regard to key form parameters. A template is selected which suits the topology of the ship at hand (e.g. ship features a bulbous bow). The determined form parameter values are then inserted in the form parameter file in accordance with the chosen template. Afterward, additional form parameters might be added as desired.

3. Hydrodynamic Optimization

Figure 4 depicts the typical process flow of a formal hydrodynamic optimization. A very tight integration of four modules is essential for the optimization’s success:

- Shape generation,
- Hydrodynamic analysis,
- Performance assessment and
- Optimization strategies.

Throughout this study the shape generation was done via the FRIENDSHIP-Modeler. The hydrodynamic analysis was carried out with the potential flow module of the well-known SHIPFLOW code, see website in reference list. (Complementing computations were undertaken with SHIPFLOW’s boundary layer module.) The prime objective function of the optimization having been the wave resistance, a sophisticated wave cut analysis was utilized for performance assessment in addition to pressure integration over the panels. SWASH, a tool developed at the Technical University Berlin, was applied for longitudinal wave cut analyses, see Heimann (2000). It yields the wave pattern resistance $R_{WP}$ while – by convention – a pressure integration provides the wave resistance $R_{W}$. SWASH treats wave cuts of finite length with a special truncation correction and allows a detailed examination of gains and losses via the spectral distribution of a wave energy equivalent along the components of the steady ship wave system.
Finally, for integration of the various tools and as a workbench providing many different optimization strategies the multi-objective design environment modeFRONTIER was utilized, see website in reference list.

From figure 4 it can be seen nicely that hydrodynamic optimization is an iterative and interactive design process that should not be confused with a black box that yields the best ship for simple questions asked. Even though many design variants might be automatically assessed at each time an optimization run is started, the entire design procedure requires the users to evaluate and reconsider their problem set-up as the investigation progresses.

The process commences with a pre-processing phase. Once a parametric model is established – a form parameter file is produced for the FRIENDSHIP-Modeler – the geometry can be generated and modified very elegantly. A detailed analysis of the initial design will then be undertaken which implies panel variation studies, convergence tests and accuracy checks for instance.
Then follows the actual optimization phase which can be further subdivided into three steps. In the first step the optimization task is formulated. The free variables – i.e., the form parameters that shall be varied – are chosen along with their appropriate bounds. One or several objective functions are identified and the constraints are incorporated. In the next step the design space is explored so as to gain a first insight. One might chose a design-of-experiment (DoE), for example a random distribution of variants. For each set of free variables three key modules are executed one after another: shape generation in accordance with the current set of form parameters, flow analysis on the basis of the present hull geometry and performance assessment based on the flow field just computed. A set of designs is thus produced which (hopefully) contains some improved hull forms. This step is carried out several times, possibly applying different optimization strategies. It might also happen that the optimization set-up needs to be adapted to better suit the problem at hand. For example, the bounds of selected free variables are shifted, increased or decreased. (Depending on the gains in the objective functions achieved and the general behavior of the shape modification, even a further step backward can be advisable in order to amend the parametric model.) Following the exploration step, a third step is taken. Utilizing the set of designs investigated so far, a further exploitation of promising variants is attempted. This is done by either a deterministic or a stochastic optimization strategy. An example of the former is the SIMPLEX algorithm according to Nelder and Mead, see Press et al. (1994), an example for the latter being the MOGA (multi-objective genetic algorithm), see Spicer, Poloni et al. (1998). The exploitation is usually performed several times until satisfying solutions have been found. (One interesting alternative to going through the time consuming and resource intensive flow analysis and performance assessment is to bypath the full simulation by a response model.) At the end of the exploitation step several improved designs are available from which the best is to be identified.

In the case of a multi-objective optimization the selection of the final design is a non-trivial task in itself. It is undertaken in the third phase of post-processing. The process may then end with a further analysis of the final design.

4. Towards Optimal Hull Lines – Elaborate Example

4.1. Optimization Process

The optimization set-up as realized with modeFRONTIER is shown in figure 5. A set of 15 form parameters was used here for modifying the hull geometry. The free variables correspond to two basic curves – the design waterline DWL and the sectional area curve SAC (for the forebody) – and the bulbous bow BBOW. In table III the form parameters are listed along with the (final) bounds and the values they assume for several designs that were considered favorable with respect to various objectives. The values of the initial hull form are given, too.

From figure 5 you may also note that inequality constraints were imposed on displacement and longitudinal center of buoyancy, see table II. (Actually, the displacement of the bare hull was implicitly kept constant within the FRIENDSHIP-Modeler during the optimization but slight changes in the overall displacement occurred due to variations in the bulb volume.) Several important quantities like trim and sinkage, waterplane area, wetted surface area etc. were monitored.

As already stated above, the wave pattern resistance \( R_{wp} \) and the wave resistance from pressure integration \( R_W \) were employed as the major objective functions. The total resistance \( R_T \) was also taken into consideration. Following Froude’s hypothesis, the viscous component of the total resistance was simply added to the wave resistance component. The frictional resistance was computed from the ITTC 57 line, a reasonable form factor was estimated and assumed constant since no changes were permitted in the afterbody.
Fig. 5: Optimization set-up in modeFrontier

Tab. III: Form parameters in example optimization

<table>
<thead>
<tr>
<th>Free variable / form parameter</th>
<th>Lower bound</th>
<th>Upper bound</th>
<th>TUB parent (initial hull)</th>
<th>TUB 876</th>
<th>TUB 848*</th>
<th>TUB 879*</th>
<th>TUB 882*</th>
<th>TUB 882*modi</th>
</tr>
</thead>
<tbody>
<tr>
<td>SACcaForShoulder</td>
<td>75.0 %</td>
<td>85.0 %</td>
<td>77.9059 %</td>
<td>79.25 %</td>
<td>76.75 %</td>
<td>78.20 %</td>
<td>77.65 %</td>
<td>77.65 %</td>
</tr>
<tr>
<td>SACcaForFrame</td>
<td>10.0 %</td>
<td>18.0 %</td>
<td>13.1966 %</td>
<td>15 %</td>
<td>15.85 %</td>
<td>16.25 %</td>
<td>15.45 %</td>
<td>15.45 %</td>
</tr>
<tr>
<td>SACtanAtFrame</td>
<td>20.0</td>
<td>60.0</td>
<td>35</td>
<td>37.5</td>
<td>26.0</td>
<td>28.75</td>
<td>27.75</td>
<td>27.75</td>
</tr>
<tr>
<td>DWLtanAtBow</td>
<td>5.0</td>
<td>12.0</td>
<td>9.5</td>
<td>10.5</td>
<td>11.2</td>
<td>11.3</td>
<td>11.75</td>
<td>11.75</td>
</tr>
<tr>
<td>DLWAreaCoeff</td>
<td>0.57</td>
<td>0.65</td>
<td>0.625</td>
<td>0.57375</td>
<td>0.57</td>
<td>0.6125</td>
<td>0.587</td>
<td>0.587</td>
</tr>
<tr>
<td>TADWLatFrame</td>
<td>60.0</td>
<td>100.0</td>
<td>63.1275</td>
<td>63.1275</td>
<td>61.0</td>
<td>60.0</td>
<td>60.25</td>
<td>60.25</td>
</tr>
<tr>
<td>BBOWareaCoeffFor</td>
<td>0.825</td>
<td>0.925</td>
<td>0.85</td>
<td>0.9125</td>
<td>0.925</td>
<td>0.925</td>
<td>0.925</td>
<td>0.925</td>
</tr>
<tr>
<td>BBOWareaCoeffAft</td>
<td>0.35</td>
<td>0.45</td>
<td>0.4</td>
<td>0.4175</td>
<td>0.45</td>
<td>0.414</td>
<td>0.45</td>
<td>0.4</td>
</tr>
<tr>
<td>BBOWAreaAtFP</td>
<td>7.0</td>
<td>10.0</td>
<td>8.57</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>BBOWdAreaAtFP</td>
<td>30.0</td>
<td>50.0</td>
<td>35</td>
<td>42.5</td>
<td>43.5</td>
<td>49.6</td>
<td>46.1</td>
<td>40.0</td>
</tr>
<tr>
<td>BBOWlength</td>
<td>4.0</td>
<td>5.25</td>
<td>4.24</td>
<td>5.175</td>
<td>5.25</td>
<td>5.25</td>
<td>5.25</td>
<td>5.25</td>
</tr>
<tr>
<td>BBOWHeightAtFP</td>
<td>87.5 %</td>
<td>95.0 %</td>
<td>90.0 %</td>
<td>90.0 %</td>
<td>87.5 %</td>
<td>87.5 %</td>
<td>87.5 %</td>
<td>87.5 %</td>
</tr>
<tr>
<td>BBOWzTop</td>
<td>-0.2</td>
<td>0.3</td>
<td>-0.08</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2975</td>
<td>0.3</td>
</tr>
<tr>
<td>BBOWxTop</td>
<td>60.0 %</td>
<td>75.0 %</td>
<td>68.0 %</td>
<td>68.0 %</td>
<td>75.0 %</td>
<td>72.6 %</td>
<td>73.3 %</td>
<td>73.3 %</td>
</tr>
<tr>
<td>BBOWzTip</td>
<td>-1.5</td>
<td>-0.5</td>
<td>-1.15</td>
<td>-1.15</td>
<td>-0.735</td>
<td>-0.66</td>
<td>-0.685</td>
<td>-0.685</td>
</tr>
</tbody>
</table>

Note: TUB 876 stems from a different optimization run as the variants marked with an asterisk * TUB 882*modi resembles TUB 882* apart from a slight modification introduced a posteriori right aft of FP
Fig. 6: Optimization history

Fig. 7: Scatter diagram with $R_{WP}$ and $R_F$ as ordinate and abscissa, respectively
For the optimization runs a medium size panelization with 1185 panels covering one half of the hull surface and 5125 panels for the free surface per symmetric half was employed. The free surface panelization was extended x/L_{pp} = 2.4 downstream of the stern in order to allow for the longitudinal wave cut analysis. The non-linear flow analysis was restarted from previously converged non-linear solutions so as to reduce computational effort. Trim and sinkage were free.

An exemplary optimization history is provided in figure 6, the ordinate being the ratio of R_T to R_{T_{initial}} and the abscissa being the design id. A design-of-experiment with 150 candidates is followed by a MOGA optimization. Figure 7 depicts a scatter diagram of R_{WP} vs. R_T. The designs are marked by
square boxes while a Pareto front – the set of all solutions for which a single objective cannot be further improved without deteriorating any other objective – is drawn as a dotted curve. All hull forms lie above and to the right of the Pareto front. Those designs that are closest display low wave pattern resistance along with low total resistance. Apparently, it is impossible to decrease $R_{WP}$ below certain limits without impairing $R_T$. For those variants that make up the Pareto front the design id is given. In tables III and IV data for good designs are summarized.

Throughout the process several optimization runs were executed. Instead of discussing them all only the final run shall be explained in detail: From the data basis built up in previous optimizations a MOGA was started with the 80 best candidates for the initial population. The wave pattern resistance and the total resistance were utilized as the two competing objectives. A total of 30 generations was generated and traced. Finally, in order to further benefit from the results achieved already a SIMPLEX search was activated from the very best designs. For the SIMPLEX just one objective was accounted for, namely the sum of $R_{WP}$ and $R_T$. The SIMPLEX was finally terminated at a stage were improvements became marginal.

4.2. Further Analysis

For a more thorough comparison of the most promising designs further flow analysis and assessment was undertaken with a finer panel mesh of 1767 panels on the hull surface (247 of them for the bulb upstream of FP) and 7976 panels on the free surface per symmetric half. Along the hull more than 40 free surface panels per fundamental wavelength ($\lambda_0/L_{pp} = 0.609$) were utilized in longitudinal direction. The width of the free surface panels next to the hull was 0.02 $L_{pp}$. A slight panel stretching was applied sidewise and downstream of the stern.

The results are presented in table IV. When compared to the initial hull form one may observe that TUB 848 star reduces the wave generation the most while TUB 879 star promises the highest improvement in pressure distribution. TUB 882 star along with its slightly modified version TUB 882*modi might be characterized as the trade-off bringing together the advantages of both former designs. TUB 876 (see figure 8 for a lines plan) is an intermediate result from one of the first optimization runs. It features pronounced (S-shaped) buttocks resulting from a narrow design waterline. It therefore resembles an interesting deviation from conventional hull shapes.

<table>
<thead>
<tr>
<th>Results</th>
<th>TUB Parent (initial hull)</th>
<th>TUB 876</th>
<th>TUB 848*</th>
<th>TUB 879*</th>
<th>TUB 882*</th>
<th>TUB 882*modi</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_T$[TITC57] [N]</td>
<td>403601</td>
<td>386148</td>
<td>380127</td>
<td>376900</td>
<td>378362</td>
<td>378541</td>
</tr>
<tr>
<td>Gain [%]</td>
<td>–</td>
<td>4.32</td>
<td>5.82</td>
<td>6.62</td>
<td>6.25</td>
<td>6.21</td>
</tr>
<tr>
<td>$R_W$ [N]</td>
<td>165726</td>
<td>146184</td>
<td>139786</td>
<td>137528</td>
<td>138568</td>
<td>138944</td>
</tr>
<tr>
<td>Gain [%]</td>
<td>–</td>
<td>11.79</td>
<td>15.65</td>
<td>17.01</td>
<td>16.39</td>
<td>16.16</td>
</tr>
<tr>
<td>$R_{WP}$ [N]</td>
<td>136031</td>
<td>118114</td>
<td>113869</td>
<td>117098</td>
<td>115238</td>
<td>115757</td>
</tr>
<tr>
<td>$R_{WP}$ due to transverse waves [N]</td>
<td>98817</td>
<td>78374</td>
<td>79354</td>
<td>82741</td>
<td>80887</td>
<td>81459</td>
</tr>
<tr>
<td>Gain [%]</td>
<td>–</td>
<td>20.69</td>
<td>19.70</td>
<td>16.27</td>
<td>18.14</td>
<td>17.57</td>
</tr>
<tr>
<td>$R_{WP}$ due to diverging waves [N]</td>
<td>37213</td>
<td>39741</td>
<td>34515</td>
<td>34357</td>
<td>34351</td>
<td>34298</td>
</tr>
<tr>
<td>Gain [%]</td>
<td>–</td>
<td>-6.79</td>
<td>7.25</td>
<td>7.67</td>
<td>7.69</td>
<td>7.83</td>
</tr>
<tr>
<td>Wetted surface area [m²]</td>
<td>2479</td>
<td>2501</td>
<td>2505</td>
<td>2495</td>
<td>2499</td>
<td>2497</td>
</tr>
</tbody>
</table>

Tab. IV: Comparison of promising variants.
Fig. 10: Wave contours of initial hull form, TUB 876 and TUB 882 modi

Fig. 11: Longitudinal Wave cuts at $y/L_{pp} = 0.25$
Fig. 12: Distribution of the wave pattern resistance coefficient as a function of wave direction

The initial hull, TUB 876 and TUB 882*modi (see figure 9 for the lines plan) shall be looked at a little closer: The reduction in resistance accomplished by TUB 876 and TUB 882*modi when compared to the initial hull originates in more favorable wave interferences. Figure 10 displays contour plots of wave height for the three hulls. It is evident that the wave system has been substantially influenced. Longitudinal wave cuts are depicted in figure 11 for which a wave pattern analysis is presented in figure 12. Figure 12 reveals that the major portion of wave resistance improvement is attained by a reduction of the transverse wave components comprising wave directions up to appr. 35°. With the exception of TUB 876 further though smaller gains stem from the diverging waves as associated with the wave directions above 35°, compare to table IV. TUB 876 is best with regard to transverse waves but does perform less beneficial for diverging waves while TUB 882*modi represents an excellent trade-off capable of significantly minimizing wave energy losses.

Figures 13 and 14 feature the pressure distribution and limiting streamlines of the initial hull, TUB 876 and TUB 882*modi. The forebodies and bulbous bows are distinctively different and their effect on pressure distribution and flow direction is pronounced. From figure 14 one may also get an appreciation of the wave profiles along the hulls.

5. Conclusion

A comprehensive optimization of a fast ferry was presented on the basis of a synthesis model which comprised advanced parametric modeling, state-of-the-art flow analysis and detailed performance assessment. The optimization process features a multi-phase, multi-step, interactive and iterative character whose roots lie in the complexity of hydrodynamic design. The process relies on a complete IT integration as the prerequisite for automated optimization. The automated optimization itself enables the designer to investigate many variants without the overhead of tedious and non-creative work. In a multi-objective problem the best result depends on the designer’s preferences (introducing also some subjectivity).

The process is initialized by redesigning a hull form in a fully parametric design tool. The selection of suitable free variables and the determination of their bounds is an important issue and needs adjustment during the optimization. For the exploration and the exploitation of the design space deterministic and stochastic strategies are applicable, each having their advantages and disadvantages. In order to get a first insight into the problem, one may start with a random distribution. A particularly good candidate can be improved quickly with a deterministic search. A more time-consuming but very comprehensive search is provided via genetic algorithms. One should not limit oneself to just one strategy but should try to combine all available tools to the best advantage.
Acknowledgement

Parts of the research reported in this paper were performed within the European R&D project FANTASTIC – Functional Design and Optimisation of Ship Hull Forms (GRD1-1999-10666 under the 5. FP). The consortium comprised the following partners: CETENA, Chalmers, CIMNE, FINCANTIERI, FLOWTECH, FSG, HSVA, IZAR, MARIN, NAPA, SINTEF, SIREHNA, SSPA, TU Berlin.

References


http://www.esteco.it/
http://www.flowtech.se/
http://www.friendship-systems.com/
http://www.napa.fi/


Fig. 13: Pressure distribution on initial hull form, TUB 876 and TUB 882 modi

Fig. 14: Limiting streamlines for initial hull form, TUB 876 and TUB 882 modi
Multi-Agent Systems In Ship Design

Bekir S. Türkmen, Univ. of Glasgow and Strathclyde, Glasgow/UK, bekir.turkmen@na-me.ac.uk
Osman Turan, Univ. of Glasgow and Strathclyde, Glasgow, UK, o.turan@na-me.ac.uk

Abstract

The complex nature of ship design results in using a variety of simulation tools in order to reach optimum and cost effective design. Every simulation tool for an engineering discipline requires the user to have a certain degree of expertise in that discipline. The collaboration among engineering disciplines is a very important factor in overcoming design errors and to reach optimum designs. The multi-agent systems are very promising systems for collaboration, communication and cooperation.

An intelligent agent has the capability to perceive its environment by sensors to reason, plan and learn by implicit representation of intelligence, and to act by effectors in the environment. We can think of intelligent agents as building blocks of artificial intelligence. They can perceive, infer, learn and act. They can be software agents or physical agents (robots, humans); they can migrate through networks (mobile agents) and use the resources available in a network environment to accomplish their designated tasks. One of the important features of intelligent agents in engineering design is that they can encapsulate the design knowledge and additionally they can solve communication problems in a multidisciplinary design environment.

The multi-agent systems have been used for a wide range of applications, such as robotics and manufacturing. The implementations of multi-agent systems in concurrent design have been experimented with for over past decade and they are getting to such a mature level that they can now be applied by the industry.

This paper will briefly describe agent technology and multi-agent systems, and it will present the intelligent agent architecture as well as the multi-agent system architecture of the proposed system. The intelligent architecture for the proposed system is chosen as a layered approach and the proposed multi-agent system is defined in an organisational structure. Decision maker and optimisation agents provide decision support to the designers to solve conflicting and cumbersome tasks. Finally, the proposed system is presented by using an illustrative multi-objective ship design optimisation problem, which is followed by conclusion and discussions.

1. Introduction

The design of ships is always a challenge for ship designers, because it involves many engineering disciplines and a lot of complexities. There have been many attempts to formalise the design process of the ships to have a consistent and re-implemented framework. The first attempts in ship design to create a design framework could be seen as a design spiral methodology. In this methodology, the design starts from a single point and iteratively and interactively tries to evolve the design into the optimum design. The iterative cycle results in many man-hours being spent due to a change in design at especially production stage. We cannot avoid the iterative cycle of the practical ship design, but we can put the producibility and supplementary needs into the cycle before any production work starts, perhaps at the preliminary design stage. The second approach in ship design can be thought of concurrent engineering for solving the problem of design changes later in the production stage. Formally, concurrent engineering is a concept of distributing and constructing the design process in a way that all the important aspects of the design such as production and user requirements have been taken into account from conceptual stage to the disposal of the product.

Teams and tools are very important concepts of the concurrent engineering, and communication among the teams is the key factor for the effective concurrent engineering design practise. However,
overwhelming communication needs and meetings will create bottlenecks in the concurrent engineering design, Klein (1994). There must be a good communication environment amongst the teams to enable concurrent engineering to practise efficiently. On the other hand, during the last decade simulation became a very important factor for the navy defence sector. Evaluation of designs in respect to different scenarios in a war game and assessing the performance of the designs in game scenarios is the main purpose of the simulation. The simulation-based design has resulted from the needs in the defence sector and it is based on the simulation of the design from different aspects to evaluate the performance of the design.

After computers were involved in the design process and in building network systems, the use of computers in collaborative design for routine tasks and for intelligent design support systems became a reality. Several collaborative design methodologies have been proposed. Most of them are based on the distributed artificial intelligence methodologies to deal with the communication, control and coordination aspects of a distributed system. Since the designer’s experience is a very important factor in design, several methodologies integrated this experience into collaborative distributed software architecture. One of the early collaborative design studies for mechatronics dates back to the nineties and was named PACT (Palo Alamos Collaborative Testbed)[Cutkosky et al. (1993)]. In PACT, the proposed methodology was in simple terms to wrap up the tools and knowledge bases from the different vendors to make it work together for collaborative design. The research along this line has been followed by similar projects First-Link [Park et al. (1994)], Next-Link [Petrie et al. (1994)], and Process-Link [Petrie (1997)], which can be seen as the first important projects of multi-agent design systems. The detailed description and review of the multi-agent design systems can be found in Shen et al. (2001). Control and collaboration mechanisms used in the above-mentioned systems will be discussed in the multi-agent systems section.

A-Teams (Asynchronous Teams) is defined in Talukdar et al. (1998) as, “An A-Team is a problem-solving architecture in which the agents are autonomous and cooperate by modifying one another’s trial solutions”. An A-Team combines features from a number of systems particularly insect societies, cellular communities, genetic algorithms, blackboards, simulated annealing, tabular search, and brainstorming. The idea behind the combination of all these techniques is described again in Talukdar et al. (1998) as “we have been working on ways to organize algorithms so they can suppress their weaknesses through cooperation, and together do what separately they might not”.

The next kind of design methodology has recently been proposed by Campbell et al. (1999). It is based on the idea developed in A-Teams but is further improved by proposing preservation of good individuals during search. The proposed design methodology, A-Design, combines aspects of multi-agent systems, multi-objective optimisation and automated design synthesis. The proposed approach suggested for conceptual stage of engineering design and defined in Campbell et al. (1999) as “A-Design – a new search strategy for the conceptual stages of engineering design that incorporates agent collaboration with an adaptive selection of designs.”

The following sections will describe the intelligent agents, multi-agent systems and proposed multi-agent system architecture for the ship design.

2. Intelligent agents

Before going into detail about intelligent agents, we must give the definition of an agent, although there is not one consistent definition. The ‘agent’ verb comes from the verb ‘ager’ which means “to drive, lead, act or do” and it is described in the American Heritage Dictionary as “one that acts or has the power or authority to act or represent another” Bradshaw (1997). There are basically three types of agents: mobile, software and physical agents (robots).

Software agents vary greatly in the literature and on the INTERNET, some of them dealing with retrieving customer oriented web pages, some others try to find best book deals for their users. Although agents vary incredibly, they reflect some basic properties, Bradshaw (1997)
• Autonomy
• Proactivity
• Reactivity
• Adaptivity
• Collaborative Behaviour
• Mobility

First of all, we should give the definition of an intelligent agent, before dealing with the above-mentioned properties that an agent fully or partially consists of. Intelligent agents are the building blocks of the artificial intelligence, their differences come from rationality. Intelligent agents are the agents which do the things rationally in a given situation (rational agents)[Russell et al. 1995]. Now, we can start briefly overviewing the afore-mentioned basic properties.

Autonomy is the ability to have self-activation mechanism. In certain situations an intelligent agent should be able to sense and act, for example in the case of an intruder’s attack on the main server, the intelligent agent for network security should trace the problem caused by intrusion by sensing and tracking down the intruder’s attack and blocking it. While we are concerned with the autonomy, there must be available modules for an agent to detect or perceive the environmental changes. That can be accomplished sometimes via sensors, for example photo sensors in the case of a robot or monitoring software for the intrusion detection, and it must have some kind of effectors to change the environment, in which the agent is situated.

The proactiveness is also one of the important aspects of agent architecture, but implementation of proactiveness for a design agent is especially difficult for the distributed design problem solving. Proactivity is the property of acting towards a goal. Goal definition of an agent can be defined for improving its utility but it is hard to get an overall picture of the design process and act towards that, in the case of a global goal. Therefore, considering the current state of knowledge human intervention is crucial at this point.

Reactivity is another key factor in the agent systems. Reactivity is the direct reaction of an agent to an event, which could be responding to another agent or environmental perceptions, without reasoning about the event. Adaptivity is another property of an agent; an agent should evolve and adapt when it is situated in different environments. The most important aspect among the intelligent agent properties for concurrent engineering design can be thought of as the collaboration part; our agent must be collaborative and cooperative with other agents in the environment. Collaborative behaviour could be a solution of a problem or analysis of a system by more than one agent. Sometimes, an agent cannot deal with the problem alone and it needs another agent’s involvement. That is why collaborative behaviour is a really important feature and one of the basic distinctive properties of an intelligent agent when compared to expert systems.

Mobility is another distinctive property and in the engineering domain, we believe that the analysis software will be provided in mobile agent form and by migrating through networks it will deal with different organisations’ problems throughout the INTERNET.

The above mentioned aspects are divided into three categories in agent research literature;

• Reactive (Direct Reaction to the environment without reasoning, Subsumption Architecture)
• Deliberative (Goal Driven, Belief, Desire, Intention)
• Hybrid Approaches (Combined Approaches, Touring Machines, INTERRAP)

More detailed description of the architectures of the above is outwith the scope of this paper and further detailed definition on those listed architectures can be found in Wooldridge (2002).
3. Multi-agent systems

Multi-agent systems are distributed artificial intelligence systems, which have the properties of distribution, communication and coordination. Decentralization and distribution of expertise are the key factors of the multi-agent systems’ use in the engineering design.

The Communication is the one basic property of a multi-agent system. The communication among agents can simply be divided into two categories as our language does, one is syntax and the other is semantics. Agents must use the same kind of syntax and semantics to communicate. In the first category, the basic aim of the syntax is just text-processing or parsing, although it can be further complicated to voice recognition or handwriting, which nowadays are quite popular especially after the latest technologic and software developments. There are basically two categories of the messaging syntax used, KQML (Knowledge Query and Manipulation Language) and FIPA-ACL (Foundations for Physical Agents-Agent Content Language), which are all based on speech-act theory. Detailed review of both content languages is given in Wooldridge (2002). Semantics is the second important property of the communication aspects of the multi-agent systems. There are currently two approaches for semantics language: KIF (Knowledge Interchange Format) and FIPA-SL (FIPA Semantics Language).

The second property of the multi-agent systems that must be assessed is the control of a multi-agent system. If we have only a central agent, which does all the planning task decomposition and task allocation of the multi-agent system, then it converges to a centralised control structure. There are different ways of controlling the multi-agent systems. One of them is so called federated architectures. Federated architectures are implemented in literature such as, facilitators, matchmaker agents or design boards. Facilitators and matchmaker agents simply deal with task sharing and communication of the agents. The communication among agents is done through the use of facilitators, which parse the messages and send it to the related agent or agents. The last control system is the autonomous control system, where basically there is no control over the agents (Autonomous agents know when to communicate and how to communicate). Performance analysis of these control structures can be found in Lejter et al. (1996).

The last property of the system, where we believe the problem has not yet been solved especially for design and real world applications, is the coordination aspect of the multi-agent design systems. There are many coordination mechanisms used in multi-agent systems, from game theory to auctions. None of them seem reasonable to implement in real world design problems. Catalogue of conflicts or case-based negotiation is much more practicable for the design systems and they are proposed in several multi-agent design systems.

3.1. Brief literature review on multi-agent design systems

As mentioned before, in pioneering research studies and a sequence of the multi-agent design system research projects have been started by PACT as a consequence of First-Cut and Next-Cut projects at the Center for Design Research in Stanford University. The main idea behind PACT is described in three categories, Cutkosky et al. (1993). “Cooperative Development of Interfaces, Protocols and Architecture; sharing of knowledge among systems that maintain their own specialized knowledge bases and reasoning mechanisms; and computer-aided support for the negotiation and decision making that characterize concurrent engineering.” “What PACT actually demonstrates is a mechanism for distributed reasoning, not a mechanism for automatically building and sharing a design model.” PACT uses the federation architectures based on the facilitators, which do all routing jobs for agents. Agents do not communicate directly with each other but through the use of facilitators. Communication protocol used in PACT is based on KQML and KIF.

PACT showed the working system of the distributed design problem solving by using different wrappers (engineering tools). Following PACT, First-Link has been developed and in this project the most significant part was the introduction of the Central Node, Central Node is the directory service to
contribute to the architecture in three ways, which is stated in Park et al. (1994), “1) Maintenance of the adjacency matrices, $U$ and $G$; 2) provision of a directory service for inter-agent notifications, mail and requests; and 3) version management of the various designs”. Above mentioned $U$ and $G$ are the adjacency matrices. $U$ is used to determine which agent uses domain-feature $F_i$. $G$ is used to determine which agent affects or generates a specific domain feature $F_i$. First-Link also includes version control mechanism, check-in and check out concept. First-Link architecture is only limited to small number of agents since the size of the adjacency matrices would grow and the time to find effected agents and vice versa would be a problem.

Next-Link [Petrie et al. (1994)] uses both First-Link and REDUX [Petrie (1992)] in order enable the system to work without the information needed about the domain knowledge. To achieve that, a subset of REDUX has been developed, Redux’ [Petrie (1993)]. The Redux’ system is used in this project in order to inform agents of the position changes and design revisions and opportunities.

Following Next-Link, Process-Link is proposed to improve the system further, by adding the management side of the design. Redux’ system is further developed and modified. The changes are Goal Assignment Validity, Input Change Notification, Thrashing and Goal Blocks. Procura [Goldmann (1996)] model is developed for the design management of the project. Procura is principally developed to allow planning and scheduling of agent-based design projects in a hierarchical top-down approach.

One of the important studies on multi-agent design systems and intelligent agent architecture for the design is proposed in SIFA (Single Function Agents)[Dunskus et al.1995]. In this study, agents are defined by three parameters, target (what to work on), point of view (what to consider) and function (what to do). Different kinds of the agents are formed to change and evaluate the parameters, evaluator, selector, raiser, and so on. In this multi-agent design system, a conflict catalog approach, similar to Klein’s proposal [Klein (1991)] is used. Catalog of Conflicts approach simply categorise the conflicts between agents, (raiser, evaluator) and after categorising conflicts, either a human agent intervenes or hard-wired conflict resolution is used.

There are a few implemented agent-based design systems for the ship design in literature. One of them is represented in Parsons et al. (1999). In that paper, a systematic market approach developed in RAPPID (Responsible Agents For Product-Process Integrated Design) project is used as the multi-agent system architecture. Agents are assigned to different tasks, for example, resistance agent, manoeuvring agent and then market-based coordination (through bidding and selling) is used to find the solution of a design problem.

The other design system for the ship design is presented in Fujita et al. (1999). Importance in that system is given to the communication parts of the agents and an experimental system has been created for basic ship design studies.

The last system we can mention on agent-based ship design systems is developed by Lee et al. (2002). In their proposed system they are mostly focused on the negotiation part of the multi-agent system. Case-based negotiation mechanism is used for solving conflicts.

4. **Proposed multi-agent system for ship design**

The current systems in the literature generally deal with either distributed design control or design coordination. However, the above-mentioned systems implement the design analysis and exploration processes partially or they do not deal with design analysis and exploration at all. The proposed architecture has similar properties with the above-mentioned design and multi-agent design system studies but it differs from above-mentioned systems since in the proposed system we are more concerned with design decision support, not distributed problem solving without user interaction. So, our system is built by taking into consideration the innovation in ship designs by hybridising computational search and human thinking. Another difference comes from the design exploration,
which has been done with specialised operators and guided multi-objective optimisation processes for ship designs. In the proposed system, we connect all the exploration and analysis into a robust system. Our system has the capabilities of exploration of the design from different perspectives by using coupled multi-criteria decision making tools in distributed agent architecture. We further reduce the time spent on future design problems by introducing learning modules for extracting rules from the designers’ decisions.

The proposed system combines different artificial intelligence principles into a solid and implementable framework and it further develops them in order to tackle engineering design problems, particularly for ship design.

In order to achieve the proposed multi-agent system, there must be efficient architectures for both intelligent agent and multi-agent systems. Since current architectures are not readily suitable for our purposes we developed our architectures to enable us to explore and implement our studies for future research and ship design problems.

4.1.Intelligent agent architecture

The proposed intelligent agent architecture is defined in a three layers approach (likewise in hybrid intelligent agent architectures), which are communication, coordination and task layers (Figure 1). The communication layer deals with aspects such as sending and retrieving messages, message queuing, message routing with the other layers, and user interface. The coordination layer is the second layer in the agent architecture (there is no hierarchical order in layers; consequential order is an unavoidable fact). Four sub-levels construct the coordination layer.

The first sub-layer is the acquaintance module. The acquaintance module simply lists the other agents and restores the information related to the agents in the multi-agent system. We should point out here that we are not trying to create a complete intelligent system as in today’s technology that is not a reality. What we are trying to do is to give the decision support to the designer via running simulations, which can be in some cases automatically done or via multi-criteria decision-making tools. The proposed system does not deal with the versioning and dependency issues for the time being.

The second sub-layer in the coordination is the conflict resolution module. It is inspired by Mark Klein’s work [Klein (1991)] on collaborative engineering; the idea behind this methodology is to create hierarchical or organisational conflict resolution strategy before the conflict occurs. We are going to follow the same kind of idea, but without going into a detailed description of the conflicts; when a serious problem arises the designer will override the problem. Conflict resolution is divided into two groups one is rule-based and the other one is the case-based conflict resolution strategy. The selection of conflict resolution strategy is purely connected to the nature of the problem. Sometimes there is no need to use it but in some cases, there can be a fair combination of the rule and case-based strategies, however these sub-layers need to be further detailed and investigated especially for fair hybridisation of the rule and case-based conflict resolution.

The third sub-layer in the coordination module is the optimisation module. The optimisation module can work in conjunction with the optimiser agent or it can give the desired decision support to the designer by using local or global search algorithms.

The fourth and last sub-layer, learning module, generally deals with the learning capabilities of the system. As we can see from Figure 1, it has a close connection with the conflict resolution strategy. The task layer deals with the knowledge base and the simulation tools. Queries to the knowledge base or performance calculations are done through this layer.

The communication layer deals with communication amongst agents and designers. Communication aspects will be further specified in the next chapter. The user interface is where a human agent
intervenes in the process. Since we want to enable a human agent into the design loop where it is required we must develop easily understandable user interfaces for the human agent; in some cases it may be virtual reality (by contracting virtual reality agent), simple GUI or via 2D/3D chart representations.

Fig.1. Intelligent Agent Architecture

4.2. Multi-agent architecture

In our proposed architecture there is no such high level controlling mechanism for the intelligent agent as there is in blackboard systems over the knowledge bases. Agent communication is going to be done by using FIPA-ACL (Agent Content Language) and for the semantics FIPA-SL (Semantics Language) will be used. Data transfer among agents is going to be done by internationally accepted data exchange formats for engineering such as IGES, STEP or STEPML (bridge between XML and STEP).

Overall implementation of the multi-agent system will be organisational. Agents must be able to work in different geographical places and different networks on the Internet and they must be ruled by their organisation; for example, an agent should not be able to publish experience of the company and agents are limited to the resource bounds of the company. In an organisational coordination scheme agents are defined according to their roles, their positions, resources and limits. So, agents are defined beforehand.

Coordination amongst agents will be guided by either designer (via Multi-criteria Decision Maker Agent or single performance checks) or optimisation agent. There may be also contracting mechanisms (contract net protocol) for task sharing.

There are three types of agents in multi-agent architecture (Figure 2): Worker agents, Decision Theoretic Agents and User Interface Agents. However, there is no hierarchical order as they all have the same kind of authority. Worker agents do the jobs generally used for simulation and production, such as dynamic stability simulations or resistance. Multi-Objective optimisation agent helps the designer to do sub-design optimisations from different perspectives; here optimisation algorithms must be well connected to the worker agents to use knowledge bases as well as objective calculations.
for improving optimisation. The Multi-Criteria decision maker agent is going to be used to choose the best option from the Pareto-optimal solution set resulted from the multi-objective optimisation. Another use of the decision maker agent is obviously for selection of the materials, main engine or propulsion system.

The user-Interface agent is used for the easy evaluation of the design from operational and design performance perspectives. Its use can be further extended for crew training purposes by providing 3D-Real Time Simulations (Game alike) or virtual reality (again this can be further extended to the distributed virtual reality).

The system proposed is not a complete design system since there is a neither product model nor versioning and consistency checking procedures for the product model. The main aim of this architecture is to create a framework for design analysis and exploration. The data transfer in the system is done through either data transfer, by using directory services in the practical implementation of the system, or via the use of the mobility capabilities of the system.

![Multi-Agent System Architecture](image)

Fig.2: Multi-Agent System Architecture

5. **Illustrative example: Engineering design optimisation**

The advantages of using proposed distributed architecture in a ship design optimisation problem can be listed as;

- **Creation of Alternatives**, Initial Population creation or inserting new individuals into population
- **Modification of Alternatives**, by the help of specialised recombination and mutation operators
- **Guiding the search**, Goal Attainment method, or by the help of expert knowledge during the search.
- **Exploring the initially unfeasible search space**, by softening constraints or variable bounds
- **Qualitative fitness assessment** by the help of knowledge bases, when there is no assessed objective function, or when the objective is attained in practical terms.
The example is chosen to show the capabilities of the system from the automation perspective as well as the use of simulation tools. There can be other examples to show the implementation of the system such as the selection of the superstructure’s material from a cost and structural point of view or designing of an anchoring system. However the best worked example is chosen to show the working of multi-agent design system. The example is an engineering design optimisation problem from a resistance and stability point of view (Figure 3).

![Diagram](image)

**Fig.3: Engineering Design Example**

The optimisation works in the following way: a human designer launches the multi-objective optimisation agent (MOOA) to do design optimisation from a resistance and stability point of view. MOOA then broadcasts the resistance sub-task (actually this is the most ideal way of working of the system because the resistance agent can be situated over the INTERNET and there can be more than one resistance agent and it may not be in acquaintance model of the agent). However in our assumptions on acquaintance, MOOA agent uses its a priori knowledge about whom to connect to for resistance calculations, gets the bids for that task and contracts the job to one of the bidder resistance agents, the same goes for the stability agent as well as the hull generation agent. The hull generation agent (Figure 4) uses its knowledge base to create sound hull forms and gives that hull form in a way that stability and resistance agent can do their calculations. For example, only the hull section information would be enough for resistance calculations however we might need hull surface for stability calculations. The help of file transfer agent implemented in a directory service solves the file transfer issues in the multi-agent system. Resistance and stability agents are used here for only calculation purposes.

The second phase of the optimisation process is the modification of the individuals in the population via recombination and mutation operators. Basically genetic algorithms based multi-objective optimisation algorithms use either one-point or two point crossover operators, which are not specialised for the domain, where the optimisation proceeds. Although randomisation is a good way of exploring the search space, however it may increase the optimisation cost. By allowing both specialised operators and random operators in optimisation would reduce the time cost as well as keeping the exploration of the search space, either with random or specialised recombination and mutation operators. In this study we are not applying specialised operators for recombination and mutation operators, however in future it will be developed for the better exploration of the design solutions.

The third phase of the optimisation process is the evaluation of the individuals in order to decide who should be selected for the next generation. In this phase generally individuals are selected by using Pareto-dominance principle. However, we can add goal information to achieve more focused Pareto-front. This phase is actually related to the integrated decision making and search approaches used in multi-objective optimisation. Basically in here we are going to use Pareto-dominance principle only. In future, goal information will also be embedded to the system.
The last phase of the optimisation is the selection of the best solution (The best individual in genetic algorithms terms). The multi-attribute decision making agent gets involved in the process and either uses decision maker’s ratings (while interacting with decision maker’s ratings the learning module of the multi-attribute decision maker agent is in progress of supervised learning for the other two agents, resistance and stability agents) or uses resistance and stability agents to give the preference information for selecting the best individual (from the point of decision makers or agents). The TOPSIS (Technique for Order Preference by Similarity to Ideal Solution), AHP (Analytic Hierarchy Process) algorithm is used for the selection of best solution among Pareto-optimal solutions.

The proposed system obviously improves the performance of the optimisation problem by the distributed execution of the objective function calculations, although here master/slave approach used for parallel processing in future the system can be extended to include island models. Secondly by the integration of the knowledge used in hull generation agent, optimisation search time can be reduced. By the integration of interactive expert knowledge and implementation of goal programming, optimisation search cost can be further reduced.

<table>
<thead>
<tr>
<th>Communication Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface Module, To get user’s modifications to randomly or by case-base or rule-base interactions, and store the modifications and learn from the changes done by the hull form designer and store learned experience in knowledge base.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coordination Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquaintance Module</td>
</tr>
<tr>
<td>Stability Agent (Server: OSMAN:1099, Name: StabAgent),</td>
</tr>
<tr>
<td>Resistance Agent (Server: BEKIR:1099, Name: ResAgent),</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised Learning from the designer’s interactions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimisation Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Optimisation Routines for Geometrical Fairing (It is either wrapped from the tool for hull generation or developed)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions Wrapped or Implemented by the Agent</td>
</tr>
<tr>
<td>• Create a V type bulbous bow, ROPAX Vessel</td>
</tr>
<tr>
<td>• Create a A type bulbous bow, ROPAX Vessel</td>
</tr>
<tr>
<td>• Create a bulbous aft</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Knowledge Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>If running angle at waterline is higher than 25° then reduce the first five sections from bow by 5 percent.</td>
</tr>
<tr>
<td>If bow flare is higher than 30° then reduce the flaring angle</td>
</tr>
</tbody>
</table>

Fig.4. Hull Generation Agent

Realisation of the illustrative example is not very far away. Most of the parts of the system have been implemented separately by different people, but not collectively for engineering design. The combination of them is a challenge, since a lot of expertise is needed to build a working system: Multi-criteria decision-making, knowledge based systems, learning mechanisms, agent systems, programming and wrappers for the engineering tools. However, the progress is very promising and
the illustrative example seems reasonably achievable. The tools, which will be used for solving this optimisation problem, are also investigated. JADE (Java Agent Development Environment) [JADE], Jess (Java Expert System Shell) [JESS], and in-house developed multi-objective optimisation and multi-attribute decision making software or in some cases via FRONTIER Java Plug-in structure [FRONTIER] will enable the working of the system in a real world case study.

6. Discussions and conclusions

This paper gives a brief overview of the intelligent agents and multi-agent systems, and proposes multi-agent system architecture for analysis and exploration of ship designs, so it is meant to be used in the conceptual design stage. Implementation of such a system will always be problematic because the currently available technology is not “intelligent” enough to embed easily every layer of the system. In this proposed architecture, human intervention is a crucial fact as it is the case for almost every design. However this system gives the designer available tools in a compact and easy way to use. As the illustrative example is not far from reality and we believe that it will give us a way for the solutions of the other design problems that could be faced in future.

In the proposed architecture, two important aspects are missing. The first one is that there is no computer aided geometry interface for the geometrical design of a ship. The first missing part is only partially implemented in hull generation agent and it is based on the already developed hulls. Secondly, there is no consistency manager for distributed design. The two mentioned missing parts, CAD and consistency manager are thought to be embedded in to the system in later stages either by embedding a commercial CAD tool, since the issues related to the CAD and consistency checking is well addressed in commercial design software, or by embedding our system on top of a CAD/CAM/CAE ship design software. We believe by using the latter that the system would be more compact and robust.

Ongoing research projects in Europe are trying to assess the software needs for the ship design and shipbuilding industry and hopefully the tools will be available for future real-world testing of the system for more complicated test cases.

The system is still in a development process and the further development paths will be:

- **Including designer’s preferences into optimisation search**: Both multi-objective optimisation problem solver and multi-attribute decision-making tools are developed seperately. We are looking into combining them together in order to reduce the optimisation cost. The current literature has been reviewed and a particular method or new developed method will be used in this part of the system.

- **Learning from designer’s preferences**: It has not implemented yet, however the implementation would be easier since it is an established research area and many research papers and software available in the literature.

- **Specialised Recombination and Mutation Operators**: Specialised operators will be proposed and hybridising the optimisation process with other optimisation methods will be investigated.

The system is very promising for future research. Future research paths will be:

- **Integrating agents in a distributed virtual environment**: This research area is very new and it is called intelligent virtual environments, integrating such a system into the virtual environment dynamically and directly would cause additional benefits for the designers, for example, detecting the design errors at an early stage with online decision support via walk-
through in the ship. Moreover, changing the available geometry in virtual environment and assessing the performance of the system by the help of the intelligent agents.

- **Better Learning Capabilities:** The learning mechanism although included in the agent architecture is not implemented for the conflict resolution part for the current state of the system. However, case-based learning and other learning mechanisms are very promising candidates for the improvements to the proposed system.

- **Integration of Agent-based Optimisation Algorithms:** There are reactive agent-based optimisation algorithms, which use ant-colonies or nature of species as an analogy to solve optimisation problems. This kind of optimisation algorithms can be easily integrated with the proposed structure.

- **Better Autonomy and Constraint Satisfaction:** As we discussed before, autonomy is a very important feature of an agent, but implementation of autonomy is quite difficult especially when costly simulation is involved. In future implementations, agents can also be designed to perform simulations in a virtual reality system or as an add-on to CAD/CAM tool.

Lastly, this paper gives the capability of the formal methodology for ship design analysis and exploration by encapsulation of the knowledge and current artificial intelligence techniques in a compact and implementable way.

**References**


Artificial Neural Networks For Prediction of Ship Resistance and Wetted Surface Area

Kourosh Koushan, Marintek, Trondheim/Norway, Kourosh.Koushan@marintek.sintef.no

Abstract

New methods are presented for prediction of ship resistance and wetted surface area of ships based on analysis of database of tests performed in the towing tank of MARINTEK. Artificial neural networks method is applied for analysis of the database. The methods are verified using several towing test results available. These methods show generally reliable simulation of residual resistance and wetted surface area of the ships.

1. Introduction

There has been a significant development in the field of numerical calculation of ship resistance in the recent decade. These have led to useful tools for detail analysis of ships. Inevitably require all these tools a complete physical description of vessel. However at an early design stage require naval architects often a tool for reliable prediction of ship resistance based on few main parameters, usually Froude number and some geometric coefficients. Again there have been remarkable efforts to cope with this situation and several empirical methods are developed and optimised over the years applying regression analysis, e.g. Holtrop (1984) and Hollenbach (1998).

Experience has shown that all these methods can predict some cases very well whereas in some other cases predictions might not be as reliable. The method presented in this paper applies artificial neural networks (ANN) for the analysis of the extensive database of towing test results performed at MARINTEK in recent two decades to predict residual resistance coefficient. ANN method allows for non-linear effects and interdependence of input parameters. The database is divided into several ship categories and a network is designed for each category, allowing a more accurate prediction for each category. For the first time a reliable empirical method is developed for prediction of resistance of offshore vessels and car ferries. Objective of the method is to keep the number of input parameters low however at the same time deliver a reliable prediction for an early design stage.

To predict not only the resistance coefficient but also the resistance force, every ship resistance prediction method requires a reliable prediction of wetted surface area. Presented method includes prediction of wetted surface area based on main geometric parameters using ANN method. Again different networks are designed for each ship category.

ANN method offers the unique capability of easy adaptation to new data, which makes the developed methods quasi-dynamic models. That means with the access to new measurement data, developed networks can be upgraded with minimum amount of efforts. Networks are translated to DLLs that can be implemented easily to existing software. ANN is successfully used at MARINTEK for rudder force prediction, Koushan and Mesbahi (1998), and prediction of propeller induced pressure pulses, Koushan (2000).

2. Ship Resistance from Model Tests

MARINTEK’s standard definition of residual resistance coefficient is applied. It is assumed that the total resistance coefficient $C_T$ is divided mainly into the viscous resistance and the residual resistance coefficient $C_R$, where the latter is due to vorticity, wave making and wave breaking. The following equation is applied for resistance coefficients:
\[ C = \frac{R}{\rho \cdot S \cdot V^2} \]

Where \( R \) is the resistance, \( \rho \) is the density of water, \( S \) is the wetted surface area and \( V \) is the ship velocity. The residual resistance coefficient is defined as:

\[ C_R = C_{Tm} - C_{Fm} \cdot (1 + k_o) - C_{AA} - C_{BD} \]

The base drag coefficient \( C_{BD} \) is calculated applying the wetted area of the transom stern \( S_B \).

\[ C_{BD} = \frac{0.029 \cdot (S_B / S)^{3/2}}{(C_F)^{1/2}} \]

The air resistance coefficient \( C_{AA} \) is calculated using transverse projected area above the waterline \( A_T \).

\[ C_{AA} = 0.001 \cdot \frac{A_T}{S} \]

The frictional resistance coefficient \( C_F \) is based on the ITTC-57 correlation line, which is a function of Reynolds number \( Rn \).

\[ C_F = \frac{0.075}{(\log Rn - 2)^2} \]

The form factor \( k_o \) is calculated as:

\[ k_o = 0.6 \cdot \frac{C_B}{L_{WL}} \sqrt{T_{AP} + T_{FP}} \cdot B + 145 \cdot \left( \frac{C_B}{L_{WL}} \sqrt{T_{AP} + T_{FP}} \cdot B \right)^{3.5} \]

The total ship resistance coefficient is defined as:

\[ C_{Ts} = C_{Rm} + (C_{Fm} + \Delta C_F) \cdot (1 + k_o) + C_A + C_{AA} + C_{BD} \]

Residual resistance coefficient \( C_R \) can be predicted by the artificial neural networks presented in this paper. The result is correlated against full-scale trials applying a resistance correlation coefficient \( C_A \), which usually varies for different towing tanks. MARINTEK usually applies resistance correlation coefficient values between \(-0.20 \cdot 10^{-3}\) and \(-0.23 \cdot 10^{-3}\) for conventional ships.

The roughness allowance \( \Delta C_F \) is calculated using hull surface roughness \( H \) in \( \mu = 10^{-6} \) m. Typical value of hull surface roughness is 150 \( \mu \). Only positive values of \( \Delta C_F \) are used.

\[ \Delta C_F = \left[ 110.31 \cdot (H \cdot V_s)^{0.21} - 403.33 \right] \cdot C_{Fm}^2 \]

### 3. Prediction Using Artificial Neural Networks

Prediction methods are based on measurements performed in the towing tank at MARINTEK in recent two decades. Analyses of the database are performed using Artificial Neural Networks Method. ANN is a network of many simple processors (processing elements), each probably having a small amount of local memory. The processors are linked by communication channels (synapses), which usually carry numeric data, encoded by any of various means. The units run only on their local data and on the inputs they receive via the communication channels.
Networks have some sort of "training" rule whereby the weights of synapses are adjusted on the basis of data. In other words, artificial neural networks "learn" from examples (here model test results) and show some capability for generalisation beyond the training data. Simple linear regression (a minimal feedforward net with only two processing elements plus bias) is usefully regarded as special cases of artificial neural networks. Fig. I demonstrates main components of a feed-forward recall artificial neural network. These are input layer, synapses, one or more hidden layers (axons) and an output layer.

![Diagram of a feed-forward recall network](image)

**Fig.1:** Main components of a feed forward recall network with a single hidden layer

Input layer is used to feed the network with data. Synapse is connecting different layers Synapse has a weighting factor, which is obtained during design of the network. Hidden layer is made of one or more processing elements and corresponding activation function, which transforms the input of processing element. Typical activation functions are linear, sigmoid and tanh functions. Output layer operates usually like a hidden layer and can in addition denormalise the output.

Back propagation procedures is used for training the network. In a backpropagation procedure, network starts with a random set of weights. Then network output is compared to model test results and the error is verified. Then these errors are propagated backwards through the network to find better weights. Apart from weights and activation functions, number of hidden layers and number of processing elements in each hidden layer must also be optimised during the training. Usually one half part of database is used for training whereas the other half is used for verification of the network, i.e. the network does not “see” verification set during optimisation process.

The database includes 487 ships and 3481 measurement points. A preliminary analysis of the database shows that higher accuracy can be achieved by using different neural networks for different categories of ships. Therefore one ship type or several similar ship types are grouped in one category. This allows using different input parameters according to sensitivity analysis performed for each group. The pre-study led to five categories. Categories, number of ships in each category and number of measurement points available in each category are presented in Table I. If a ship type is not represented in the selected categories, closest category may be selected for the prediction.

<table>
<thead>
<tr>
<th>Category</th>
<th>Car ferries</th>
<th>Passenger &amp; cargo</th>
<th>Tanker &amp; bulk</th>
<th>Offshore</th>
<th>Fishery</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of ships</td>
<td>26</td>
<td>163</td>
<td>158</td>
<td>64</td>
<td>76</td>
</tr>
<tr>
<td>No. of measurements</td>
<td>228</td>
<td>1142</td>
<td>935</td>
<td>525</td>
<td>651</td>
</tr>
</tbody>
</table>

The input parameters chosen for each category and their validity range are presented in Table II for residual resistance prediction and in Table III for prediction of wetted surface area. $Z$ is the number of propellers, $B$ is the breadth, $L_{WL}$ is the length of waterline, $LCB$ is the longitudinal centre of buoyancy relative to $L_p/2$ (half of length between perpendiculars, positive forward), $C_B$ is the block coefficient, $FN$ is the Froude number, $C_M$ is the mid-ship coefficient and $T$ is the mean draught.
Table II: Selected input parameters and their validity range for each category
(Residual resistance prediction)

<table>
<thead>
<tr>
<th></th>
<th>$Z$</th>
<th>$10 \cdot B_{WL} / L_{WL}$</th>
<th>$100 \cdot T / L_{WL}$</th>
<th>$100 \cdot LCB / L_{WL}$</th>
<th>$10 \cdot C_{RBL}$</th>
<th>$10 \cdot \text{FN}$</th>
<th>$10 \cdot C_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car ferries</td>
<td>1-2</td>
<td>1.25 - 2.20</td>
<td>3.05 - 5.75</td>
<td>-</td>
<td>3.00 - 6.30</td>
<td>1.6 - 3.50</td>
<td>6.30 - 9.85</td>
</tr>
<tr>
<td>Passenger &amp; cargo</td>
<td>1-2</td>
<td>1.15 - 3.41</td>
<td>2.65 - 8.22</td>
<td>-5.00 - 3.22</td>
<td>4.15 - 8.10</td>
<td>1.1 - 3.70</td>
<td>5.80 - 10.00</td>
</tr>
<tr>
<td>Tanker and bulk</td>
<td>-</td>
<td>1.30 - 2.20</td>
<td>3.7 - 7.30</td>
<td>-1.20 - 4.50</td>
<td>6.35 - 8.60</td>
<td>1.1 - 2.62</td>
<td>-</td>
</tr>
<tr>
<td>Offshore</td>
<td>1-2</td>
<td>1.96 - 2.80</td>
<td>5.95 - 9.70</td>
<td>-3.90 - 1.02</td>
<td>4.71 - 7.15</td>
<td>1.8 - 3.50</td>
<td>8.15 - 100</td>
</tr>
<tr>
<td>Fishery</td>
<td>-</td>
<td>1.79 - 3.50</td>
<td>6.00 - 19.0</td>
<td>-4.10 - 3.42</td>
<td>3.77 - 6.90</td>
<td>1.75 - 4.0</td>
<td>6.60 - 9.55</td>
</tr>
</tbody>
</table>

In case of prediction of wetted surface area, a category is defined containing all available data, this category is called “All”. Though this category has a broader validity range, however the prediction results may be not as accurate. The networks are verified within valid range of input parameters in the database.

Table III: Selected input parameters and their validity range for each category
(Wetted surface area prediction)

<table>
<thead>
<tr>
<th></th>
<th>$C_M$</th>
<th>$C_{RBL}$</th>
<th>$B_{WL} / T$</th>
<th>$T / L_{WL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car ferries</td>
<td>-</td>
<td>0.299 - 0.641</td>
<td>3.290 - 4.956</td>
<td>-</td>
</tr>
<tr>
<td>Passenger &amp; cargo</td>
<td>0.571 - 1.000</td>
<td>0.415 - 0.812</td>
<td>2.460 - 7.700</td>
<td>0.0261 - 0.0824</td>
</tr>
<tr>
<td>Tanker and bulk</td>
<td>-</td>
<td>0.616 - 0.864</td>
<td>2.179 - 5.037</td>
<td>0.0368 - 0.0764</td>
</tr>
<tr>
<td>Offshore vessels</td>
<td>-</td>
<td>0.468 - 0.719</td>
<td>2.301 - 3.789</td>
<td>0.0584 - 0.0981</td>
</tr>
<tr>
<td>Fishery</td>
<td>-</td>
<td>0.374 - 0.695</td>
<td>1.833 - 3.066</td>
<td>0.0597 - 0.1908</td>
</tr>
<tr>
<td>All</td>
<td>0.571 - 1.000</td>
<td>0.299 - 0.864</td>
<td>1.833 - 7.700</td>
<td>0.0261 - 0.1908</td>
</tr>
</tbody>
</table>

4. Verification

There are different statistical coefficients, which can define the quality of an empirical prediction. Average mean squared error $AMSE$, average absolute error $AAE$, standard deviation of error $St.Dev.$ and Pearson correlation coefficient $r$ are used here. Pearson correlation coefficient $r$ reflects the extent of a linear relationship between two data sets $X$ (measurement) and $Y$ (prediction).

$$AMSE = \frac{\sum (X - Y)^2}{\sum X} \cdot 100$$

$$AAE = \frac{\sum |X - Y|}{\sum X} \cdot 100$$

$$St.Dev. = \frac{1}{n} \sum \left[ \left( X - \bar{X} \right) - \frac{1}{n} \sum (X - Y) \right]^2 \cdot \frac{1}{n} \sum X$$
\[ r = \frac{\sum(X \cdot Y) - \sum X \cdot \sum Y}{\sqrt{n \sum X^2 - (\sum X)^2} \cdot \sqrt{n \sum Y^2 - (\sum Y)^2}} \cdot 100 \]

Table IV presents results of verification of predicted residual resistance coefficient related to measured residual resistance coefficient, Table V presents corresponding values for total resistance coefficient and Table VI for wetted surface area.

Table IV: Correlation factor, standard deviation of error, average mean squared error and average absolute error related to prediction and measured results of RESIDUAL RESISTANCE COEFFICIENT

<table>
<thead>
<tr>
<th></th>
<th>Total database</th>
<th>Verification set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r</td>
<td>St.Dev.</td>
</tr>
<tr>
<td>Car ferries</td>
<td>96%</td>
<td>14%</td>
</tr>
<tr>
<td>Passenger &amp; cargo</td>
<td>98%</td>
<td>13%</td>
</tr>
<tr>
<td>Tanker and bulk</td>
<td>99%</td>
<td>13%</td>
</tr>
<tr>
<td>Offshore vessels</td>
<td>98%</td>
<td>10%</td>
</tr>
<tr>
<td>Fishery</td>
<td>97%</td>
<td>13%</td>
</tr>
</tbody>
</table>

Table V: Correlation factor, standard deviation of error, average mean squared error and average absolute error related to prediction and measured results of TOTAL RESISTANCE COEFFICIENT

<table>
<thead>
<tr>
<th></th>
<th>Total database</th>
<th>Verification set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r</td>
<td>St.Dev.</td>
</tr>
<tr>
<td>Car ferries</td>
<td>96%</td>
<td>5%</td>
</tr>
<tr>
<td>Passenger &amp; cargo</td>
<td>97%</td>
<td>3%</td>
</tr>
<tr>
<td>Tanker and bulk</td>
<td>98%</td>
<td>2%</td>
</tr>
<tr>
<td>Offshore vessels</td>
<td>98%</td>
<td>5%</td>
</tr>
<tr>
<td>Fishery</td>
<td>97%</td>
<td>6%</td>
</tr>
</tbody>
</table>

Table VI: Correlation factor, standard deviation of errors, average mean squared error and average absolute error related to prediction and measurement results of WETTED SURFACE AREA

<table>
<thead>
<tr>
<th></th>
<th>Total database</th>
<th>Verification set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r</td>
<td>St.Dev.</td>
</tr>
<tr>
<td>Car ferries</td>
<td>87%</td>
<td>3%</td>
</tr>
<tr>
<td>Passenger &amp; cargo</td>
<td>84%</td>
<td>3%</td>
</tr>
<tr>
<td>Tanker and bulk</td>
<td>77%</td>
<td>2%</td>
</tr>
<tr>
<td>Offshore vessels</td>
<td>69%</td>
<td>4%</td>
</tr>
<tr>
<td>Fishery</td>
<td>70%</td>
<td>4%</td>
</tr>
<tr>
<td>All</td>
<td>70%</td>
<td>4%</td>
</tr>
</tbody>
</table>

The number of input parameters is kept low to obtain an easy to use method. This has the consequence that design special details and appendages like bulbous bow are not considered. Experience has shown that appendages may result up to 10% difference in total resistance. In general an average
accuracy of approximately 96% can be expected from total resistance coefficient prediction. For the wetted surface area an average accuracy of 97% can be expected. This leads to an average accuracy of approximately 93% for predicted total resistance force. As expected, the accuracy is higher in case of tankers and bulk carriers, due to more homogenous ship geometries in this category. On the contrary fishery vessels have less homogenous forms leading to a lower accuracy for this category.

Fig. 2 presents measured total resistance coefficient for a passenger vessel compared to prediction performed by Hollenbach, Holtrop and ANN. The test was performed after the ANN method was developed so that the data was not available in the analysis database. Prediction by ANN method is very good. Holtrop method predicts also the total resistance coefficient fairly well. However Holtrop method underestimates the wetted surface area by 13%. Wetted surface area is underestimated 5% by ANN method and 3% by Hollenbach method.

![Total resistance coefficient, PASSENGER](image1)

**Fig.2**: Predicted and measured resistance coefficient for a passenger vessel

![Total resistance coefficient, SUPPLY](image2)

**Fig.3**: Predicted and measured resistance coefficient for a supply vessel
Fig.4: Predicted and measured resistance coefficient for a tanker

Fig.3 presents measured and predicted total resistance coefficient for a supply vessel. The resistance test was carried out after the ANN method was developed. This example shows the excellent ability of the ANN method in predicting total resistance of supply vessels. ANN methods predicts the resistance coefficient fairly good and prediction of wetted surface area shows only 1.2% error.

Measured total resistance coefficients for a tanker compared to calculated values using Hollenbach, Holtrop and ANN methods are presented in Fig.4. Measurements were carried out after the ANN method was developed. Total resistance coefficient calculated by ANN method is almost matching the measurements. In addition the error in wetted surface area prediction by ANN method is only 0.3%.

5. Conclusions

The method presented is a novel procedure for prediction of total resistance. A reliable prediction method is presented for the first time for prediction of total resistance of offshore vessels. To keep the method easy-to-use, the number of input parameters are limited. This has the consequence of ignoring special design details like bulbous bow and appendages. These details may result up to 10% difference in the total resistance. Adding more input parameters will increase the accuracy for special cases, however without access to sufficient additional exemplars this may reduce generalisation of the method. The method can be further developed with the access to new measurement data.

References


Integration of CAD and CAM Systems in Shipbuilding

Reinhard Staebler, Reinhard.Staebler@atlantec-es.com
Matthias Grau, Matthias.Grau@atlantec-es.com
Atlantec Enterprise Solutions, Annapolis, MD/USA and Hamburg/Germany

Abstract

Flexible integration concepts for CAD and CAM systems have been identified as a key to let shipyards select and implement best in class software components for their CAD and CAM operations. Current implementations are dominated by bilateral links based on proprietary data exchange formats and are too complex to upgrade parts of a CAD/CAM infrastructure without negative impacts on the other parts. This paper describes the ongoing development of a connector architecture for CAD and CAM systems in shipbuilding. The architecture decouples CAD and CAM systems on the basis of a flexible integration technology, utilizing XML data exchange, LDAP, and message-based communication. An Enterprise Reference Model describing all relevant shipbuilding business objects forms the basis for the integration. So-called adapters connect the various CAD and CAM systems to the architecture. An automatic nesting solution is presented as a sample business solution in the connector architecture environment.

1. Introduction

Today most shipyards have implemented Information Technology (IT) infrastructures consisting of one or more CAD systems, connected to many different CAM systems using mainly bilateral links based on proprietary data exchange formats. The number of connections between these systems, as well as their complexity, has lead to a stalemate: the potential advantage in upgrading one part of the infrastructure is outweighed by the negative impact on other parts. Often connections between CAD and CAM systems are multi-staged, requiring additional data conversions and a high degree of manual input.

![Diagram showing connections between CAD and CAM systems](image)

Fig. 1: CAD and CAM systems connected through bilateral links

This paper describes an information architecture for CAD and CAM systems, a so-called connector architecture, that provides interoperability between typical CAD and CAM systems used in shipbuilding. The architecture features a flexible systems integration concept that allows shipyards to choose CAD and CAM software applications that best fit their design and production requirements. In addition information management and workflow management functions are provided.
The connector architecture has been designed to support complex integration scenarios, where a large number of different legacy information systems, installed at different locations, interoperate in the design and manufacturing processes. Technologies, like XML based data exchange, LDAP based directory services, and message based communication, have been utilized for this integration architecture.

An Enterprise Reference Model, a model that identifies and describes all relevant shipbuilding business objects, forms the basis for the integration. So-called publisher and/or subscriber adapters connect the various CAD and CAM systems to the architecture. An Information Server provides a directory of all relevant information, utilizing LDAP technology.

The connector architecture concept is validated with respect to its applicability to typical business cases, e.g. plate part nesting and robot welding of assemblies. A first implementation of an automatic nesting solution for plate parts in the connector architecture environment will be described in detail.

2. Decoupling CAD and CAM Systems

To overcome the stalemate caused by too many bilateral proprietary links, the decoupling of CAD and CAM systems is one of the primary goals of the connector architecture development. Decoupling of CAD and CAM is an integration task in the first place, despite the literal contradiction. The goal is to develop a flexible systems integration on the basis of a system-independent intermediate representation of the data.

The successful implementation of a decoupling methodology

- enables shipyards to select and implement best in class software components independent of their CAD and CAM counterparts,
- enables collaboration in design and production between shipbuilders independent of their respective CAD and CAM environments.

![Diagram: CAD and CAM systems connected through system-independent intermediate data representation](image)

Fig.2: CAD and CAM systems connected through system-independent intermediate data representation
3. Production Information Management

Current CAD/CAM implementations in shipbuilding provide production information in a variety of proprietary formats. Reports, drawings, NC-files and other documents are generated in a lot of different ways. There is no common, accessible data model and typically no common information storage and management component that takes care of the information elements and their relationships. Shipbuilders are forced to work on the least common denominator level, the production document (be it on paper or in a digital document management system) as opposed to working directly on information model level.

![Diagram](image)

**Fig.3: CAM systems generate a variety of reports, drawings, NC-files and other documents**

This situation prevents further improvements in cycle time and planning horizon reductions. The high concurrency of design, procurement and production processes in shipbuilding requires real-time access to all relevant information. Document based information does not come close to providing real-time access. First, production documents represent a specific, derived view on information without capturing the important interdependencies of information elements. Second, as soon as they are produced, they are out of date.

Consequently another important goal of the connector architecture is the development of a means to make manufacturing information available not only to other CAD and CAM systems but to planning and management type systems also, thus implementing common, improved production information management functions. It is anticipated that the ability to explicitly manage production information at the product-model level will

- stimulate process changes, improvements and automation,
- improve consistency of production information which in turn leads to quality improvements and reduction of rework,
- enable distributed, collaborative environments and
- enable the integration/participation of other stakeholders (e.g. the ship owner)
4. Connector architecture

4.1 Enterprise Reference Model

The connector architecture for CAD and CAM systems is based on the concept of a neutral reference data model, the Enterprise Reference Model (ERM). The ERM is a shipbuilding specific reference model that captures all relevant aspects of the design and production domain and has been developed by Atlantec Enterprise Solutions as part of the NSRP, ISE program. It covers product model information, processes, organizations and facilities. This model provides a neutral representation for the exchange of data between CAD and CAM systems.

Various standards exist for different information domains including

- ISO 10303 (STEP) covering the representation of product data (design and manufacturing)
- Standards defined by the Workflow Management Coalition (WfMC) covering the representation of process and facility data
- the X.500 standard (ITU) defined by the Network Working Group for organizational data

Those standards are only weakly or not at all interrelated. For an integrated view of information from these domains it is required to convert them into a combined view allowing to express existing inter-domain relationships. The Enterprise Reference Model (ERM) follows this approach and provides such a combined view on all kind of information that is relevant for CAD-CAM integration.

The ERM includes data structures from ISO 10303 (AP214, AP215, AP216, AP218, AP227), WfMC and X.500 while putting them on a common, related basis. It includes data structures for the representation of

- general product structures (design and manufacturing) as well as identity
- arrangement, space and compartmentation data
- ship hull form data
- steel part data (design and manufacturing)
- outfitting part data (design and manufacturing)
- processes and activities
- organizational data

Fig.4: Make manufacturing relevant information available to planning and management type systems
Information according to the ERM can be associative and topologically related on product structure level. Documents of arbitrary types and geometrical representations can be assigned to virtually every ERM object.

In order to support the transfer of neutralized model geometry, features and attributes from CAD to CAM systems, a number of specific model requirements exist. Most important are the differences between design and production geometry. While the design geometry represents the final, as build product shapes, the production geometry may differ significantly. This is obvious for curved steel parts, where a number of manufacturing steps, e.g. the burning, are related to a plane, developed representation. In general the manufacturing geometry of parts may differ with respect to

- access
- shrinkage margins
- bend allowance
- edge preparation and beveling

Also, there can be additional production geometry and attributes for elements like

- marking lines (for Cut lines; bend lines, center punch, near side, far side, etc)
- marking triangles (for measurement control devices)
- marking text

In addition to the geometry, features and attributes, object relations between design and manufacturing objects and administrative information for manufacturing objects play an important role.

### 4.2 Information Management Infrastructure

The CAD – CAM connector architecture utilizes an LDAP based Information Server, several CAD and CAM system adapters and a message based communication server that provides functions for Internet wide information distribution. An important aspect of the concept is a secure data exchange utilizing authentification, and access control.

![Connector Architecture for CAD and CAM systems](image)

*Fig. 5: Connector Architecture for CAD and CAM systems*
Information management functions are provided on top of the data exchange features. Efficient control and management of information for production is a key to further process and productivity improvements in the generation of production information and the production activity as such.

The specific information management requirements for the CAD – CAM architecture are:

- controlled release for production of CAD models
- version history and dependency of all CAD and CAM information handled
- management of NC-files and other manufacturing information
- control of information exchange and flow
- change management functions

Based on generic information exchange and management features the specific requirements for the CAD - CAM integration scenario are:

- **Status monitoring**
  The ability to capture and monitor the status of information objects (i.e. who or which system has processed information, which information has been derived). This enables tracking of object changes and automatic evaluation of the consequences a change might imply (e.g. a geometry change for a plate part requires a new nesting of that part).

- **Event based status concept**
  An event based status concept, i.e. the fact that the status of an object changes is an event that automatically triggers a series of required downstream activities/system runs.

- **Workflow and process support**
  Within the connector architecture the flow and use of information is modeled as a “process” or “workflow”. A workflow is defined through a process template that is written in a process definition language. A so-called process engine initiates and executes the process templates. This allows the Connector Architecture to control and monitor the information exchange processes. In the specific case implemented the process scripts would be a formal description of the information flow and would also allow the automation and Topgallant controlled run of the individual system steps. A typical example could be: The connector architecture triggers the run of system A when a specific file is uploaded, waits for the results of system A, and passes them on to system B.

The connector architecture concept includes functional modules and applications that allow specific activities to be performed - e.g. information viewing, selection and release – which are not provided by the CAD and CAM systems to be integrated. The functions will be implemented based on the Enterprise Reference Model representation of objects.

### 4.3 Connecting Systems through Adapters

CAD and CAM systems are connected to the architecture through so called “adapters”. Figure 6 illustrates this concept. Connector Architecture adapters implement the “publisher-subscriber” paradigm. These adapters allow transparent access to legacy information using published interfaces. They will then be combined with the connector architecture, thereby making different legacy applications interoperable.

Information sources, like CAD systems, are connected through Publisher Adapters. Based on an adapter framework they are custom-written for each system and may use a combination of the systems API’s, database tools and output files.
Fig. 6: Connecting CAD and CAM systems through adapters

Information consumers, like CAM systems, are connected through Subscriber Adapters. These adapters extract the information from the connector architecture in a neutral ERM representation and prepare it for the subscribing system, e.g. in the form of input files or database tables.

CAD system adapters publish design and production geometry, features and attributes to the Connector Architecture. If possible and supported by CAD system interfaces, the CAD system adapters automatically monitor the CAD model and identify changes in the product model. Changes are analyzed with respect to their influence on processes and systems that already use the previously published information. These adapters are based upon a generic CAD adapter framework, and are customized for particular CAD applications such as CATIA V5 or Tribon M2.

The “Native System Wrapper Import Layer”, Fig. 7, utilizes one or several interfaces or API’s of a CAD system.

Fig. 7: CAD Adapter – Information Publisher and Subscriber Concept
CAM system adapters subscribe to information from the Connector Architecture. They also can be used to publish data to the Connector Architecture. Part geometry and features that have been published in the connector architecture are retrieved and converted into a format that is appropriate for the target CAM system. This can also be a collection of native CAM system file formats.

Again, these adapters are based upon generic CAM functions, and are customized for particular CAM applications.

5. Automatic nesting for plate parts

The automation of plate part nesting has a potential for major cost savings in the preparation of manufacturing control information for piece part production. Therefore a solution for the automatic nesting of plate parts has been selected as a first application for the connector architecture environment.

In this context automation is the automatic generation and processing of “nesting jobs” for a given set of CAD models (e.g. one or several hull blocks). A nesting job is a grouping of piece parts according to shipyard specific rules, which can be nested on one or a number of plates. The automation eliminates the need to sort and group piece parts one by one into nesting jobs, which is a time and resource consuming activity for the several 10 thousand piece parts in typical ships.

The solution integrates the Tribon M2 CAD system and the Alma act/cut nesting system on the basis of the connector architecture. Additional functionality is provided through a nesting control application in order to support the automation of the plate part nesting. Fig.9 shows the concept for the nesting solution.

![Diagram of nesting concept](image)

**Fig.8: Automatic Nesting Concept**

5.1 CAD System Adapter for Tribon M2

The Tribon M2 Hull system was selected as one of the candidates for integration into the Connector Architecture. The Tribon Publisher extracts all required parts information from Tribon and publishes it in the Connector Architecture. This includes:

- hull design structure
- assembly structure
- hull parts including geometry and all production relevant features
- marking information

Tribon software is not equipped with a single interface necessary to extract all required information. The information from a number of different Tribon data export interfaces is combined for the implementation of the adapter. Design structure and geometry are combined with production geometry and marking information. The assembly structure is extracted through an additional Tribon interface. The adapter monitors Tribon model databases and automatically publishes changed information.

5.2 CAM System Adapter for Alma act/cut

The Alma act/cut nesting system has been selected as a first CAM system candidate for integration into the Connector Architecture. Alma is the market leader for nesting solutions in Europe.

To connect the Alma act/cut nesting system to the connector architecture a nesting adapter with both a subscriber and a publisher functionality are being developed.

The subscriber supplies the Alma modules with nesting jobs, i.e. groups of parts to be nested, and part geometry and features. The part geometry including edge preparation and marking details is supplied in Alma “.ini” file format. The nesting job description, including the raw material assignment, parts multiplicity and burning machine, is supplied in Alma “.rin” file format. The Alma specific directory layout for the file handling is taken into account.

The publisher returns nesting information to the connector architecture. Information published includes the nesting identification for parts and the raw plate consumption. The publishing is based on the processing of XML files that are generated by the Alma act/cut modules.

NC-files and burning sketches that are generated by Alma act/cut are also published, utilizing the information management features of the connector architecture.

5.3 Raw Plate Adapter

Raw plate data are required to generate nesting jobs for Alma. Information required includes the plate identification, thickness, material quality and size.

A file based publisher adapter publishes raw plate data in the connector architecture, which in turn controls the raw plate consumption/use. Raw plate information can also be retrieved from a material management system.

5.4 Automatic Nesting Control Application

An automatic nesting solution cannot be implemented just by integrating the CAD and CAM system. Additional nesting logic and interactive functionality that supports the preparation, launching and control of automatic nesting jobs need to be developed.

The nesting control application utilizes the neutral ERM representation of relevant information in the connector architecture. All application functions are developed independent of the CAD or CAM system. This methodology supports the addition of CAD or CAM system to the connector architecture while still being able to use the nesting control application.

Generic browser functionality is required to view and control all published CAD and Alma information. This includes a tree browser feature for model trees, detailed attribute viewing, and optional viewing of parts geometry.
Release for Production/Nesting
An application function allows the selection of hull blocks, panels or parts. The selected items can then be released for production, thereby initiating the generation of nesting jobs and the actual nesting of the parts. The release for production is implemented through the generation of an event in the information directory of the connector architecture.

Nesting Job Generation
The generation of nesting jobs is supported in a batch/automatic mode as well as in an interactive mode. The nesting job generation implements the “nesting logic” based on shipyard specific configuration information.

Parts are analyzed with respect to their type (plane, curved, knuckled, with/without marking), multiplicity and symmetry. Parts that have been released for production/nesting are grouped into nesting jobs using the following rules and criteria:

- material thickness
- material quality
- part type and size
- shipyard specific filter rules for part types
- selection of a suitable target burning machine
- edge preparations
- burning mode (normal, symmetric, parallel)
- marking side
- optional part rotation or mirroring

Suitable raw plates are selected from the stock of available plates.

By default the grouping is a fully automatic process. But it can also be performed in an interactive mode. In interactive mode it is e.g. possible to add parts with lower material thickness and quality to a nesting job or to manually add piece parts to existing nesting jobs.

Alma Controller
A controller component allows nesting jobs to run in batch mode. The controller component launches the Alma act/cut nesting modules in the required sequence. This includes

- Alma Drafter: the loading of nesting jobs and parts geometry
- Alma Nester: the nesting itself
- Alma PathFinder: the generation of cutting sequences including NC file generation
- Alma Fiche: the generation of burning sketches

If required, nesting jobs can also be processed interactively using the Alma act/cut modules utilizing the input generated by the nesting control application.

Reporting
A reporting component allows the generation of all sorts of list and file output of the parts and nesting information. Shipyard specific standard reports can be configured using report templates.

6. Conclusion

The ongoing development of the automatic nesting solution clearly demonstrates the potential of the connector architecture and the adapter concept. Even in a yet bilateral utilization of the connector architecture provides important additional information management advantages.

Anticipated advantages of connector architecture implementations include:
- New systems could be added or used to replace existing systems while causing a minimum of disruption to the ongoing production processes. The use of adapters to connect CAD and CAM systems to the connector architecture would allow an almost plug and play flexibility in system selection.
- Shipyards will be able to easily select modular software components to perform specific functions such as robotic welding, nesting or lofting, as opposed to being constrained to one single vendor-specific system.
- The CAD publisher adapters will automatically monitor the CAD data sources to identify any changes to information that have already been processed and provide a corresponding notification to the subscriber adapter for the CAM system. This will ensure that part data and attributes provided to the CAM system always represents the latest released version of the product model.
- Shipyards will be able to easily collaborate with other shipyards and design agents using different systems without a widespread migration to the same software.
- Shipyards would be able to provide just in time production data to subcontractors using different systems.
- Information generated by the CAM system may also be published through the connector architecture, thereby allowing an integrated view of the design and production information.

Next to come is a robot welding solution on the basis of the connector architecture environment.

**Acknowledgement**

The authors would like to thank the US Office of Naval Research (ONR), which is sponsoring the research and development described in this paper through funding from the Small Business Innovation Research (SBIR) program.

**References**


STEP, ISO 10303, *Industrial automation systems and integration – Product data representation and exchange*


Economic Feasibility Prediction for Welding Robots in Shipbuilding

Jenny Coenen, IHC Holland N.V. Dredgers, Kinderdijk/Netherlands, j.coenen@ihcdredgers.com
Ubald Nienhuis, Delft University of Technology, Delft/Netherlands, u.nienhuis@wbmt.tudelft.nl
Reinier Rijke, IHC Holland N.V. Beaver Dredgers, Sliedrecht/Netherlands,
  r.rije@beaverdredgers.ihcholland.com
Jan van der Wagt, Delft Univ. of Technology, Delft/Netherlands, j.c.vanderwagt@wbmt.tudelft.nl

Abstract

The current application of welding robots in Europe is mainly limited to large-scale operations at the largest yards. At these yards significant steel volume throughput is paired with a high content of similar structures. For the medium sized yards however, which in aggregate have a dominant share of the European shipbuilding market, this situation does not apply. Not only the moderate steel throughput is to blame, also their involvement in smaller and specialised ships lead to more complex shaped structures with a lower degree of repetitive elements. Nevertheless these yards are confronted with the same economical pressure and labour situation facing the big European yards. The challenge is to develop and implement robots that make sense economically in these cases. This requires the use of more versatile robots, of smaller size and hence investment cost, which are optimally integrated in the production environment.

Applied research projects, performed in a joint effort of Dutch yards and Delft University of Technology, included experiments with small stand-alone robots. These projects have shown the challenge of using these units on the shop floor, especially with respect to economics. An important step in deciding the application and further development of welding robots is arriving at a sound judgement of their economic potential. To this end, Delft University of Technology and IHC, the world’s leading supplier of dredging ships and dredging equipment, joined forces to establish an elaborate evaluation model of welding robot performance in a realistic business environment.

The paper will present the fundamental structure of the model and will detail how considerations of product parameters as well as yard capacity are reflected in the balancing process that is inherent to properly comparing the conventional situation with a, possibly partly, robotised welding environment. The analysis will show the influence of various robot types on their application potential. The presented techniques and tools allow yards to arrive at the optimum mix of volume, type and functionality of welding robots and will give impetus to robot developers to tailor their products to the situation prevailing at medium-size shipyards.

1. Introduction

The ship production department of Delft University of Technology, faculty of Marine Technology has carried out several research projects on the subject of welding robots in cooperation with industrial partners, including IHC Holland N.V. Academic and practical knowledge in the field of production technology and management was combined to achieve the results that will be described in this paper. The paper explains the set-up of a model for judging the economic feasibility of robots and describes a case study in which the model was used to gather results for the IHC Beaver Dredgers shipyard.

1.1 Why use robots?

Pressure on costs, a shortage of personnel and strict regulations on the working environment regulations are three important reasons to expect an increase in the application of welding robots in North-West European shipbuilding countries. Until now the use of robots has been confined to relatively few applications, but it seems that the technical possibilities finally meet the shipyard’s challenges because:
• the information necessary for future automatic production is more and more defined in the design and engineering phase (for instance: 3-dimensional information on the location of welding seams);
• mainstream robots are getting smaller thus having more access to confined spaces;
• the prices of robots have lowered to the same range as for instance a numerically driven operation centre for milling, turning and drilling; this makes a robot more like other machinery tools.

Considering welding, there are several arguments and strategies in favour of using robots of which the most important are:
• decreased man hours in the light of increasing wages;
• improved working conditions with reduced exposure to welding emissions;
• increased production efficiency (by improving switch-on times and working ratios);
• realization of a more constant quality;
• increased protection of product information in the light of increased in-house production.

Frequently cited disadvantages of robot-welding are:
• teething troubles causing temporarily decrease of production capacity;
• lack of motivation of production and engineering personnel to switch to different working methods;
• less flexibility in production capacity and possibly also production order;
• fear for lost competence in welding technologies;
• financial risks resulting from investment costs.

1.2 Economic feasibility of robots

1.2.1 The influence of robot size

Former research and experiments have lead to the conclusion that the economic feasibility of welding robots was depending on the steel throughput of the yard. Big, inflexible, expensive installations could only be paid back by welding many, easily accessible seams. The high costs were partly due to the cumbersome method of on-line teach programming. These drawbacks were expected to be overcome if robots became cheaper or required fewer investments in adaptations of the production facilities. ‘Smarter’ robots could be able to handle diversity: A robot with ‘vision’ could be an answer to complicated and less repetitive constructions. Therefore systematic research for small and medium shipyards was initiated by conducting experiments with a portable, small and relatively cheap robot. The experiments were meant to be continued to solve some fundamental problems with the programming of the robot and positioning techniques (by sensors). However, the experiments had focused on a portable robot. This means a maximum weight of circa 25 kg, in order to meet the Occupational Health and Safety Act demands. These robots were not available on the market, which prevented the original research from being finished.

1.2.2 Using off-the-shelf robots

Holding on to the portability requirement means choosing for a long development path for specific shipbuilding portable robots. An alternative could be to adapt an available robot, keeping in mind that robots, robot sensing and robot programming have distinctly improved since the last tests in 1998, and giving up the demand of portability. Shipyards have become even more cautious with large investments in production facilities in the Netherlands (or other west-European countries). The physical construction of hulls is more and more moved to Eastern Europe. This makes a trustworthy economic analysis even more important when suggesting new research efforts in this field. If an existing robot (technically supported by a robot vendor) could be proven to yield savings in time and money, then tests and experiments in welding and programming might be picked up with more enthusiasm.
1.3 Remarks

Although not too many robots have been customized for medium-size European yards, in other parts of industry (large-size shipyards in Japan and Denmark, automotive etc) developments in programming and configuration have been made. Research groups all over the world have improved sensor techniques and developed new tools. This means there are many different robots, configurations and program tools to choose from. This raises the question of how to select swiftly the most economical solutions from these existing welding robots. Or phrasing the question the other way round: what are the minimum demands for welding robots to be suitable for a specific Dutch shipyard? This calls for an evaluation of aspects such as robot welding speed, working ratio, accessibility of seams, maximum programming costs and maximum investment costs for a given payback time. To what extent can costs and benefits of a robot be predicted without implementing or even testing it?

Using values that cannot be validated until the robot has been bought implies uncertainty regarding this prediction. (And this will always be the case, whether evaluating only one robot or a range of robots.) Therefore the economic model in this research concentrates on the hypothetical first-order effects, and leaves the secondary effects primarily out of view. This still means uncertainty regarding the aspects mentioned above, but other, even more uncertain, effect as repair costs, labour conditions etc. are not taken into account for the moment. The evaluation method will be explained in section 3.1.1.

The ‘IHC Beaver Dredger’ yard builds sophisticated, standardized as well as customized dredging equipment. The main product range consists of standard stationary cutter suction dredgers. Centrifugal pumps transport the dredged soil as a fluid mixture (slurry) through a pipeline to the dumping site, or discharge into barges. The main advantage of this type of vessel is its ability to operate in variable water depths and to dredge a wide range of materials efficiently. To date IHC Holland has supplied over 600 of standard cutter suction dredgers. Two ranges of standardized designs are available, the “Classic” Beaver and the New Generation range. This research focused on the New Generation range that comprises six models with delivery lines ranging in diameter from 450 mm up to and including 750 mm and maximum standard dredging depths from 10 to 18 m. Beavers are produced both by order and for stock. In short the yard is an example of building more or less repetitive series of ships with a relative simple construction. The yard therefore considers itself as a suitable candidate for the new welding approach reported in this study and so IHC Beaver Dredgers has been willing to provide specific product and production data for further analysis and prototype modelling. Where necessary the model has been adapted to their specific situation.

2. Striking results of evaluating welding costs

2.1 The effect of volume of steel throughput

Application of the model to the Beaver yard situation, has led to the conclusion that also with low steel throughput a robot can be feasible. For this particular yard robot configurations were found to be cheaper compared to the reference situation of manual welding at the yard. Despite of the fact that the robots required substantial investments it proved to be possible to cut down costs with 20%. However, relocating the production to low-wage countries offers still cheaper alternatives than the robot. If relocation would be considered to say Romania, with its still considerable lower wage rate the costs would amount to 75% of the reference situation, Fig.1. This means that more outsourcing will in the short time be cheaper (even more lowering the steel throughput at the Dutch locations). But in long term a “less outsourcing strategy” may be more desirable; to prevent product knowledge and information leaks to sub-contractors. Also the expected raises of wages in the Eastern block would make the robot more attractive.
Also it was established that at the very moment that robots prove to be able to weld a certain percentage of the construction details, or if the design is more adapted to robot welding, the robot becomes a better alternative (65% of original costs, Fig.1). Also secondary effects as decrease of repair costs (not yet included in these figures) would tip the balance in favour of a robot.

Fig. 1: presentation of welding costs in relation with construction

2.2 The effect of complexity of shape

In opposition to the general thought that robots are particularly suitable in simple constructions, robots actually become more attractive when one the one hand the robot can weld more complex constructions, while on the other hand the complexity of the construction increases. This remarkable last part of the conclusion is a logical result of the observation that the costs per unit length of manual welding increase with the complexity of the seam (complex form, difficulty of access, vertical orientation or overhead position). This is because the work ratio and welding speed slow down considerably (and material costs increase). The costs per unit length of robotic welding only slightly increase when dealing with the more complex constructions, as the melting speed of the material slows down only marginally for the more complex (e.g. vertical) seams. The working ratio of the robot moreover stays the same.

Of course this requires the robot to be capable of welding a substantial part of the more complex details of the construction. For a Beaver (the topic of the test case) 85% of total seam-length covered by the robot resulted in welding costs ranging between 95-85% of the costs for the situation without robots. Covering a higher percentage of seam length\(^1\) would result in 80% of the costs, i.e. in 20% cost reduction, Fig.2. This is a realistic value for modern robots used for comparable constructions.

2.3 The effect of flexible deployment of robot and welders

To realize the highest savings in welding costs, the robot must be viewed as an extra tool for the welder to choose from, instead of a separate production unit. This means that even for a part of the construction that is fit for robotic welding, manual welding remains possible. For example, it might be decided to weld the stiffeners of four of the six sections by robot, and the other two by hand. Both flexibility and feasibility will increase as fewer robots are necessary to limit the throughput time, while the remaining welders spend less time waiting for the robot to finish the work allocated.

Combining a robot in a situation with normal complexity of seams (85% of the manual welding costs as explained in the previous section) with ‘normal’ welders improves the performance: the costs come down to 80%. If the robot covers more seam length (Fig. (right)) this becomes even 65%, Fig.3.

\(^1\) seam-length is defined as length in [m] multiplied with the amount of layers to weld
Fig. 2: Presentation of welding costs with improved robot capabilities

Fig. 3: Presentation of welding costs with respect to deployment

Fig. 4: Reducing welding costs with new construction elements
2.4 The effect of constructional design

To achieve major improvements in welding costs the construction should be adjusted for accessibility (for robots and human welders), and to reduce the number of parts and the length of vertical seams. For instance using panels with integrated stiffeners results in the case of the Beaver yard in an additional welding cost decrease of 20% compared to the robot solution without changing the construction, Fig.4.

2.5 Other results

The calculations showed that the production capacity could eventually be increased (while still cutting costs on the welding). Also using more specialized robot set-ups, for instance for double corner welds or welding gusset plates from both sides, would be very helpful in improving the results. Obviously any additional investment cost must be regarded with great care.

3 Overview and highlights of the economical model

3.1 Overview of the method

3.1.1 Definitions and basic assumptions

The economical model is aimed at comparing the costs of manual or automatic welding as it is done now, with corresponding costs of welding by means of robots. The results should support the choice of a suitable robot by giving the conditions for which robotic welding is competitive with the welding techniques and processes used now (manual, Under Powder and MIG/MAG).

The welding costs are defined as the hourly rate of the welding multiplied by the necessary hours for welding. These hours include everything related to welding: preparation, transport, breaks, changing wires, cleaning etc. Primary effects of robotic welding are changing hour rates by changing the surcharges for tools, and changing the welding speeds and working ratios. Changing the necessary welding hours by changing the welding speeds and working ratios, and changing the amount of resources used, are also primary effects.

Secondary effects such as changing the welding costs by changing material costs\(^2\), changing the repair costs\(^3\), and protecting information from competitors cannot be taken into account without actual data, resulting from using the robot. Therefore the economical model focuses on predicting the primary effects of deploying robots.

3.1.2 Material and data on welding seams and capacities

The input data being used to test the model are the welding seams of a section of a serialized, standardized cutter suction dredger. An example of a Beaver can be seen in figure 5. It can immediately be seen that the hull form is very simple. All seams of the left, mid- and right pontoon have been inventoried. In the front view it is visible that the middle unit on deck level contains the engine room and the associated foundations for engines and auxiliary equipment, and is therefore more complicated than the side pontoons.

\(^2\) A steady hand uses less melting material as it keeps the seams as small as possible and does not diverge from the track.

\(^3\) After welding seams are submitted to a quality check. Sometimes ‘repair’ of the seam is necessary, which means removing the old seam and welding it again. It is expected that robotic welding may prove to be of more constant quality, which would reduce the ‘repair costs’.
The input data for the model regarding the capacities and efficiency of the welders at the yard were based on former measurements at the yard. The input data for the model related to the robot capacities were based on former measurements at the yard, data from a robot vendor, former research and assumptions based on the improved sensing and position techniques for robots during the last years.

The choice for testing the model on a Beaver was based on the initial assumption that structures should be relatively simple to justify a robot investment. Please notice, that this has been contradicted by the results of this study, Ch.2.2. The data on robot capabilities and prices given by vendors are used with a safety margin on it. The model has been tested on sensitivity for variations in model parameters input by filling in a systematic range of data on welding speed, working ratio, robot costs, and accessibility of the structure for the robot.

3.1.3 Conditions of production capacity

The case study covered only comparison of welding costs per section while maintaining the production volume at the yard at a constant level. Thus no changes in the amount of work that was outsourced were effected and it was assumed that no additional new building contracts ships can be secured as a consequence of any reductions in cost price.

3.1.4 Special criteria for determining production capacity and costs

The throughput time for a section is used as an indication for the production capacity. Combining this with the condition that the production capacity maintains at a constant level results in the condition that throughput times of sections should remain the same for robotic welding as it was for manual welding. The amount of welding personnel (=welding hours) is used as a measurement for welding costs. This results in an important constraint for this research: the number of necessary welding hours (both operators as well as ‘normal welders’) for a section should diminish to justify the investment while keeping the throughput time the same.

3.1.5 Procedure followed in the model

To compare the primary –computable- effects of using robots with the reference situation of manual and automatic welding, the following procedure is followed:

a. calculate the necessary manual\(^4\) welding time for seams in the section
b. derive the hourly rate for manual welding;
c. calculate the (manual) welding costs;
d. classify and select the welding seams suitable for the available types of robots;

\(^4\) With ‘manual welding’ are meant the actual welding methods at the yard. ‘Manual’ is not a fully covering term, as also semi-automatic machines as tractors are used. For convenience these are for now included in ‘manual’
e. calculate the ranges of necessary welding times when using robot(s) and combinations of robot(s) and manual welders/operators;
f. select the suitable robots or robot/human welder-combinations on the ‘special criteria’, which means the configurations that keep the throughput time the same while deploying less personnel;
g. calculate the ranges of the new hour rates for robot welding;
h. calculate and compare the new (robot-assisted) welding costs with the reference situation of manual welding and with each other.

After a considerable amount of calculations had been made according to this model (about 60 different robot configurations have been compared), the method has been automated. This gave cause to some adaptations. The procedure was changed in steps e and f to read: ‘setting acceptable ranges for welding times’ and ‘finding the optimum combination of robots and manual welders for these welding times’. This will be further explained in Ch.3.2.

3.2 Key Elements of the method

3.2.1 The algorithm for necessary welding times

In both the ‘normal’ as well as the automated procedure the necessary time for welding should be calculated in order to arrive at the welding costs (step e in the procedure of Ch.3.1.5). Subsequently, this welding time has to be translated into throughput time to see if the production capacity remains identical.

The necessary time for welding follows from dividing the welding distance [m] by the welding speed [m/hour] and the ‘working ratio’ of the welder c.q. robot. This reflects that ‘real welding time’ is only a part of the welder’s work time, resulting from all kind of supporting activities as transport time, ignition time, cleaning, corrections, pauses etc.\footnote{To give an idea of the values used: The measured welding length in one of the six Beaver types is about 4.5 km of which 2.5 km is now welded in-house. Welding speeds for manual MIG/MAG welding relate as vertical/horizontal=0.3/1. Welding speeds for robots (MIG/MAG) are estimated to be about 1.2 times the horizontal manual speed. The working ratio for robot welding of vertical seams is about 2.4 times that of manual welding, for horizontal seams this is 1.6 (exact values depending on the robot type)} The welding times per phase are marked by I, II, III, IV, the total time per ship is V, Fig.6.

Fig.6 (lower half) illustrates calculation of the throughput time. To establish the relation between value V (the throughput time per ship) and VI (the throughput time per cluster of ships) is more complicated. If the number of ships built at the same time at the yard exceeds one, then this relation must be known to determine the production capacity.

In the manual calculations, the total throughput time (VI) was obtained by scheduling all activities in the most efficient way. (This total throughput time is necessary for step f). No algorithm was developed for this, but a set of heuristic and experience rules. Efficiency in this context relates to short total throughput time and minimization of waiting times of personnel. Fundamental to this algorithm is the assumption that a resource (human and/or robot) can only work on one section and phase at a time and a section can only be submitted to one action at a time.

As already mentioned the procedure in the model was changed for step e and f when automated, this to by-pass the difficulties of implementing heuristic rules into scripts. The automatic model now asks for a maximum acceptable time per phase, and looks for a fitting robot configuration. The user should decide what total throughput time is acceptable.
A rule of thumb can be used to find the maximum phase times that lead to throughput times that are in the range of the manual welding situation. The lower part of figure 6 shows the parameters X and Y. From experiences with the manual calculations it is known that the most pessimistic scenario for robot welding is: total throughput time (for the described activities) = n*(X+Y). Here n is the size of production cluster in sections: i.e. how many ships are built at the same time. X is the welding and waiting time for phase 1, Y is the welding time for phases 3 and 4. In this case there won’t be any waiting times between phases. It means that the relations between lengths of phases can be used for calculating X and Y: X_1, Y_II+IV.

This means for example with a production cluster of 3 sections and a normal throughput time of seven weeks for this cluster that X+Y=7/3. In a more effective/efficient solution the X and Y periods would overlap, each thus being allowed to take more time. Thus, using X+Y=7/3 is a safe constraint for choosing a start value for the maximum phase time.

3.2.2 Algorithm for optimizing resources for given maximum throughput time.

After the choice of a fitting phase time (step e in the automated model, as explained in the previous section), the automatic economic feasibility model starts with calculating the optimized number of resources (i.e. robots, manual welders or both). For this a maximum number of units in a phase should be defined. Two considerations are of importance for this maximum value: 1) there are practical limitations on the amount of resources that may work at the same time within the same space, 2) the total amount of robot operators and remaining manual welders working on a project should be less than the original amount of welders.

Then the computerized procedure starts:
Per phase: Is the maximum number of robots [R] equal to zero?
  o If yes: the calculations start with the maximum number of manual welders [W].
    - If the resulting phase time is shorter than the defined maximum (the value that was set as start value) the configuration is accepted and the calculation is repeated with the maximum minus one welder [W*=W-1] etc. When a new configuration is accepted, the previous cases are automatically rejected.
    - If the resulting phase time is longer than the defined maximum, the configuration is rejected.
  o If no: the calculations start with the maximum units of robots [R],
    - If the resulting phase time is shorter than the defined maximum, the configuration is accepted and the calculation is repeated with the maximum minus one robot [R*=R-1] etc. When a new configuration is accepted, the previous cases are automatically rejected.
    - If the resulting phase time is longer than the defined maximum, input is asked from the user: “are there practical limitations on splitting the work in this phase?”.
      - If no, the available manual welders in this phase take over that part of the work suited to robots but not completed in time.
        o If the resulting phase time is longer than the defined maximum, the configuration is rejected.
        o If the resulting phase time is shorter than the defined maximum, the configuration is accepted and the calculation is repeated with the maximum minus one welder [W*=W-1] etc. When a new configuration is accepted, the previous cases are automatically rejected.
      - If yes, the configuration is rejected.

This is carried out for each phase in each section in the defined production cluster. The result is an optimization of the amount of resources (for this robot type that is the subject of the evaluation) for the given maximum phase times. The next step would be to include an iteration that sets a new maximum phase time based on the total production capacity.

3.2.3 Comparing costs and payback times

The algorithm in the previous section delivered a configuration that meets the criteria of a suitable solution. Now the costs of welding with this configuration should be compared to the reference situation: “manual” welding. The parameters in the cost calculations are the optimized hours necessary for welding that followed from step e and f and the hourly rate of a welder (with or without use of a robot). An important component of the hourly rate is the wage that is assumed to stay the same irrespective of the use of robots. However when the number of welders drops the overhead costs per welder increase proportionally. The costs for the robot are surcharged on the hourly rate of the welder only during the time he or she operates a robot. (Robot hour rates are based on the estimated time that the robots are used). To evaluate the possibilities of robot welding on the Beaver Dredgers shipyard, five years has been assumed to be a reasonable (maximum) term for the write-off of a robot. The costs of installing robot have been based on an offer of a robot vendor and a conservative estimate is used for the expected costs for programming.

3.2.4 The matrix with values tracked in this study

With no recent experimental data for robot speed, working ratio, and robot functionality available, assumptions were made based on former experiences and (although with some caution) specifications of robot vendors. To test the model’s sensitivity to differences in this set a range of values was used. The same was done for the robot price: much is depending on the costs for programming macros and adapting the robot software to the situation on the yard, which makes it difficult to give a generalised cost indication.
Table I: Range of values used for robot features

<table>
<thead>
<tr>
<th></th>
<th>vendor/own research</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed</td>
<td>75% 100% 125%</td>
</tr>
<tr>
<td>working ratio</td>
<td>45% 60% 75% 90%</td>
</tr>
<tr>
<td>accessibility seams</td>
<td>74% 87% 100%</td>
</tr>
<tr>
<td>price robot</td>
<td>75% 81% 100% 113% 125%</td>
</tr>
</tbody>
</table>

The behaviour of the model was stable for these ranges of values: the results of calculations showed trends that were to be expected given the changes in input. Systematic combination of the ranges has not taken place yet.

4 Discussion of the model and the results for the test case

4.1 A pragmatic approach

This method for quick economic comparison of welding costs has proved its value in both theoretical and practical situations. It allows judging conceptual robots in order to set constraints for useful future developments as well as the possibility to evaluate actual offers of robot sellers. Certainly with the availability of the automated procedure it provides a swift way to consider many options with hardly any exclusion of options beforehand. This wide range of options leads towards superior interpretation of trends, culminating in the selection of the most suitable robots. The choice of conceptual robots strongly depends on the frame of reference of the user of the method. When using the model to set up demands for an optimal robot it is difficult to deviate from available, conventional robots. More creativity on robot possibilities and brainstorming may lead to better results.

4.2 Heuristic and experience rules

The rule of thumb as used in Ch.3.2.1 has only been validated for the IHC Beaver Dredgers yard. Further collection of data from other yards is necessary to extend and check the rules used. The ‘rule’ that less personnel means lower costs has only been validated for small sections. In all the calculated cases the practical limit of manual welders at a section was four. Thus a decrease in personnel immediately resulted in 75% or less of the old staff. With new data on larger sections, the rule could be checked for values between 100 and 75%. The coupling of the throughput time to production capacity is occasionally not only confusing but also false. For instance when work is outsourced, the sections are built in stock or the total production capacity is set by for instance the painting capacity. But for a yard that is bound to construction at a single slip yard the coupling probably will prove true.

4.3 Sensitivity range

The sensitivity range for which the behaviour of the model has been observed was given in Table I. The automatic model improves the possibilities to investigate new data to enlarge this set. The size of the production cluster (i.e. the number of ships built at the same time at the yard) proved to have no influence for the range tested in this case.
5 Conclusions

Robot performance can be compared with manual welding performance, without knowing all details on such robots. Using conventional, commercial-off-the-shelf robots for welding of profiles, plates and finishing seams in a Beaver could be an attractive alternative for IHC. Necessary preconditions are that programming costs keep within the estimation, the repair costs of robot welding are lower than for manual welding, the robot proves to be able to weld gusset plates, and the amount of work outsourced to others does not increase. Although this research shows that simple constructions can be welded economically by robots, the real profit must be made on the more complex parts of these constructions provided the robot is capable of reaching those seams. This means that, in opposition to the prevailing assumption that robots should be used for high steel volumes and simple structures, also shipyards with a relatively low in-house steel throughput (as Beaver Dredgers, but also other Dutch yards) can profit from using robots. Calculations show that starting ‘in a small way’ (initial investment of only one or two robots) and flexible deployment of this robot(s) in only a part of the suitable constructions lead to the best results. The shipyard has therefore started experiments with automatic programming of welding seams out of CAD drawings and quality measurements, in order to see if the preconditions for profitable implementation can be met. Delft University of Technology will continue its research on the possibilities of robot welding in shipbuilding. For this feasibility study the next step will be extending the automatic model: to include an iteration that sets a new maximum phase time based on the total production capacity. Combination of these efforts will lead to the desired result of decreasing man hours, increasing production efficiency, constant quality and improving production capacity.

References


Simulation Models in Marine Engineering: From Training to Concept Exploration

Hugo T. Grimmelius, Delft Univ. of Technology, Delft/Netherlands, h.t.grimmelius@wbmt.tudelft.nl

Abstract

In marine engineering, simulation models can be used for a wide variety of purposes. This paper will highlight four applications: concept exploration studies, system integration and optimisation, intelligent condition monitoring, and simulators for education and training. This paper shows that, with a good selection of modelling techniques and implementation, modelling concepts can be used for several of the above mentioned applications. The selection should be based on a proper analysis of the problem at hand. Numerous techniques for building simulation models are available, ranging from detailed component models for design optimisation, to observer like black box models for intelligent plant control. Examples are given of modelling of diesel engines used for concept exploration studies and used for ship propulsion plant analysis and optimisation. Further a comprehensive compressor refrigeration plant model is discussed. This model is used for analysis and prediction of faulty plant behaviour and the design of intelligent condition monitoring systems. Finally the application of models in engine room simulators is discussed, using the ships electric net as an example. The difference between the use of simulators for education and for training purposes is explained. This effects the choice of system and component models. The challenge is to achieve a more effective use of computer simulation and a more extensive use of simulation techniques in all stages of the design and operation of systems.

1 Introduction to modelling

In many fields, simulation models have become established tools during the design process. In marine engineering however, the application stays mainly limited to research project. Unfortunately, no uniform approach to building the required models exists. In fact, such uniform approach would be difficult to envisage, given the many different applications and the many different settings simulation models are used in.

When creating a mathematical model, the engineer faces a number of problems. The first challenge (from a technical point of view) is to define where the own expertise ends and where other experts have to be called in. Compatibility of the fields of expertise of the different experts then becomes an issue. A general trend in marine engineering is the growing complexity and scope of systems. Aided by increasing computer capacity (both software and hardware) the focus more and more lays on a total system rather than on individual components. The result is that the modelling of such a total system necessarily is done by a team of experts, each creating its own sub-model. The engineer linking the sub-models is offered a number of black boxes that comply with set interface protocols. From a modelling point of view, aspects as time scale, input and output etc. are of importance. From a physical point of view however not only the interfaces are of interest, but to some extent also the content of the black boxes. One would like to know something about the physical background of the model in order to be able to assess the applicability of the model. In fact, not only modelling guarantees but also physical guarantees have to be given. The black box should be grey. This is discussed in some detail in Schulten et al. (2003).

To limit the scope in this paper, only models that are used to represent the behaviour of a physical system are considered. But even then, a great variety of techniques exist. In this paper, modelling of physical systems is categorized in four dimensions: one indicating the level of detail of the description of the actual processes (model level: L), another indicating the time domain for which the model is developed (model time domain: MT), the third indicating the time domain in which the model is to be used (application time domain: AT) and the forth the amount and character of data required to get
useful results from the model (model data: MD).

It must be emphasised that this paper in no way intends to discuss the whole field of simulation, or to present a comprehensive literature overview. Its intention is to give the reader some examples the author has been working with, to put these examples in a broader perspective, and in this way provide a starting point for discussion and further study.

1.1 Modelling levels

Simulation models as discussed here are basically mathematical functions, suitable for automated calculation, that calculate outputs or states, given inputs, initial conditions and parameters. The required mathematical functions can be derived in several ways, and their representation depends on the hard- and software used. A rather general subdivision is possible, however, based on the applied level of physical process knowledge during development and implementation of the model \((L)\):

- \(L1\): black box models: describe purely the input-output relations, using for instance transfer functions, regression modelling or neural networks. The required process knowledge is very limited, measured data is in most cases sufficient to create an accurate and useful result. However, often expert knowledge on signal processing and statistics are required.
- \(L2\): Grey box models: a combination of black and white box models, for instance a black box model for the process, based purely on measured data, but a white box model for the controller and actuator.
- \(L3\) white box models: based fully on physical first principles. All functions are derived from physics, resulting in - sometimes complex - differential equations, representing most often laws of conservation such as the first law of thermodynamics or conservation of mass. Often, semi-empirical equations cannot be avoided, for instance for heat transfer coefficients or properties of matter.

1.2 Modelling time domains

Simulation models are used to obtain knowledge about the real system behaviour. The time domain of the behaviour that is to be studied also dictates to a certain extend the form and contents of the model and the modelling time domain \((MT)\):

- \(MT1\): Steady state behaviour. This means that no dynamics are required. Such models need only a numerical solution protocol that will render a valid working point, to check the system behaviour for various stationary operating conditions. Useful for, for instance, concept exploration, feasibility studies and initial configuration determination.
- \(MT2\): Time domain behaviour. Here system dynamics play an important role. This however opens a very wide area, and one should always question what part of the dynamic behaviour is important. For instance, if one is interested in the manoeuvring capabilities of a ship, the torsional vibrations in the shaft are of no interest, and can be omitted, though a time domain simulation model is built. Vice versa is also often true: for fast dynamic phenomena, very slow processes can be considered in steady state, and the associated dynamics can be omitted.
- \(MT3\): Frequency domain behaviour. This is a special representation of time domain behaviour, but has developed its own 'language', and is for many applications more efficient. Especially for stability checks and controller design, stochastic processes, vibration and sound problems the frequency domain offers many possibilities.
- \(MT4\): Additional domains, such as the ensemble domain, using measured data points from a finite time windows for evaluation.

Note: Clearly, depending on the goal of the simulation, the most suitable domain must be chosen, preferably before any measurements are carried out.

1.3 Application time domains

The application time \((AT)\) domain of the simulation model also influences the modelling itself:
- **AT1**: Off-line - not real time – use. The required calculation effort is of minor importance. Computers are ever becoming faster and more readily available to everybody. Of course, sometimes practical limitations have to be kept in mind, but in general calculation time is not anymore - an important limit.
- **AT2**: For on-line - real time - use, the consequences are much more dominant. The use of variable step size integration algorithms, which are often preferred in off-line simulations, becomes dangerous since the required calculation time is not exactly known in advance. Especially during transients, the required calculation time may exceed the available timeslot, resulting in a run time error. So, care must be taken to ensure that the required results from each time step are available in time.

*Note*: The definition of what 'real time' is, though it may seem trivial, depends strongly on the application and should be considered carefully in advance!

### 1.4 Model data requirements

One of the often neglected aspects of simulation model building is the data required for the model (MD). Here four categories are distinguished:
- **MD1**: Models requiring limited measurements, such as simple transfer function models.
- **MD2**: Models requiring limited system parameters, such as first principle models for concept exploration.
- **MD3**: Models requiring only or mainly measurements, such as statistical models.
- **MD4**: Models requiring only or mainly extensive system parameters, such as detailed first principle models

### 2 Diesel engine model examples

#### 2.1 Seiliger cylinder process model

The main part of the diesel engine model is the description of the processes in the cylinder. The cylinder model consists of two parts, the bottom cycle, or gas exchange model, and the top cycle, or closed cylinder model. The gas exchange model calculates the trapped conditions - the pressure and the temperature at the start of compression and the scavenge efficiency - and calculates the mass flows, the amount of work required or performed and the blow down process. It incorporates the heat pick up by the scavenging air in the inlet channels and the cylinder and the outlet channels.

The top cycle is modelled using the (6-point) Seiliger process, Fig.1, with the heat losses to the cylinder wall integrated. The 6-point-Seiliger process has six predefined points and five predefined processes:

1→2: Compression stroke (polytropic process)
2→3: Constant volume (or premixed) combustion
3→4: Constant pressure combustion (and expansion)
4→5: Constant temperature combustion (and expansion)
5→6: Expansion stroke (polytropic process)

Once the trapped conditions are determined in the gas exchange model, and the engine speed and fuel consumption are known, the Seiliger diagram is calculated. This gives the mean effective pressure and the condition in the cylinder when the exhaust valves open (point 6).
The advantage of this thermodynamic approximation of the combustion cycle is that internal and external cylinder conditions (e.g. $p_{\text{max}}$, $T_{\text{max}}$, $p_0$ and $T_0$) are calculated, with reasonable accuracy, without the need for a detailed model of the cylinder. The matching of the model becomes much simpler and the number of required engine parameters is reduced.

In the following sections the use of the Seiliger cylinder model in different studies is discussed.

2.2 Dynamic system design and control evaluation

A modular modelling concept was chosen for the dynamic engine model, based on the bondgraph method, *Karnopp (1999)*. Fig.2 shows the block diagram of the dynamic diesel engine model with the internal connections between the various components.

![Fig.2: Block diagram of turbocharged diesel engine](image)

The turbocharger components, the compressor and the turbine, are modelled as resistance elements. The turbocharger speed is determined from the energy balance across the turbocharger shaft. Other resistance elements are the bypass valve and the charge air cooler and the cylinder, which is a very complex resistance element. The inlet receiver and the exhaust receiver are volumes, in which the mass balance and the energy balance are used to calculate the – ideal mixed – conditions. Heat transfer to the wall and to the ambient is also included.

The engine model itself is incorporated in a ship propulsion model, as shown in Fig.3. At the right hand side the force balance between ship resistance and propeller thrust provides the ship speed. On the left hand side in the torque balance between engine output torque and propeller torque provides the shaft speed.

The ship speed, after correction for the wake factor, provides the entrance velocity of the propeller. With the shaft speed this results in the effective propeller blade angle of attack. Propeller thrust coefficient and torque coefficient are a function of this angle of attack, the propeller pitch and the shaft speed. The propeller pitch is set by a pitch control and actuating system (often hydraulic) that gets its set point from the ship’s propulsion control system. The propulsion control system also provides a set point to the prime mover control block that normally contains the engine governor. For diesel engines the governor actuates the fuel rack.

The two main disturbances will act on the system through the ship side. The first important disturbance is caused by the added ship resistance in case of for instance heavy seas. The second disturbance works on the entrance velocity and angle of the propeller and is caused by the induced ship movements and orbital water velocities in heavy seas. In Fig.3 this disturbance is introduced through the wake factor.
2.2.1 Control of sequential turbo charged engine (L3, MT2, AT1, MD4)

The dynamic simulation model was used to compare the performances of a conventional engine to that of a sequential turbocharged engine. The sequential turbocharged engine has two or more turbochargers of which one or more chargers may be shut down by means of closing or opening valves. The advantage of this charging concept is an increased performance of the turbo chargers under part load, resulting in a higher power output at lower engine speeds.

In Fig.4, the simulation results are plotted for a propulsion system with a ST engine and with a conventional engine. At time is 100s the acceleration of the shaft speed is initiated. The plots of the ST engine show the switching procedure: the charge air pressure ($p_{ch}$) initially increases very fast, due to the relatively small turbocharger, running at high efficiency. Then the valve for the second turbine opens, the acceleration of the turbocharger causes a drop in charge air pressure because less mass flow is available for the turbine of the first charger. As soon as the second turbocharger is speeded up, the charge air pressure is restored.

It is clear that across the entire range the ST engine has a higher charge air pressure ($p_{ch}$) and therefore can deliver a higher torque ($T_{me}$). In fact, charging is not the limiting factor, but the maximum cylinder pressure, trough the torque limiter in the governor, determines the maximum limit of the fuel injection. The air fuel ratio ($\lambda$) is at any moment better in the case of the ST engine and therefore this engine has a reduced thermal loading. This is also visible in the plot of the maximum temperature ($T_i$). For both engines, the fuel rack position initially follows the rate limiter, and it is clear from the air fuel ratio that the limiters are set correctly. On both engines the air fuel ratio remains just above 1.5. The simulations show that the engine ($N_{eng}$) and the ship ($v_{sh}$) accelerate much faster with the ST engine configuration compared to the conventional engine configuration. During acceleration, the ship with the ST engine has travelled approximately 150 m more, on an overall travelled distance of 1500 m.
2.2.2 Evaluation of propulsion plant control regimes (L3, MT2, AT1, MD2)

A generalized version of the model used in the previous section is very suitable for propulsion control research. Many overload problems for the propulsion plant of a ship originate from the governor trying to keep the engine speed constant. Other control strategies are possible, in particular in conjunction with the controllable pitch propellers. Electronic control now being state-of-the-art both for engine and pitch control pave the way for a more integrated approach of the two. Several control strategies were tested on the model, including a multiple input, multiple output controller for disturbance rejection.

The commonly applied control strategy to diesel engine propulsion plants, Fig.5, is to control the engine speed, using the governor to set the fuel rack. However, the governor, in trying to keep the speed constant when load increases – induced for instance by waves – will force the operating point of the engine closer to, or even across, the engine overload line. A possible alternative is the use a Multiple Input, Multiple Output (MIMO) controller, Fig.6. As in conventional control the Propulsion Control System generates a demanded speed and pitch angle. Both are compared to actual values and put into the MIMO controller that produces the setpoints for both the fuel rack and pitch angle servo system.

Fig.7 shows the behaviour of the propulsion plant and the ship under the two different control regimes. The two top graphs show the two different controlling variables: fuel rack and pitch. The different control actions of the two controllers counteracting the wave induced disturbances are readily recognisable. From the engine speed graph it is clear that the MIMO controller does a better job in maintaining a constant engine speed. The main reason for this is that in order to maintain a constant engine speed, the fuel rack controller has to adjust the power delivered by the diesel engine through adjusting the fuel flow. The response of the turbo-charging system however is not very fast.
Fig. 5: Conventional fuel rack control of shaft speed

Fig. 5: Multiple Input - Multiple Output (MIMO) control of shaft speed

Fig. 7: Time simulations showing fuel rack, pitch, engine speed and ship speed in sea state 6, for both fuel rack and pitch control regime. On time = 200 s the requested speed in increased from 80% to 100% of the rated engine speed, Grimmeius and Stapersma (2001)
The MIMO controller on the other hand has a second possibility: changing the pitch instantaneously changes the torque, and in this case, the changing of the pitch is very fast. The impact on the behaviour of the ship as a whole is limited, as could be expected, shown through the ship speed in Fig.7. Any disturbance generated in the propulsion plant has to pass through the ship inertia before showing up in the ship speed, effectively filtering out all higher frequency components. Only during the transient the MIMO control renders a slightly better stabilisation after acceleration, mainly because the fuel rack controller is disturbed, through the charge pressure limiter by the dynamics of the turbocharger.

2.3 Concept exploration

To use the Seiliger diagram for a broad concept exploration of the diesel process was inspired by Paro et al. (1993). For this application the Seiliger model is completed with the power balance over the turbocharger, additional models for mechanical and heat input efficiency and the same gas exchange model as used in the dynamic engine model. With the exhaust receiver pressure as input, the turbocharger power is balanced in an iterative loop.

2.3.1 Power and efficiency limits for Diesel engines (L3, MT1, AT1, MD2)

As an example of the possibilities this approach offers, the trade-off between maximum pressure and charge air pressure will be discussed. Fig.8 is constructed using constant values for constant pressure combustion ratio, constant temperature combustion ratio and air fuel ratio \((a, c \text{ and } \lambda)\). Also charge air cooler efficiency and cooling medium temperature are kept constant.

Fig.8: Trade off between efficiency and mean effective pressure. Grid of charge pressure, peak pressure, compression ratio and scavenge pressure, Stapersma and Grimmelius (2001)
To illustrate the development of the diesel engine over the last 30 years, two state-of-the-art reference points are included: for the year 1969 and the year 1999:

<table>
<thead>
<tr>
<th>Year</th>
<th>Brake Mean Effective Pressure $b_mep$</th>
<th>Overall Efficiency $\eta_{tot}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1969</td>
<td>14 bar (4-stroke), 10 bar (2-stroke)</td>
<td>40%</td>
</tr>
<tr>
<td>1999</td>
<td>24 bar (4-stroke), 18 bar (2-stroke)</td>
<td>48% (4-stroke), 50% (2-stroke)</td>
</tr>
</tbody>
</table>

This increase in performance, both in fuel economy and power density, has come about by a simultaneous increase in charge air pressure and maximum pressure, but other parameters are changed as well, therefore in particular the 1969 cases do not completely correspond to the curved grid.

Apart from the constant charge pressure and constant peak pressure lines, also a set of lines of constant compression ratio is included. Yet another set of lines are drawn for constant scavenge pressure ratio across the engine, which is particularly important for the 2-stroke engine.

Starting from the 1999 nominal point, a vertical direction (where power density remains constant and efficiency goes up) almost coincides with an increase in peak pressure at constant charge pressure. Pretty soon the compression ratio becomes unrealistically high while scavenge pressure across the engine decreases. A more realistic "route in the map" would be along a line of constant compression ratio: then charge and peak pressure increase simultaneously. For the 4-stroke engine in Fig.8 would suggest that a $b_mep$ of 50 bar at 50% would be possible at 8 bar charge pressure and 350 bar peak pressure. This would require 2-stage turbo-charging. For the 2-stroke engine a constant (effective) compression ratio is also a good strategy: Fig.8 suggests that a $b_mep$ of 30 bar is attainable for 6 bar charge pressure. With 300 bar peak pressure it would have an efficiency of 52%, i.e. not much higher than today’s engines. The scavenge pressure for the 2-stroke engine of course is a hard limit. In its present form the scavenge pressure is generated by the turbocharger and than the limits are in sight: the peak pressure of 340 bar almost coincides with a scavenge pressure ratio of 1.1. Of course for both engine types the compression ratio could be increased, thereby trading off an increase of power density against an increase of fuel economy.

2.4 Applications in real time

2.4.1 Engine room simulator ($L3, MT2, AT2, MD4$)

The dynamic simulation model, mentioned earlier, is very suitable for application in an engine room simulator, such as used in training facilities all over the world. Because the model is based on first principles, failures may be simulated very realistically, without extensive preparations or adjustments. For example, simulating fouling of the charge air cooler is done by changing the heat transfer coefficient and/or effective area of the cooler, just as in practice. The resulting effects and consequences are automatically taken into account in the other component models and thus, the impact on the performance of the diesel engine. As a result, the instructor has a very large freedom in choosing the faults he wants to use for his particular exercises.

To illustrate the possibilities, Fig.9 shows an approach for the development of a simulator software, starting with systems drawings and accompanying descriptions and a database of component models. The component database contains parameters, inputs and outputs, and a library of already built component models.
Fig. 9: Implementing general modelling software in developing a simulator, Grimmeius (2000)

The diesel engine model is just one component of the engine room simulator, but is implemented just like any other component model. A main condition for this simulator development approach is that all the models are build up in the same way: based on first principles, fully parameterised and modular. Otherwise, the component models can not be standardized and stored in a component database. Although other engine room will have one or more components that are not conventional, or that have some extra features, most components are of the same type. The advantages of this approach combined with the earlier mentioned modelling philosophy are a reduction of the development time of new simulators and more effective and realistic simulator training facilities.

2.4.2 Diesel engine model in electric net simulator (L3, MT2, AT2, MD2)

At the Nautical College “Willem Barentsz” several ship electric net simulators are available. These simulators consist of the actual switch boards, including breakers, voltage controllers and load sharing facilities, connected to scaled down loads and fed through scaled down generators driven by electric motors. The electric motors act as prime mover (e.g. a diesel engine) for the system. To get a more realistic behaviour of the simulator, the control of the electric motor is controlled through a simulation model of the diesel engine with governor and rotational dynamics, running on a PC, Fig.10. The actual speed of the engine and the delivered power are measured and used to calculate the torque of the load. The simulation model of the diesel engine and governor provide the delivered torque of the engine based on the actual speed and the set-point. Through the rotational dynamics model a new speed for the electric motor is determined. The settling time of the electric motor speed control combined with the rotational dynamics of the motor-generator set is an order of magnitude smaller then that of the simulation model and can therefore be neglected.
3 Condition monitoring examples

Fig. 3 gives a generalised overview of a plant with intelligent condition monitoring. The measured input signals to the plant are also used to generate reference values. These are compared with validated sensor readings from the plant’s output, giving the residuals. Symptoms are derived in the health monitoring part. After the decision ‘healthy or not’, diagnosis is attempted using the knowledge database, which has been filled using different knowledge sources.

Fig. 11: Terminology and structure for intelligent condition monitoring systems, Grimmelius (2002)

Simulation models can play two roles here:

- Within an intelligent condition monitoring system, some reference values are required, to compare the actual behaviour of the plant with. Only limit values can be used, not taking into account changing operating conditions. The use of simulation models provide more accurate reference values, by incorporating the influence of the operating conditions, and thus make it possible to detect failures more quickly and more accurately. Section 3.1.1 gives an example.
- To generate the knowledge needed to diagnose a fault, simulation models may also contribute. The use of models for this purpose would enable the development of diagnostic routines also for custom made machinery configurations, which is up to now far too time consuming and expensive. Section 3.1.2 gives an example.
3.1 Compressor refrigeration plant

On board ships, refrigeration plants are used for domestic uses (food stores, air conditioning), cargo conditioning and to fulfil special cooling requirements (electronics and weapon systems on naval vessels). The plants used for cargo conditioning are complex and sophisticated, while this cargo is often expensive and very susceptible to temperature changes. The plants used on naval vessels are often relatively uncomplicated, but the cooling they deliver is a prerequisite for operation of the vessel. Failures in both cases therefore are to be detected promptly and correctly to enable timely corrective action by the crew. Monitoring these plants has proven to be a difficult task for the ship's crew, since often expert knowledge is required to assess the plants working condition. For these reasons, ship owners have requested further research into intelligent monitoring systems for refrigeration plants.

3.1.1 On line healthy behaviour (L1, MT1, AT2, MD3)

A black-box healthy plant model was developed, based on a linear regression, using over eight thousand measured sets of data. The regression models for the variables contain a maximum of nine coefficients. The models were based on physical considerations and a regression model for the saturation line of R22 from literature. Estimates for the coefficients were calculated using a multi-variable least-squares method.

![Residuals of suction pressure](image)

Fig.12: Residuals for compressor suction pressure, vs time and vs calculated value, Grimmeius et al. (1995)

Plots of residuals (differences between estimated and measured values) are presented in Fig.12. The residuals are shown against the estimated value and against time. As can be seen in these plots, the residuals are small. Relatively large residuals occur during capacity steps (this is accomplished by changing of the number of cylinders in operation). This could be expected since the regression model incorporates no history terms, so dynamic effects are described poorly.

3.1.2 Faulty behaviour prediction (L3, MT2, AT1, MD4)

In a preliminary study it was found that obtaining knowledge about the behaviour of a failing compression refrigeration plant was difficult and also strongly depended on the layout and component choice of a specific plant. To avoid expensive measurements while still enabling detailed early warning diagnostic capabilities an attempt is being made to mathematically model the compression refrigeration plant in such a way and with such detail that simulation of the - dynamic - behaviour, after introduction of a fault, is possible.
Fig. 13: Example of measured symptom determination: deviation of the chilled water temperature for an increased cooling water flow.

Fig. 14: Example of simulated symptom determination: deviation of the chilled water temperature for an increased cooling water flow.

Using a laboratory chilled water test plant, for a limited series of faults the faulty behaviour is measured to be able to evaluate the simulation results. These measurements are repeated at several different operating conditions to determine possible influences of the operating conditions on the faulty behaviour. The same faults are simulated with the model, under the same operating conditions. Two combinations of measured and simulated behaviour are used for the evaluation of the results:

1. Measured faulty behaviour versus simulated healthy behaviour, which renders the measured symptoms of the faults, Fig. 13.
2. Simulated healthy behaviour versus simulated faulty behaviour, which renders the simulated symptoms of the faults, Fig. 14.

This is discussed in more detail in Grimmelius (2000).

4 Discussion and conclusion

The classifications used here can provide a first indication and be instrumental in the communication between various experts. There are however many more and more detailed classifications possible, taking into account e.g. the number of states in a model, and/or the dominant time constants. If a model provided by an industrial partner (e.g. a diesel motor model) is considered, the different attitudes towards modelling between a researcher and an industrial designer can be illustrated: From a researchers point of view, transparency of a model is very important. For industrial designers
accuracy and reliability are more important. Thus, a very detailed model provided by industry is of limited use to a researcher if he has no access to the model itself (black-box). On the other hand, to an industrial designer, if the quality of the model is assured, the model will be very useful. Clearly, the envisioned use of the model and the goal of the user with the results is a often underestimated factor in model development and in the evaluation of model suitability. From a researchers point of view, the amount of parameters required to compose a model should be limited, especially if these parameters are hard to obtain. On the other hand, for the industrial designer, the parameters are provided with (or rather included in) the model, and of no concern to the user.

The number of parameters or other data required for a model is also crucial in the concept design stage: the user does not have all the parameters yet, and still wants to be able to use the model to explore possibilities. For an example of an approach to tackle this problem, see *Dirix et al. (2003).*

It is my conviction that simulation models are an important tool for the marine engineer, and that they should be used more often, and more diversely. The examples shown here illustrate a number of possibilities, but there are many more. However, both in research and in design applications, care must be taken with the interpretation of results. Just like so called ‘expert systems’, simulation models are not a replacement for an experienced and well-trained human expert, but they are a valuable aid in an increasingly complex technical world.

**References**


GRIMMELIUS, H.T. (2000). *The use of first principle modelling for faulty behaviour prediction in the design stage,* 1st COMPIT, Potsdam


Underwater Climbing Robot for Ship Cleaning

Manuel Armada, Pablo Gonzalez de Santos, Teodor Akinfiev, Roemi Fernández
Instituto de Automatica Industrial – CSIC, Madrid/Spain, armada@iai.csic.es

Abstract

The AURORA project is introduced and a solution is proposed to improve the performance of an underwater climbing robot designed for cleaning and surveying tasks on ship hulls. Nevertheless, this solution can be used for any wheeled mobile robot that moves upon curved surfaces, surfaces with irregularities and small coefficient of static friction between the wheels and the surface. The solution implies that the wheeled mobile robot has as many traction motors as wheels controlled independently. It also implies the utilization of a mixed velocity-force control algorithm. In this way, an increase in traction force is achieved and the slipping of the wheels during the movement process is prevented. The effectiveness of the new algorithm is demonstrated in laboratory experiments using a prototype.

1. Introduction

It is well known that all kind of ship’s underwater hull become overgrown with sea adherence (weed, barnacles) very fast. This means raise of fuel consumption, and freeing atmosphere an extra amount of CO₂ (incrementing greenhouse effect) and of sulphur dioxide (acid rain), apart from deterioration of ability of ship’s control. This situation becomes important even after six months of ship activity. For recovery of ship’s required operational performance, it is necessary for Ship-Repairing and Conversion Industries to dry dock a ship and proceed to cleaning. This procedure is very time consuming and of high cost, but it is the only available solution nowadays for SRY’s. On the other hand, this cleaning activity is the first to be done when a ship needs maintenance and/or some repairing, being the last the main activity of SRYs. So hull treatment is required and, at present time, is done manually in drydock using different adapted methods like grit blasting or water jet, and it has to be noticed that, in itself, it is a very contaminant operation (dust contains always painting particles), it is harmful for human operators health and it is a very uncomfortable job.

To provide a solution to these problems an EC funded project (G3RD-CT-000-00246) has been organised: AURORA (Auxiliary Climbing Robot for Underwater Ship Hull Cleaning of Sea Adherence and Surveying). The present project partnership brings together 7 partners with complementary roles: the Industrial Automation Institute (IAI-CSIC) which is the Project co-ordinator, two ship-repairing yards, T. Kalogeridis&Co. Inc. and Unión Naval de Barcelona, Algosystems S.A., the Division of Robotics, Department of Mechanical Engineering, from Lund University, SAIND, manufacturer and vendor of equipment for shipyards, and Riga Technical University.

So, the AURORA scenario consists in the underwater hull that after some time of ship operation is plenty of marine incrustations, where a new kind of underwater climbing robot equipped with special tools should perform cleaning and surveying tasks. That scenario presents large dimensions and exhibits some areas of very difficult reach-ability and poses some additional technical difficulties. The paper describes the overall design of the underwater robot with special focus on its control system. As it has been conceived the underwater climbing robot control is a human-in-the-loop process. Human-Machine Interface (HMI) design takes this into consideration whether using direct control or supervised control. This interface includes a control command set used to select the machine trajectory and a graphic representation used to get information about the robot and its environment. Fig.1 shows the concept of AURORA project.
2. Advantages provided by AURORA project

As it was commented above, all known technical solutions on ships' hull cleaning are mainly based on the use of dry dock. The use of dry dock is very expensive, so ships' hull cleaning from sea adherence is not made very often usually, something like every three years. By that time, usually, there is a necessity of repairing of protective coat. Therefore, during the dry dock cleaning process a ship is being cleaned both from sea adherence and old protective coat. So, a ship operates having a lot of sea adherence during a long period of time, which worsens its operational characteristics (dotted line in Fig. 2). Sea adherence that has been got in warm seas could decrease speed of a ship considerably (about 10% in a short time: six months). Moreover, a presence of large amount of sea adherence leads to more intensive destruction of protective coat.

![Ship performance improvement with AURORA concept](image)

In contrast to above mentioned, the use of proposed technical solution allows to clean a ship from sea adherence without the use of expensive dry dock. In this case it is possible to clean a ship much more often, for example, twice a year (solid line in Fig. 2) and a ship could be in a much better state at all times. Even in the case when it is necessary to make inspection, repair or restoration of protective coat of a ship in a dry dock, it is possible to make a ship's hull cleaning out of a dry dock first, which would allow to use dry dock more efficiently than now.
3. AURORA control system

3.1. Basic features

In this paper special features of design and new control algorithms for an underwater climbing robot are presented. Because the robot is being implemented to execute cleaning tasks on ship hulls, it must be emphasized that special considerations have to be taken into account in the development of the control system because of both the operating environment and the requirements of the cleaning process.

Thus, to achieve a proper quality of cleaning, the robot has to move with a suitable velocity, following some kind of strategy to cover the foul zone. This leads to the importance of controlling the direction of the robot and its velocity. In addition, the robot has to move upon an unknown steel curved surface with local irregularities. The proposed robot has three wheels for balance reasons and it is endowed with magnets to attach to ferromagnetic surfaces. Note that some buoyant elements compensate completely the gravity force, so the normal forces are not affected by the robot orientation. Usually, the design used for a three-wheeled mobile robot includes two motors: one power motor and one steering motor on the same wheel, Jones (200), Borenstein et al. (1996), Necsulescu et al. (1993), Povazan et al. (1995), or two power motors on different wheels, with a third wheel having passive steering, Zhao et al. (1992), Shim et al. (1994), Kanayama et al. (1990). Such designs demonstrate a good performance only when it is not necessary to have a big traction force. However, in the case considered, the robot moves underwater (which creates a water resistance force) and performs a technological cleaning process (creating an additional resistance force), such that a significant traction force is needed. The processed surface is also characterized with a very low magnitude of static friction coefficient, causing difficulty to obtain a big magnitude of traction force. Under all these constraints, the robot can experience traction problems and sliding conditions on one or more wheels. This situation makes the maintenance of the programmed velocity of the robot and the calculation of the robot’s position corresponding to the rotational angles of the wheels difficult. Moreover, significant trajectory tracking errors can be produced because of the external forces that are acting on the robot.

To solve these problems, it is necessary to have as many power motors for traction as wheels controlled independently. However, having such configuration, a new control algorithm is needed. This velocity-force control algorithm has to sustain the required velocity for the technological process, and,
at the same time, distributes optimally the traction forces between the wheels, Akinfiev et al. (2001). The following sections are focused on explaining the control algorithms for this particular robot.

Fig.3 shows a first prototype and its dimensions. Fig.5 shows the disposition of the wheels, the cleaning unit and the magnets, also acting forces are depicted. Fig.4 shows one view from the teleoperation station where images and graphics are mixed to provide a virtual reality feedback to the operator.

![Teleoperation of AURORA](image)

**Fig.4: Teleoperation of AURORA**

![Grasping and Normal Forces](image)

**Fig.5: Grasping and Normal Forces**

### 3.2. Control algorithm

Conventionally, the wheeled mobile robot movements are either velocity controlled or torque controlled. However, in the specific application described above neither of the methods suggested satisfies completely the requirements. In the first one, there are many complications to estimate the correct desired velocity for each wheel since the mobile robot moves upon a curved surface. Moreover, the surface has a lot of irregularities that could even brake the wheels. The second possibility represents a good choice to solve the control problem. Controlling torques applied to the wheels prevents the wheels from sliding on low coefficient of friction surfaces. But it is not enough because it is very important that the underwater climbing robot moves with a suitable and constant velocity to perform the cleaning tasks efficiently. If the velocity is too high the obtained quality of cleaning may not be
enough. If the velocity is too low, despite a good quality of cleaning being obtained, the polish of the hull material can be produced. If the velocity is varied the result will be a not homogeneous cleaning. Therefore, not only an accurate velocity control must be provided but also a torque control is required in order to improve the wheeled mobile robot performance.

The mixed velocity-force control algorithm implemented chooses one of the wheels as a reference. On this reference wheel a velocity control is realised using a PID compensator with a feedforward term. Modifying the voltage of the motor the control algorithm minimizes as much as possible the difference between the real velocity measured on the wheel and a programmed velocity.

To prevent an excess traction force that may produce sliding of the wheels, the control system will verify simultaneously that the traction motor load torque $\tau_i$ applied to this wheel does not exceed a certain defined limit load torque $T_i$. The real limit load torque $T_{r_i}$ of each motor is connected with the maximum static friction force and it is the torque from which the sliding conditions begin. In this study, the real limit load torque $T_{r_i}$ is estimated experimentally. Nevertheless, if normal force sensors are installed on the wheels, the real limit load torque $T_{r_i}$ can be calculated using the equation:

$$T_i = N_i \mu R \quad \forall i, \quad i = 1 - 3$$

where $N$ is the normal force, $\mu$ is the friction coefficient, and $R$ is the wheel radius.

However, because of the risk of working close to the real limit, a safety margin for the robot has been imposed defining a limit load torque $T_i$ less than the real limit load torque $T_{r_i}$. Thus, while the traction motor load torque $\tau_i$ remains within the limit, the voltage applied to the traction motor can be modified by the velocity control. But when the traction motor load torque $\tau_i$ begins to be equal to the limit, it is necessary to reduce the motor voltage until the traction motor load torque $\tau_i$ equalizes the limit load torque $T_i$ without maintaining the conditions established by the velocity control. This voltage will be held until the moment in which the voltage needed to support the velocity is smaller than the needed voltage to support the limit load torque.

The next step is to achieve a proper traction force distribution while the sliding of the wheels is prevented. For this purpose, the special algorithm controls the traction motor load torques of the rear wheels using a lead-lag scheme with dynamic feedback and distributes these load torques according to the maximum possible load torque on each wheel:

$$\frac{\tau_1}{T_1} = \frac{\tau_2}{T_2} = \frac{\tau_3}{T_3}$$

where $\tau_1, \tau_2, \tau_3$ are the load torques applied to the wheels, and $T_1, T_2, T_3$ are the limit load torques.

Then the control uses the load torque measurement from the reference wheel to minimize

$$\tau_i - \tau_1 \cdot \frac{T_i}{T_1} \quad \forall i, \quad i = 2 - 3$$

where $\tau_i$ is the load torque applied to the wheel $i$, and $T_i$ is the limit load torque on the wheel $i$. 

520
It also verifies that the traction motor load torque $\tau_i \ [i = 2 - 3]$ applied to each wheel does not exceed its corresponding limit load torque $T_i \ [i = 2 - 3]$.

Due to the robot movement being a quasi-static process, the traction motor load torque is approximately equal to the internal motor torque. At the same time, the internal torque is proportional to the stator current of the motor so it is possible to work directly with currents instead of load torques, achieving a simplification in the control.

3.3. Control implementation

A sensor system provides suitable signals required for the control algorithm. Some of the elements that compound this sensor system are encoders and current sensors. A flexible control card was produced at the Industrial Automation Institute. It has 16 high-speed A/D channels for sensors, 4 PWM output ports and 24 input/output single bit ports. The card is responsible for interfacing the control PC with the sensors and the actuators.

The control algorithm was implemented on PC running real time operating system QNX 6.0. A human machine interface running on Windows 2000 was used for front panel control. This operator interface communicates with the control program through Internet Domain TCP/IP sockets. The robot operates not only by operator but also in autonomous mode. Fig.6 shows the mixed velocity-force control algorithm.

![Fig.6: Mixed velocity-force control algorithm](image)

3.4. Experiments and results

To demonstrate the effectiveness and applicability of the proposed method, a real time implementation of the control algorithm was developed and experimental trials were realised using a prototype with three independently driven wheels and with one additional DC motor for the steering. In the first experiment, the robot was programmed to move upon a horizontal flat surface with a constant velocity of 30\%s on the front wheel and in such way that the traction motor load torques of the rear wheels were equal to the measured load torque from the front wheel. This means a traction force distribution of 1:1:1. Fig.7 shows the behaviour of the velocity control while Fig.8 shows the load torques on the wheels. Then the robot was programmed to turn left 45° upon an inclined plane with a constant velo-
ity of 50°/s on the front wheel and with a force distribution of 1:1:1. Additionally, the robot had to move over two obstacles. Fig.9 shows the velocity control behaviour during the movement process. Fig.10 shows the load torques of the traction motors. It is interesting to remark that all the load torques are practically equal while the robot is turning.
Finally, the robot was programmed to move upon an inclined plane with a constant velocity of 50°/s on the front wheel and with a force distribution of 1:0.9:0.7. Additionally, the robot had to move over two obstacles. Fig.11 shows the desired velocity and the measured velocity on the front wheel. Fig.12 shows the load torques applied to the wheels.
4. Conclusions

An overview of AURORA project has been done. A control method that distributes force for improved traction, developed for the AURORA underwater robot has been presented. It is especially suitable for wheeled mobile robots that move upon curved surfaces with irregularities and small coefficient of static friction where it is critical for them to maintain good wheel traction without sliding. The results of the experiments show the good performance of the mixed velocity-force control algorithm while the robot is moving upon flat planes, inclined planes and planes with irregularities. In every situation, it was possible to distribute traction forces between the wheels while the velocity required was supported. A significant increasing in traction force without sliding of the wheels was achieved.

Acknowledgements

The present project partnership brings together 7 partners with complementary roles: the Industrial Automation Institute (IAI-CSIC) which is the Project co-ordinator, two ship-repairing yards, T. Kalogeridis&Co. Inc. and Unión Naval de Barcelona, Algosystems S.A., the Division of Robotics, Department of Mechanical Engineering, from Lund University, SAIND, manufacturer and vendor of equipment for shipyards, and Riga Technical University. AURORA project is funded by EC Growth’99 under contract G3RD-CT-000-00246, and author’s acknowledgement is extended to all the partners and to the European Commission.

References


ZHAO, Y.; BEMENT, S.L (1992), Kinematics, Dynamics and Control of Wheeled Mobile Robot, IEEE Int. Conf. Robotics and Auto., pp.91-96


The Role of IT in Shipbuilding

Stefan Krüger
TU Hamburg-Harburg, Hamburg/Germany, krueger@tu-harburg.de

Abstract

A review of the business requirements for European shipbuilding lays the groundwork for concluding requirements for information technology supporting competitive designs. Specific open system software must be combined with re-engineering the design process to obtain sustainable competitive advantage. The general guidelines are detailed and illustrated with examples from the FSG shipyard.

1. Introduction

There can be no doubt that no improvement of the shipbuilding process, regardless whether product improvement or process optimization, can be achieved today without massive support of information technology. Any program with the aim to reduce man-hours or to increase the technological level of the product itself always results in development demands of IT based tools, interfaces, data management or processors. Therefore, we can conclude that one of the strategic key values lies in the development, management and application of latest information technology.

Surprisingly, despite the strategic key value of IT based technologies, it can often be observed that investments in the IT sector are not determined by rational criteria. Neither the strategic value of that discipline nor a simple value for money balance are taken into account. This does especially hold for the shipbuilding industry with complex products, sophisticated production processes and logistics chains which all have to cope with the extremely short product development cycles and where the fierce competition has already led to a remarkable reduction of the players in the field, often due to technical delivery problems or extensive man-hours.

To remain in the market, European shipbuilding industry has to optimize products and processes to the maximum possible. Consequently, a strong engagement in R&D especially in IT technologies is a major strategic playing field. This paper describes main conclusions derived from the current situation and the author points out fields where development has to take place in the future.

![Fig.1: Actual costs and cost level fixed by the design for one RoRo ship and all 6 ships of the series](image-url)
2. Early design versus detailed design

The shipbuilding industry is especially characterized by rather complex products which have to be developed in extremely short time. This basically well known situation is demonstrated by Fig.1. Here, the actual costs generated by a specific phase of the product development are plotted against the total costs fixed by the design. The diagram was developed based on the cost calculations of FSGs RoRo series for the Turkish customer UND. The graph shows that roughly 70% of the total costs are fixed within the first four weeks of the project. On the other hand, few costs are generated during this phase only. This shows the importance of the early design phase. Errors or insufficient risk control during this phase leads to expensive corrections during detailed design or production phase or to massive man-hour increases.

There is no doubt that the delivery problems some shipyards had can be traced back to a wrong decision (or acceptance of too high risk) during the initial design phase. As competition still increases and the market is totally customer controlled, the increased productivity on one hand and the demand for faster and more tailor made designs on the other hand will increase the gradient of the initial design phase even further, making the strategic value of this phase more important. Speed becomes the relevant key value, and in the future, we have to face the fact that it will not be sufficient to control about 70% of the costs during the initial design phase only, the task is to set this number to 100% (total risk control). The importance of the initial design phase of the product development has been disregarded by the software suppliers: Most of the software development focused on improvements of the detailed design or production planning.

In total, these facts result in the following conclusions for the strategic development of IT based projects:

- To survive in the market, a shipyard has to offer more ship (design competence) at a lower price (production competence) than a competitor. Design competence is dominated by the initial design phase, where key values are speed and risk control.
- To support the initial design phase, we need tools that are fast enough to deliver results under the severe time pressure on one hand. They have to support the design process in such a way that they can give answers to the relevant design changes that are performed during the product development.
- For many design tasks, it is not absolute numbers that are needed but gradients that can help to optimize the design and to control the technical risk.
- As the main task of the initial design phase is to analyze a design with respect to changes to optimize it, we need tools, processors, and product models that are flexible enough to handle changes. As the design process is always inconsistent, we need IT based solutions that support the designer in this respect. Designing is more important than describing.

3. First principles versus rule based design

Due to the shortage of time in product development, designs are mainly based on the fulfillment of empirical criteria or on design rules (rule based design). These rules have been developed by experts in the past (e.g. rules of classification societies) based on empirical criteria related to the ships existing at that time. Another aspect of these rules is that the design know-how is concentrated in these rules, with the consequence that designs could be made by less experienced people (rule based or cook-book designs). Nowadays, the situation has changed drastically: The pressure on the market and the permanent need for competitive designs and
products forces the designer to go far beyond the horizon defined by the empirical design rules. Margins become smaller on one hand, and completely new ship types and developments lead to physical phenomena that are not covered by empirical design rules. Consequently, there is a strong need for first principle or simulation based applications to support the design process in that way the designer needs to take his relevant decisions. On the other hand, most of these first principle based methods are numerical methods and they require a large amount of input data, which are typically idealized discretizations of the real ship. These models (e.g. finite-element method (FEM) model of steel structure) including data pre- and post-processing required much manual work, and in many cases the cycle times of the calculation did not meet the requirements by the designers. In this context, the calculations could hardly be used to check a given design (if the results were available before the first steel was cut), but were far away from being design tools.

The goal was (and still is) to develop first principle based tools based on theories fast enough and accurate enough to deliver answers to the designers questions during the initial design phase, to automate data pre- and post-processing based on information available from preceding steps of the design process and to rearrange the design process accordingly.

![Flowchart of re-engineered steel structure design process based on first principle methods](fig2.jpg)

The complete re-engineering of the steel design process within the development of the first-principle based ship design system E4 (here demonstrated by a re-engineered steel design process, Fig.2) resulted in the reduction of the cycle time for the global FEM calculation from more than three months to less than two weeks, including pre-approval by the classification society. The reduced cycle time then offers the opportunity to design and optimize the steel structure for purpose, resulting in less weight, lower vibration levels, less production man-hours and reduced delivery problems.

The concept exploits distributed developments, where research institutes and universities deliver basic calculation methods or principles, which are then made fit for practical use in close cooperation with the industry. Consequently, these findings result in the following conclusions for the strategic development of IT based projects:
• First-principle based methods should replace empirical design rules wherever possible. These tools need to be customized to meet the designers needs. Over-engineering should be avoided (appropriate methods for appropriate purposes).

• To benefit from these methods, they need to be automated as far as possible, and the design process needs to be re-engineered accordingly.

• The application of these methods requires highly skilled and trained engineers with sufficient knowledge about the modeled physical phenomena and processes.

4. Customized codes versus off the shelf developments

Like in other industries, it is often stated that IT developments should benefit from standardization, too. This aspect, which is in principle a useful approach, on the other hand often leads to misinterpretations in such a way that tools should be developed for a couple of purposes or applications to cover a greater market. From the economical viewpoint this may be correct, but it does not take into account the strategic needs which are especially required under the severe time pressure of the early design stage:

From the evolution of the species we know that the more severe the boundary conditions are, the more specialized the species need to be to survive. The same holds for the application of software tools in product development: The better a code is tailored for a specific problem, the more efficient it will solve the problem, where efficiency is defined by the quality of the result divided by the time needed to achieve this result.

Fig.3: Two examples of customized solutions: The FEM grid generator is tailored for ship structures, and the rudder flow in propeller slip-stream also is a very special application

For example, a typical FEM mesh of a steel structure, Fig.3 (left), has completely different requirements than a mesh of an automotive manifold, and consequently needs different strategies to generate the mesh, which can hardly be implemented both in one tool. The flow around the rudder in the propeller slip-stream is related to the flow around an airfoil, but for good reasons, no common aircraft code is used. As the tailored development needs additional investment in development (which has to follow economical principles to be competitive), a standard on a reasonable granularity should be introduced. This can be achieved by a tool box of standardized modular numerical calculation and simulation procedures that can be combined and tailored to specific tasks. If backed by an open modular system architecture and process knowledge, problem customized tools of high efficiency can be developed in short time from these software bricks, which are highly specialized tools that are often developed by universities or other specialists.

From these findings, the following consequence arises:

• To cope with the efficiency required during the initial design stage, customized codes that have been optimized to solve a specific problem need to be used.
To develop such kind of codes with reasonable investment, a tool kit of sophisticated standardized procedures needs to be used in an open context that can be easily combined to produce customized tools.

5. Expert systems versus experts

One major playing field in research and development is optimization of ships or subsystems. This is related to automatic decision making which in some cases results in what is called expert systems. In principle, optimizing strategies are useful tools, but they can only be applied usefully to problems where the target function can easily be expressed and where also the boundary conditions can easily be formulated.

This is not the case for most problems in ship design, and consequently the question arises if a computer is a useful tool for complex technical decision making. This is to be doubted, because one key value of human experts is the ability to make creative decisions based on poor and inconsistent information. Because human experts have a broad look on the problems also taking into account side information or experience, where computer systems tend to look on the simplest problems with the intensity of a laser beam. This indicates that the famous and sometimes still propagated "make ship" button will probably lead to a poor result. Can we then conclude that global optimization does not make sense in ship design?

Of course not, but we should look on it from a designers point of view: Especially when first-principle algorithms are used, there is a strong need to automate procedures that have proven to serve the designer in the best way: For example, a standard procedure can be programmed for the optimum adjustment of the bulbous bow or the bilge radius based on CFD calculations and systematic hull form variations. The procedure reflects exactly the most efficient way to solve the problem (because it was defined by the designer of the hull, probably an expert) and can be automated easily as one design task. In the same way, other tasks may be automated, and as results, local optimizers combining a couple of tools are developed. On the next level, if sufficient experience is gained by the designer handling these tools and the related information flow, there will arise the need to automate and combine some of the local optimizers to more powerful tools and so on.

Fig.4: Bottom up generation of design optimization tasks from design requirements
This shows that the development of tools must follow a bottom up approach to be successful, whereas top down views tend to tackle the problem too academically and therefore often fail. This bottom-up development can obviously only be carried out if sufficiently skilled designers of high process knowledge are willing to make their knowledge available. But if the same people define the development scenario for a tool that will be the end users (or close to the way of thinking of the possible end users), this is twice efficient. On the other hand, such kind of development can lead to anarchism and unclear software structures (typically most developments in the IT scene are top-down approaches). It must therefore be guided by a clear, open and modular system architecture with a high level of standardization at suitable granularity.

From these findings, the following conclusions can be drawn:

- Tool development should follow a bottom up approach, to reflect the necessities of the design process as far as possible and to identify beneficial applications. So humans and computers contribute where they are most efficient.

- Low level optimization leads to higher level optimization, provided the system architecture prevents the process from anarchism.

- The development must be based on highly skilled design engineers with sufficient system and process knowledge.

6. Data model interfacing versus complete product model

One major playing field in IT development is the development of product data models and the related interfaces. Data models are subject to standardization (e.g. ISO APs) and the same holds for the required interface programs. Most of today’s problems regarding IT developments are believed to be solved if an appropriate, standardized product data model is developed that reflects most of the views needed for today’s applications and if the data interfaces required would exist. This opinion strongly coincides with the fact that most IT development was carried out for the detailed design phase or the data exchange with sub-suppliers (e.g. classification society), where in fact most of the existing problems are data management problems.
Fig. 5: Different views on design relevant data, e.g. compartmentation. Some tasks can be served directly by the data model implemented, some by data export to other applications.

On the other hand, this development disregards the fact that the freedom to make decisions during the design process decreases drastically from the early design stage, where it is a strategic requirement, to the detailed design, where it is not necessary anymore, Fig. 1. Consequently, the data models should follow these requirements: In the beginning, we need more or less generic models that handle a minimum of dependencies whereas during the detailed design we need models that can describe the ship including all necessary details. Obviously, the demands on the data models derived from these views are completely different, and it can be concluded that one data model would never serve both demands, Fig. 5. The conclusion is quite simple: To cover the whole process from initial design to production and delivery, different tailor made data models are needed to serve each part of the whole process with the necessary efficiency. Consequently, algorithms and interfaces have to be developed to transfer a specific view into the next more detailed model further downstream in the development process. From these basic facts, the following conclusions can be drawn:

- The specific views on the design process requires different data models, each having a specific focus.
- To ensure a sufficient efficiency, powerful procedures and interfaces have to be developed to transfer the contents of one data model downstream to the next application towards detailed design.

7. Simultaneous engineering versus sequential work flow

If once fast and reliable design tools exist and if they can be applied during the initial design stage, this gives the possibility to couple different engineering disciplines early enough to optimize not only a part of the system, but the complete system itself. It is the multi-disciplinary, simultaneous engineering that makes a design competitive, as the following example will show. On the other hand, one has to be aware that new design tools also influence the design process, which in some cases has to be re-engineered as already mentioned above.
Simultaneous engineering becomes possible if design tasks, that had to be performed in the late design stage, can now be supplied with the required data in the initial design stage. Therefore, simultaneous engineering strongly depends on a well re-engineered design process as well as on a suitable IT infrastructure that has optimized the data work flow and generation. Open system architectures providing sufficient flexibility to combine methods and data flows according to the process needs are the fundamental basis for these requirements.

8. Open system architectures versus black box systems

In the past under the dominating influence of share holder value strategies, outsourcing of IT competence into turnkey black box systems has been propagated by substantial parts of the shipbuilding industry. This was a reasonable solution when competitiveness was to be gained out of productivity increases which were dominated by the late design phase. In this context, the minimization of complex interfaces was the main reason for outsourcing IT competence to external turnkey solutions.

On the other hand, it was already mentioned above that the hard competition forces specialization, which does also hold for the software industry. As a consequence, it can not be expected that a single vendor (or system) is competitive in all fields, especially when we deal with highly specialized first-principle base simulation algorithms. Therefore, a more competitive solution seems to be what we call an open method bank. The structure of such a method as first suggested by Nowacki (1988), and a refined architecture is plotted in Fig.7. The main concept behind this approach is that the method bank consists of a number of methods which share a common data base and user interface. Typically, these methods are sophisticated developments.
(mostly by universities or other research institutes) and represent the scientific state of the art. To integrate these methods into the method bank, only a pre- and post-processor, based on standardized data management structures, need to be developed. If the methods are organized according to the typical data flow of the product development phase, much information can be generated by a minimum of user input. As this concept is highly modular and flexible and as it relies on latest state-of-the-art kernels developed by research institutions combined with the experience of ship designers, it clearly follows the approach of specialization and competitiveness. The disadvantage of this concept is that extensive in-house know-how is required to adapt the concept for the individual needs. On the other hand, if this know-how is available, it serves the product development, too. From these arguments, the following conclusions can be drawn:

- Open and modular systems have competitive advantages compared to black box monolithic concepts.
- New technologies such as first-principle or simulation-based design can be more powerfully implemented in open and flexible systems.
- The open method bank concept is most efficient in integrating heterogeneous and specialized tools, provided sufficient in-house know-how has been established.

9. Conclusions

No improvement of products or processes can nowadays be achieved without massive investment in IT technologies. In shipbuilding, the initial design phase should become the main focus of strategic IT development due to the fact that the technical risk is mainly driven by the initial design. To remain in the market means control of technical risk before signing the contract. Ships which are nowadays designed by empirical rules will be designed by first-principle based simulations, simply because cook-book ships do not represent competitive products for European shipyards. IT system design as well as the culture in shipbuilding will have to take this unavoidable change into account. This general trend – combined with the severe pressure from the market – will force specialization, especially in IT development. Universities with their natural ability to create solutions for complex mathematical problems will definitely be back on the scene. The development in total will become more dynamically and complex, and the promising approach to cope with the new flexibility seems to be the open method bank concept. We should never forget that each concept – even the most efficient one – requires highly skilled and well educated people with scientific background. In this respect, the shipbuilding scene might require conceptual re-engineering.

References


A Modeling and Simulation of Production Processes at a Virtual Shipyard

Jong Gye Shin, Seoul National Univ., Seoul, Korea, jgshin@snu.ac.kr
Kwang Kook Lee, Seoul National Univ., Seoul, Korea, ykk21@snu.ac.kr
Jong Hun Woo, Seoul National Univ., Seoul, Korea, whdgn2@snu.ac.kr
Jang Hyun Lee, Xinnos Company, Seoul, Korea, jhlsl@gong.snu.ac.kr
Se Hwan Kim, Samsung Heavy Industry, Geoje, Korea, ksehwan@samsung.co.kr

Abstract

Simulation-based production can be very effective in shipbuilding in order to save costs and time, to increase safety for workers, and to prevent bottleneck processes in advance. Digital shipbuilding system, a simulation-based production tool, is being developed to achieve such aspects in Korea. Product, process, and resources of a shipyard are three groups to handle data efficiently. The forming factory performs the cutting, press bending, and line heating processes for steel pieces. Rectangular steel plates are supplied as the input for the forming factory and piece parts, flat or curved, are produced. In order to simulate material flow in subassembly lines at a shipyard, the production process is modeled for the subassembly process. The subassembly line consists of several sub-processes such as tack welding, arrangement, fit-up, robot welding, and manual welding. Rectangular steel plates are supplied as the input for the subassembly lines.

The conceptual model was used to build up a framework for product, process and resource planning. Initially, the characteristics of the shop resources were analyzed based on the shipyard data and the layout of subassembly lines was designed with the resources. A production process was then modeled using the resources and the layout. The production process modeling of the forming factory and the subassembly lines was performed using the discrete event simulation method for productivity analysis. Using the constructed resource and process model, the productivity and efficiency of the shop were investigated. Variations in the resource performance such as a new welding robot and the number of workers were examined to simulate the changes in productivity. It can be seen that the bottleneck process is floating according to the nature of new resources. The proposed model is viewed three-dimensionally in a virtual environment so that space allocations for the resources and interferences between objects can be easily investigated.

1. Introduction

Manufacturing industries try to reduce the lead-time and production cost of products by digitalizing their manufacturing systems and simulating their respective scenarios. The cutting edge industries among them work to establish new paradigms such as product lifecycle management (PLM) by using virtual manufacturing in the environment of enterprise application integration (EAI) including CAX (CAE, CAD, CAM, CAPP, etc) and product data management (PDM). PLM is an answer for today’s design-driven, customer-centric companies focused on delivering mass-customized and personalized products constructed from modules designed and built around the globe. PLM has become a strategic tool for manufacturers, along with enterprise resources planning (ERP), supply chain management (SCM) and customer relationship management (CRM), Fig.1. (http://www.3ds.com)

Shipbuilding industries are walking in the same step to launch digital shipbuilding system, which is constructing a digital model of their shipyard to control and revise all of elements in their shipbuilding process. And Virtual manufacturing system (VMS) under such an environment is adequately utilized to predict how much improvement of productivity a manufacturing system will take and what behaviors it will show when its environment is changed, for example, a new resource in its layout might be inserted or exchanged.
2. System architecture and methodology of virtual manufacturing simulation

The conceptual components of the applications used for the simulations are explained in Fig.2. The system architecture was constructed by integrating the physical and logical components such as a product data model, a product data generator, a PPR (Product, Process, and Resource) of an actual simulation systems and virtual simulation systems. A PPR Hub manages this system overall. Fig.2 shows the physical deployment diagram corresponding to the components that particularly emphasize the discrete event simulator and the robot simulator.

A step toward understanding a methodology for validation of process planning would be to study the framework of the methodology. In order to verify how the sub-assembly process works, we have to investigate the material flow through DES (Discrete Event Simulation). The DES model needed a cycle time of sub-process, which was not easy to extract from the design. For example, we could check up a cycle time in the robot welding if the normal welding torch was changed to high-speed rotation torch. In addition, we obtained knowledge of how the improved sub-process influenced the other on the line. For this, it is needed to analyze and model the system to build up the framework for the process planning.
2.1. Manufacturing System Analysis

Object-oriented methodology, found in software engineering, is necessary for developing systems, *Shin* (2002). Fig. 3 indicates that all of the information is based on modeling and analyzing the system by product, process, and resource. The Use case diagram shows the main activity and usage of the target system, Fig. 4. The main activity of the forming factory is to cut, bend a plate by rolling and bend a plate by line heating.

![Fig.3: Object modeling with PPR concept](image)

The Class diagram shows the abstracted and identified objects and the relationships of the forming factory. It is shown that the forming factory is composed of several jobshops such as the cutting jobshop, the roll-bending jobshop, and so on. Each jobshop has the relevant products, processes and resources.

The standard processes of the forming factory are cutting, roll bending, and line heating. In addition, the relevant jobshops are the cutting jobshop, the roll bending jobshop and the line heating jobshop. Each jobshop has a unique process and resources. The class diagram defines the resources and their relationship with the relevant jobshop, and the sequence diagram defines their interactions, namely, the process.

![Fig.4: Use case diagram of the forming factory (left), Class diagram of the forming factory (right)](image)

An analysis and design methodology developed for the improved business (TO-BE model), which derived from the abstraction and decomposition of actual organization (AS-IS model).
2.2. System requirement through the logical tree

The system requirements should be determined according to the work. This study used logical trees, which are the tools of a thinking process, to diagnose the system performance, and then determined the system requirements. Goldratt (1990) developed a methodology, called as Theory of Constraints (TOC), for system optimization more than process maximization. Dettemer (2001) introduced an example of the logical trees developed for applying the TOC into practice.

The thinking process is part of the TOC. The thinking process is used to develop a breakthrough, and a compromise, in a complex problem that has a causal antagonistic relationship. The logical trees play the role as a communication tool among system developers and unite logic and intuition. There are five distinct logical trees. The trees include a Current Reality Tree, a Conflict Resolution Tree, a Future Reality Tree, a Prerequisite Tree, and a Transition Tree. This study used the Current Reality Tree and Conflict Resolution Tree. The logical trees on the forming factory are shown in Fig.5. The core problem of the forming factory such as those from the magnetic crane becomes clear. Therefore, the requirements of the virtual simulation modeling of a forming factory are determined.

![Logical trees of a forming factory](image)

Fig.5: Logical trees of a forming factory
Top: CRT (Current Reality Tree), Bottom: CRD (Conflict Resolution Diagram)

2.3. Manufacturing System Process Planning

The conceptual model was used to establish a framework for product, process and resource planning. Process Engineer™ was utilized to be PPR hub which is the integrated manufacturing database for persistent storage and management of all information. The implemented model enables to develop and view, in 3D, the product, process and resources objects in the process plan, Fig.6.
Process Engineer™ provides high planning quality through methodically structured planning, early recognition of process risks, re-use of proven processes, traceable changes and decisions, and usage of scattered process knowledge. It is used during the conceptual product design phase, with the process design and alternative manufacturing concepts maturing through the conceptual, pre-planning and detail planning stages up to production. (http://www.delmia.com)

2.4. Virtual manufacturing simulation

2.4.1. Robot simulation

A robot simulation model designed on the basis of the real shipyard can obtain precise cycle time of the process without any time measurement. The cycle time from the robot simulation is validated by the experiment with time study, which is an analysis of a specific job in an effort to find the most efficient method in terms of time and effort, Fig.7.

Fig.7: Robot simulation (left), Discrete Event Simulation (right)

2.4.2. Discrete Event Simulation

The object of material flow modeling is different from that of jobshop modeling. The object of jobshop modeling is to validate the behavior of the resources with its related processes and products. On the other hand, the object of the material flow modeling is to predict the overall physical and logical system behavior influenced by internal and external changes, and to suggest systematic and reliable alternatives.

Fig.8: Verification and Validation

Fig.9: Deployment diagram for the forming factory
2.5. Verification and validation

Conceptual model validation is defined as determining that the theories and assumptions underlying the conceptual model are correct and that the model representation of the problem entity is “reasonable” for the intended purpose of the model. Computerized model verification is defined as assuring that the computer programming and implementation of the conceptual model is correct. Operational validation is defined as determining that the model’s output behavior has sufficient accuracy for the model’s intended purpose over the domain of the model’s intended applicability, Fig.8.

3. Application

3.1. Robot simulation

A virtual model of the forming factory was implemented concretely based on the results from of the analyses. The physical composition of the virtual model is shown as a deployment diagram in Fig.9. The jobshops (Cutting, roll bending by 400 ton, 1500 ton, and 2200 ton, line heating), buffers, and cranes were modeled as devices, and these devices work together to fabricate the plate.

All the devices were modeled three-dimensionally and constitute each jobshop. These 3D models were reused to make the material flow model real. The resources of the forming factory include a machine, a crane, a worker and a buffer. These resources were modeled three-dimensionally using virtual reality technology.

Fig.10: Virtual manufacturing model of each jobshop (Cutting jobshop, 400T Roll press jobshop, 2200T Roll press jobshop, 1500T Roll press jobshop)

These solid models were assembled as devices with kinematics using the IGRIP™, which are widely used for robot simulation. The scenarios of the simulation model were then established according to the process sequences. The virtual simulation models of each jobshop are shown in Fig.10. The validity (about such a collision, performance, and errors) of each jobshop was examined through the virtual simulation. In addition, the cycle time of each jobshop for the material flow simulation was checked.

3.2. Discrete Event Simulation

3.2.1. Forming shop

The discrete event system requires a determination of all the interim products and their critical path. The entire product that is in processing is abstracted as an interim product in the object analysis phase. However, the interim products are broken down into parts of each resource in the implementation phase. Therefore, the interim products and their critical path were determined using material flow analysis. Fig.11 shows the material flow diagram, Table I defines the interim products.

Each product in the process (interim products) has its own working path to be processed in an actual forming factory. However, it is meaningless to assign the processing path to every supplied product without real product data information in the simulation. As an alternative, the processing path of the interim product is determined stochastically considering common hull information. Resources such as a cutting machine and a roll-bending machine fabricate the interim product during a cycle time. The cycle time in this study was determined stochastically considering both the field information and the simulation time of the jobshop model.
3.2.2. Subassembly line

Observation of the workstation permits us to define standard processes such as tack welding, arrangement, fit-up, robot welding, manual welding, finish, and back heating. Even though we could obtain the new cycle time of robot welding, it was not easy to answer questions such as how many throughputs would be produced in the sub-assembly line and how an improved process would influence the other. The DES model was needed to validate the improved system.

Variations in the resource performance such as a new welding robot and the number of workers were examined to simulate the changes in productivity. Communicating with workers at the job-site was performed for verification and validation of the DES model.

4. Conclusion

In this study, a production process model was implemented for the forming factory and sub-assembly line in shipyards. The manufacturing system analysis and modeling were carried out to build up the framework for process planning. We developed an integrated methodology for validation of process planning. Theory of constraints was applied to determine major parameters that affect the productivity. In this particular case, the material handling system appears to be a critical constraint. The productivity efficiency is investigated using the constructed model. Thus, simulations with discrete event system were carried out for possible modification of resources. Virtual manufacturing simulation was utilized to validate the productivity analysis of the high-speed rotation torch. It was concluded that the procedure of the proposed methodology could be used for more precise simulation in order to explore various parameters in to increase productivity.

Acknowledgment

This research was financially supported by Korean Ministry of Science and Technology through National Research Laboratory program and by Korean Ministry of Commerce, Industry and Energy and Ministry of Information and Communication under the contract of Industrial Technology Infrastructure Promotion Program. In the end, members of Digital shipbuilding Innovation Center (DSIC) help us to accomplish the project. The author would like to express their sincere gratitude to the supporters.
References


IT Support for the Integrated Engineering Process of Complex Vessels

Wolfgang O. Vogel, Blohm+Voss GmbH, Hamburg/Germany
Wolfgang.Vogel@tkt-blohm.thyssenkrupp.com

Abstract

Blohm+Voss has recently started an initiative aiming at the support of the "integrated engineering process" by a coherent and homogeneous set of IT tools/systems. The new IT architecture has been designed to cope with the complex technology used in modern Navy vessels and to handle the flow of information conforming to the complexity of today's shipbuilding contracts (i.e. cooperation with other shipyards; B2B communication with subcontractors and customer's organization). The core of the new architecture is established by a Product Data Management (PDM) system acting as a backbone between the various engineering and management tool clusters. The main tool clusters can be characterized as CAD systems (3D and 2D), Simulation Based Design (SBD) tools, planning systems (design and production), purchasing and controlling/reporting tools. The presentation will cover the overall architecture of the new process oriented "System of Systems", will report on strategies for its roll-out and the migration from old systems, and will finally highlight some of its "interesting constituents".

1. Introduction

Modern shipbuilding industry is characterized by its capability to develop and integrate complex systems, which have to sustain their full operability even under adverse environmental conditions. Depending on the type of vessel, an optimization of the design is to be performed against quite different sets of "Measures Of Effectiveness (MOE)", resulting in a "best value for money product". The other important area of optimization in the shipbuilding industry is the optimization of the production process itself. "Lean production", "just-in-time delivery of components" and many different types of cooperation have been introduced already a long time ago, and usually all the production is combined with powerful planning and controlling systems in order to minimize production costs. However the fierce competition on international markets dictates further improvements in shipbuilding industry. The project described in the following text shall support such improvements by providing state-of-the-art tools for information handling and exchange, workflow control and integration of various application programs.

2. Areas to Be Supported

From an analysis of the main processes, based on interviews and compiled into overall process models, but also from a simple evaluation of the IT resources used, four main areas of activities can be identified in the “overall shipbuilding process”:

- Integrated Engineering Management
- Integrated Production Management
- Integrated Planning
- Integrated Controlling.

The largest number of IT requirements result from the engineering activities, however the information exchange with the others is essential for the "holistic" approach to optimization. The following description of activities shall outline the environment and the requirements for an IT system which allows the sufficient integration of major applications into a “collaborative workbench”.

542
2.1. Integrated Engineering Management

The process of design, construction, production, test and support of the ship system is affected by an increasing interconnection of the multiple subsystems installed onboard modern vessels. Especially the interaction between platform and warfare systems of naval vessels results in linked performance criteria. The main items driving this interaction are:

- Increasing complexity
- Increasing internal linkage of own ship subsystems
- Increasing embedding of own ship into external structures
- Shortening technology cycles
- Linked Performance criteria’s
- Complex Measure of Effectiveness (MOE)
- Complex Performance analysis

A general conceptual engineering approach is necessary to ensure a well balanced product and to control the technical design process. This Integrated Engineering Management consists of the traditional and proven techniques of System Integration, the structured analysis techniques of Systems Engineering, and a tailored set of simulation and performance prediction tools of Simulation Based Design technology.

Those three columns build up a powerful tool for the coordinated and controlled monitoring of the engineering process for shipbuilding. They are discussed in more detail in the following paragraphs to explain how they fit together.

2.1.1. System Integration

System Integration, as a task, regulates the combination of the several different engineering disciplines that are engaged during design and construction of the ship. Beginning with a given integration scope and depth, system integration defines the required interfaces, identifies what kind of information has to be transferred, and identifies what data and physical preconditions have to be considered. This purely technical relationship between the subsystem experts and the System Integrator is superimposed by the contractual relationship between the customer and supplier. As a customer of the suppliers whose systems are to be interfaced, the System Integrator provides the independence necessary to look after the customer’s requirements by maintaining technical control of both interface partners. In this way System Integration answers many “How” questions. It defines the work to be done to achieve a specified integration level. System Integration is fundamental engineering work, ensuring that the design is implemented successfully.

2.1.2. Systems Engineering

During the System Integration work a lot of “Why” questions are left open. ‘Why’ questions without an obvious answer occur especially when System Integration is transferred and applied to higher system levels. Looking at the interface between generic high-level components, like a Combat System and a Communication System, not only raises the question of how it should be implemented, but also why it should be implemented at all. The same situation occurs looking at the interfaces of the vessel itself. What is the interface between a vessel and its operational command authority? Systems Engineering picks up these questions by looking at the following two main issues:

- The strict breakdown and tracing of requirements
- The development of a consistent functional system model

These issues are strongly interconnected and interdependent. Based on an operational analysis for a new vessel, a requirement specification is established. This requirement set is mapped to a hierarchical structure that allows the derivation of design requirements from high-level requirements.
The requirement specification is usually the starting point for the design process. In an iterative process requirements are analyzed and allocated to required functionality. Operational scenarios are used to ensure a “customer’s view” during the analysis process and to support identification of required functions. By allocating the specified functions to components, and identifying necessary interfaces in between, the functional model maps the specified “virtual” world to the available “real” world. The compliance statement reflects the correlation of both worlds. “Missing” design is marked as deviation, while over-engineering should be identified as a cost driver (instead of being hidden as “fully compliant”). The functional model of the basic design is further investigated and detailed during the subsequent design steps, i.e. the basic System Integration work can begin.

However, prior to this in-principle design approach, the requirement specification has to be settled by looking at operational situations which reflect the intended use of the vessel. An operational situation can be understood as a condensed, high-level scenario. It describes the operational environment and conditions in general, but maintains a level of uncertainty. The operational situation provides enough information to illustrate the principal goals and limits of the vessel capabilities, which in turn, enables the extraction of high-level requirements. The extracted requirements are complemented with initial project constraints. These constraints define the initial limits of a new design, for example a maximum ship’s length due to available port facilities. The constraints are directly elevated to the specification level. While the requirements establish the degrees of freedom for the design process, the constraints define its boundaries.

This Systems Engineering process can be reflected in different views illustrating the inherent system dependencies. The functional modeling can be condensed with the establishment of three main models: the operational model, the requirement model and the functional (core) model. While the operational model aims at the identification of vessel’s missions and the derived high-level requirements, the requirement model establishes the requirement breakdown structure and requirement verification process. The functional model describes the overall system design solution as an implementation of the operational model and an answer to the requirement model.

These superimposed views of the Systems Engineering approach also reveal the need for the close, continuous and confident cooperation between the Customer and his Prime Contractor.

2.1.3. Simulation Based Design

A third significant and currently emerging capability of shipbuilding prime contractor engineering is Simulation Based Design. Simulation Based Design provides tools for an early closing of the feedback loop between specification and design. The increasing interaction between the various systems installed onboard results in the contribution of several different areas to single ship performance criteria. The simulation based evaluation of these criteria requires a comprehensive knowledge of vessel specific parameters, single product data, the system specific chains of cause and a database for model verification. This implies a simulation coordination activity, ensuring the coordination of different system experts on one hand, and the provision of key elements on the other, which is a classic Prime Contractor role. Depending on the particular customer focus and design specifics, the Prime Contractor is able, due to his experience and complete system design authority, to identify the extent to which a subsystem contributes to a certain performance value. This enables the setting up of a simulation strategy using internal and external experts, and transferring the results directly into the design process.

2.2. Integrated Production Management

The development of Integrated Engineering Management has mainly been driven by the increasing complexity of modern vessels and can be considered – from top management level – as an instrument of Risk Management.
In contrast to that, the Production Management of a modern shipyard is directly driven by cost and time, i.e. the need to increase productivity continuously. IT support for production management has been introduced at major shipyards already decades ago, but still no standard methodology has evolved. As a consequence, the three standard components of production management

- Material Handling,
- Production of Hull and Components, and
- Outfitting /Setting to work

are usually being optimized for each shipyard in a different way, and shipbuilding processes standardized amongst several shipyards are far from being implemented efficiently. However, the individual components listed above have a strong interrelation (therefore called Integrated Production Management) and a strong need for data exchange with the other main areas of the shipbuilding process. Hence production management turned out to be a major driving factor for a PDM solution.

2.3. Integrated Planning

The Integrated Planning comprises:

- Design Planning,
- Production Planning, and
- Cost Estimation and Control

each of those following proven algorithms, which again are quite shipyard specific.

Design planning usually follows the hierarchy of the product breakdown structure, however availability of resources, special treatment of “long lead items” and of course the interdependency of certain activities (e.g. coordination processes) have to be taken into account. A simulation component using parametric workload data from previous shipbuilding programs preferably supports the generation of these quite complex networked plans.

Production planning obviously has to cover the areas listed already in the integrated production management paragraph. Efficient use of resources is a key factor in this area. and any scenario which could bring the production to a hold (e.g. missing approval of a document by the customer) has to be watched carefully. Production plans are going down to a detailed level (“job numbers” for the workshop), hence their “maintainability” (easy to adapt to a new situation) is a key factor for their success.

The Cost estimation and Control measures have specific requirements during the various phases of a program:

Contract Cost Estimation (phases prior to contract signature) is dominated by rapid changes when the details of outfitting are negotiated with customer and subcontractor. Keeping the cost estimate always “synchronized” to the contractual documents (Scope of Supply, Scope of work) is its challenging requirement.

Cost control during program execution means keeping track of every change applied to the design and/or the contract itself (interface to “Change Management Workflow”) and keeping track of the work performed (monitoring of working hours actually spent). Signaling of all deviations from the estimated budget (time and money) is essential.

Finally a retrospective cost analysis is the tool for the improvement of processes / workflows (how can we do it better next time?). Since there is no standard rule for improving processes, a flexible, configurable access to all kind of information on the finished program is the major requirement for the IT system.
2.4. Integrated Controlling

The aim of the Integrated Controlling is to answer 3 fundamental questions:

Is the product delivered:
- As required?
- In time?
- At (estimated) costs?

A wide variety of tools is being used for that; starting from “fully blown” SAP systems up to very specific tools for benchmarking or balanced scorecard procedures. The most important requirements arising from those controlling tools to a PDM can be summarized as:
- Powerful and comfortable Search and Navigation functions
- Access to every kind of stored information easily configurable
- Correlation of data supported (access to relations in database)

3. The PDM Approach

The initial decision to introduce a PDM system for information storage and retrieval was based on the steadily increasing amount of data (documents, metadata, drawings etc) produced in Navy vessel programs. The shipyards experience in using hierarchical work breakdown structures for managing data as well as work packages relieved the decision to implement a product structure oriented system instead of “flat” document management tool.

Within a 3 month definition phase an analysis of the main processes involved in a typical shipbuilding program were analyzed, a breakdown structure corresponding to the structural breakdown of a vessel was established, a data model reflecting the data content of old IT systems (to be replaced) plus the requirements for new functionality was derived, and a migration concept for a changeover from the old IT systems to the new PDM system in 3 implementation steps was drafted.

An interesting detail of the product structure definition is the coexistence of a system oriented (system consists of subsystems consists of components etc) and a “steel oriented” (platform consists of modules consists of decks etc) substructures. This structure is presented to the user by “views” which allow an easy access to information like “what components are located in room xyz?”

4. Implementation Stages

4.1. Stage 1: Document and Change Management

The first stage of the PDM implementation covers the introduction of document management functionality including change management functionality according to the CMII standard. It has to be mentioned here, that the term document management here is not restricted to real document files but also includes handling of metadata and arbitrary file formats. It also includes viewers for text and CAD file formats and interfaces to plotting servers.

Strong emphasis has been put on the implementation of a CMII conformant change management, since it neatly meets the demands arising from daily shipbuilding work: very often it means implementation of changes “right on the spot”, whereas the documentation (drawings etc) has to be corrected after the change has been implemented. CMII supports this kind of process by a “fast track” mode, provided the proper authorization (“role”) has previously been granted to the person being responsible for the decision on the change's implementation.
Stage 1 of the implementation also comprises planning functions including an interface from/to dedicated planning tools (e.g. Microsoft Project) via .csv files.

4.2. Stage 2: Integration of Technical Databases and Systems Engineering Tools

In a second step the access to equipment specific technical data, and the execution of routine calculations (weight / power balance etc) will be improved by incorporating the huge amount of information nowadays kept in separate databases as metadata into the PDM internal database and implementing the calculations as “report functionalities” to the PDM.

More difficult will be the integration of B+V’s Systems Engineering tool “Design DataBase (DDB)”. The DDB is focused on the systematic development and documentation of complex system designs and provides features like functional and physical breakdown structures, requirements traceability, compliance matrices, document generation according to military standards, etc. Today’s PDM systems cannot cope with the “granularity” of information kept inside the DDB (The smallest entity of textual information in PDMs are documents, they cannot “point” to chapters/subchapters within a document) and might get into performance problems, if they should maintain the relations from the first specification documents all the way down to production documentation. Due to the complexity of the envisaged coupling between the PDM and the DDB preliminary tests of the data exchange have been performed already. Those tests have been performed successfully, however the most advanced final interface still needs a closer look into the “business logic” within the PDM’s 3-Tier architecture.

4.3. Stage 3: Production Planning and Beyond

Production Planning (PP) and related activities like generation of “Bills of Materials”(BOM) are core processes of shipyards and therefore have been implemented on IT systems many years ago. For those reasons the borderline between PP and classical ERP functions (maintenance of legacy data) has not been re-evaluated since. Upon the introduction of a new, workflow oriented tool with web-like navigation techniques a close look a processes, user profiles and the amount of information exchange at certain interfaces of the PDM is recommended. At B+V the largest user group for the interface to the production process is established by engineers and technicians. Hence, their standard IT environment will be the PDM. Compared to this, the number of “commercially oriented people” is relatively small. Although this result looks trivial it has considerable consequences for the PDM implementation and the following ER(P) system. The design decision taken states, that the PDM is the “leading IT system” for the shipbuilding process. This applies to information storage, retrieval and exchange. Other systems have to be interfaced either directly (e.g. embedding of office tools), via import /export functions (e.g. .csv files from other project planning tools) or via more complex interfaces. An example of a “more complex interface” in the PDM-SAP connector providing (real-time) bi-directional information exchange. This design allows a proper separation of applications without a break of the “information pipeline”.

5. The CAD Project

5.1. The Evaluation

In parallel to B+V’s PDM project, a joint effort has been undertaken by the three shipyards Nordsee-Werke Emden (NSWE), Friedrich Lürssen Werft (FLW) and Blohm+Voss (B+V) to evaluate, select and introduce a 3D CAD System to be commonly used by those yards, and in particular to be used in collaboratively performed shipbuilding programs.

The major goal of this project is to obtain a consistent 3D CAD product model of a complete vessel kept inside a PDM system. The product model goes much beyond the scope of a CAD file archive because objects defined within a CAD model are handled as metadata inside the PDM and hence can be related to other elements (e.g. specifications) in the PDM’s object store. At the end all information being required for the vessel’s design and production is available from the PDM.
The evaluation process for the CAD system was performed according to a requirements specification established at project start. The tests covered typical “design scenarios” from the shipyards and were based on sketches of real designs. Special emphasis was given to tasks like “Coordination” and “Outfitting” since those are dealing with a big number of parts / subassemblies and also require the import of data from other CAD systems. Export functionality was tested as well.

The advantages of a 3D CAD model could easily be evaluated by introducing design changes (“add 3m in length and fill it with additional cabins of type xy”).

The results of the evaluation have been documented in a jointly established report and a recommendation for the new product Unigraphics NX from EDS was agreed.

5.2. The CAD Rollout

The rollout of the new CAD system started in February 2003 according to a detailed project plan. The project plan describes at a fairly detailed level the sequence of administrator and user training courses, the establishment of design rules and - most important - it defines work packages of the German Corvette Program K130 which will be performed by groups of the 3 yards in close cooperation. Hence some important parts of the K130 design are used as a pilot project for the use of Unigraphics NX and will closely be monitored. Consultancy will be available for this “training on the job” phase which is considered essential for getting experience on the coupling of “geometrical data” (CAD model) to product information inside the PDM.

6. Conclusion

The increasing complexity of modern vessels as well as the fierce competition on the world market are the main driving factors for a higher level of tool integration in the shipbuilding industry. IT technology and IT solutions available today allow already a reasonable amount of “collaborative / concurrent engineering”. Rather than waiting for a perfect standardization of “data exchange formats” or the announcement of the “all-in-one solution”, a careful analysis of current processes and a pragmatic approach with existing modular IT systems can increase the efficiency of shipyard processes today. However, the effort of successfully integrating those systems into the daily work at a shipyard should not be underestimated. The term “change management” in its modern meaning is at least a slight understatement for the start-up into a world of transparent workflows, integrated product teams, and information, which is always up-to-date and available via a simple web browser.
Index by authors

<table>
<thead>
<tr>
<th>Author</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abt</td>
<td>444</td>
</tr>
<tr>
<td>Akinfiev</td>
<td>516</td>
</tr>
<tr>
<td>Armada</td>
<td>516</td>
</tr>
<tr>
<td>Baier</td>
<td>278</td>
</tr>
<tr>
<td>Bennis</td>
<td>162</td>
</tr>
<tr>
<td>Bentin</td>
<td>118</td>
</tr>
<tr>
<td>Bertram</td>
<td>5, 235, 336</td>
</tr>
<tr>
<td>Blackshields</td>
<td>55</td>
</tr>
<tr>
<td>Bronsart</td>
<td>415</td>
</tr>
<tr>
<td>Cain</td>
<td>284</td>
</tr>
<tr>
<td>Cantow</td>
<td>415</td>
</tr>
<tr>
<td>Coenen</td>
<td>490</td>
</tr>
<tr>
<td>Chardard</td>
<td>425</td>
</tr>
<tr>
<td>Davidson</td>
<td>377</td>
</tr>
<tr>
<td>Degtyarev</td>
<td>48</td>
</tr>
<tr>
<td>De Jonge</td>
<td>196</td>
</tr>
<tr>
<td>Dirix</td>
<td>308</td>
</tr>
<tr>
<td>Duvigneau</td>
<td>154</td>
</tr>
<tr>
<td>Elsässer</td>
<td>284</td>
</tr>
<tr>
<td>Erikstad</td>
<td>255</td>
</tr>
<tr>
<td>Everett</td>
<td>70</td>
</tr>
<tr>
<td>Ewer</td>
<td>55</td>
</tr>
<tr>
<td>Fernandez</td>
<td>516</td>
</tr>
<tr>
<td>Furstenberg</td>
<td>235</td>
</tr>
<tr>
<td>Galea</td>
<td>55</td>
</tr>
<tr>
<td>Gavrilescu</td>
<td>387</td>
</tr>
<tr>
<td>Giassi</td>
<td>162</td>
</tr>
<tr>
<td>Gonzalez</td>
<td>265</td>
</tr>
<tr>
<td>Gonzalez de Santos</td>
<td>516</td>
</tr>
<tr>
<td>Grau</td>
<td>479</td>
</tr>
<tr>
<td>Grimdelius</td>
<td>308, 502</td>
</tr>
<tr>
<td>Guedes Soares</td>
<td>183</td>
</tr>
<tr>
<td>Gurrea</td>
<td>265</td>
</tr>
<tr>
<td>Gwynne</td>
<td>55</td>
</tr>
<tr>
<td>Hagen</td>
<td>255</td>
</tr>
<tr>
<td>Hamada</td>
<td>118</td>
</tr>
<tr>
<td>Harries</td>
<td>444</td>
</tr>
<tr>
<td>Heimann</td>
<td>444</td>
</tr>
<tr>
<td>Hellesoy</td>
<td>84</td>
</tr>
<tr>
<td>Hellström</td>
<td>321</td>
</tr>
<tr>
<td>Henesey</td>
<td>377</td>
</tr>
<tr>
<td>Hurst</td>
<td>55</td>
</tr>
<tr>
<td>Jonas</td>
<td>387</td>
</tr>
<tr>
<td>Jaramillo</td>
<td>133</td>
</tr>
<tr>
<td>Kalathas</td>
<td>235</td>
</tr>
<tr>
<td>Kemp</td>
<td>210</td>
</tr>
<tr>
<td>Kim</td>
<td>534</td>
</tr>
<tr>
<td>Kirukhin</td>
<td>48</td>
</tr>
<tr>
<td>Kitamura</td>
<td>118</td>
</tr>
<tr>
<td>Klehn</td>
<td>247</td>
</tr>
<tr>
<td>Koelman</td>
<td>140</td>
</tr>
<tr>
<td>Koushan</td>
<td>472</td>
</tr>
<tr>
<td>Krüger</td>
<td>525</td>
</tr>
<tr>
<td>Lawrence</td>
<td>55</td>
</tr>
<tr>
<td>Lebkowski</td>
<td>175</td>
</tr>
<tr>
<td>Lee</td>
<td>534</td>
</tr>
<tr>
<td>Ligtelijn</td>
<td>308</td>
</tr>
<tr>
<td>Lindenau</td>
<td>5</td>
</tr>
<tr>
<td>Maisonneauve</td>
<td>162</td>
</tr>
<tr>
<td>Maney</td>
<td>70</td>
</tr>
<tr>
<td>Mawhinney</td>
<td>55</td>
</tr>
<tr>
<td>Meyer-König</td>
<td>133</td>
</tr>
<tr>
<td>Moreira</td>
<td>183</td>
</tr>
<tr>
<td>Moulijn</td>
<td>308</td>
</tr>
<tr>
<td>Mutu</td>
<td>387</td>
</tr>
<tr>
<td>Nechaev</td>
<td>48</td>
</tr>
<tr>
<td>Neumann</td>
<td>297</td>
</tr>
<tr>
<td>Nienhuis</td>
<td>490</td>
</tr>
<tr>
<td>Parsons</td>
<td>220</td>
</tr>
<tr>
<td>Petkov</td>
<td>415</td>
</tr>
<tr>
<td>Pietrzykowski</td>
<td>40</td>
</tr>
<tr>
<td>Pradillon</td>
<td>108</td>
</tr>
<tr>
<td>Rigo</td>
<td>94</td>
</tr>
<tr>
<td>Rijke</td>
<td>490</td>
</tr>
<tr>
<td>Roach</td>
<td>210</td>
</tr>
<tr>
<td>Rodriguez</td>
<td>265</td>
</tr>
<tr>
<td>Saanen</td>
<td>15</td>
</tr>
<tr>
<td>Santos</td>
<td>432</td>
</tr>
<tr>
<td>Sasaki</td>
<td>362</td>
</tr>
<tr>
<td>Schom</td>
<td>425</td>
</tr>
<tr>
<td>Shin</td>
<td>534</td>
</tr>
<tr>
<td>Siksne-Pedersen</td>
<td>29</td>
</tr>
<tr>
<td>Simpson</td>
<td>235</td>
</tr>
<tr>
<td>Singer</td>
<td>220</td>
</tr>
<tr>
<td>Smierchalski</td>
<td>175</td>
</tr>
<tr>
<td>Solano</td>
<td>265</td>
</tr>
<tr>
<td>Stäbler</td>
<td>479</td>
</tr>
<tr>
<td>Steinhauer</td>
<td>203</td>
</tr>
<tr>
<td>Tate</td>
<td>70</td>
</tr>
<tr>
<td>Tepper</td>
<td>196</td>
</tr>
<tr>
<td>Toderan</td>
<td>94</td>
</tr>
<tr>
<td>Toivonen</td>
<td>400</td>
</tr>
<tr>
<td>Turan</td>
<td>459</td>
</tr>
<tr>
<td>Türkmen</td>
<td>459</td>
</tr>
<tr>
<td>Uriasz</td>
<td>40</td>
</tr>
<tr>
<td>Van der Wagt</td>
<td>490</td>
</tr>
<tr>
<td>Van Es</td>
<td>350</td>
</tr>
<tr>
<td>Van Hees</td>
<td>350</td>
</tr>
<tr>
<td>Varela</td>
<td>432</td>
</tr>
<tr>
<td>Visonneau</td>
<td>154</td>
</tr>
<tr>
<td>Vogel</td>
<td>542</td>
</tr>
<tr>
<td>Wang</td>
<td>55</td>
</tr>
<tr>
<td>Wanner</td>
<td>297</td>
</tr>
<tr>
<td>Ware</td>
<td>210</td>
</tr>
<tr>
<td>Wauchope</td>
<td>70</td>
</tr>
<tr>
<td>Weidemann</td>
<td>297</td>
</tr>
<tr>
<td>Weiss</td>
<td>425</td>
</tr>
<tr>
<td>Wernstedt</td>
<td>377</td>
</tr>
<tr>
<td>Whittaker</td>
<td>284</td>
</tr>
<tr>
<td>Wilson</td>
<td>210</td>
</tr>
<tr>
<td>Winter</td>
<td>444</td>
</tr>
<tr>
<td>Woo</td>
<td>534</td>
</tr>
</tbody>
</table>
3rd International EuroConference on
Computer Applications and Information Technology in the Maritime Industries

COMPIT'04
Parador Siguéncia, Siguéncia, Spain, 9-12 May 2004

Topics:
- CAD / CAM / CIM / Simulations / Virtual Prototyping / Virtual Reality
- Robotics / Computer Aided Planning
- Information Technology Standards / Electronic Data Exchange / Net Technology
- Management Information Systems / Executive Information Systems
- Artificial Intelligence
- Management/Legal/Economical Aspects of Information Technology

In Shipbuilding, Offshore Engineering, Port and Ship Operation

Organisers:
Volker Bertram, Manuel Armada

Advisory Committee:
Manuel Armada
Klaus-Peter Beier
Ludwig Forstenberg
Zbigniew Pietrzykowski
Martha Grabowski
Kazuo Hiro Hasegawa
Gelu Kahu
Maurizio Landrini
IAI-CSIC, Spain
University of Michigan, USA
ITE, South Africa
MUS, Poland
Le Moyne College, USA
Osaka University, Japan
InfoBlas, Romania
INSEAN, Italy
Ehsan Mesbahi
Ivan Østvik
Berend Bohlmann
Yoram Reich
Mares Salas
Christophe Captant
Lantos Bela
University Newcastle, UK
LMG Marin, Norway
FSG, Germany
Tel Aviv University, Israel
University Austral, Chile
French Navy, France
Budapest Univ. of Technology

Venue:
The conference will be held at the Parador Siguéncia, a genuine castle dating back to medieval times, 130 km north-east of Madrid. Accommodation is available at the conference venue. On 13 May, there will be visit and walking/climbing robot demonstration at CSIC in Madrid.

Format:
Papers to the above topics are invited and will be selected by a selection committee. There will be hard-cover proceedings and papers may have up to 15 pages

Important dates:
- 30.09.2003 optional “early warning” of intent to submit paper
- 07.01.2004 abstract received deadline
- 10.01.2004 notification of acceptance
- 28.02.2004 final paper and payment due for authors

Fees:
580 Euro

Keynote lectures:
Dr. Manuel Armada "Walking and climbing robots in the maritime industry"
Further keynote lectures to be announced

Information: bertram@waves.insean.it
www.i.ai.csic.es/compit04

Sponsors:
To be found…