INNOVERO DESIGN SERVER APPLIED ON NAVAL SHIP DESIGN

A Bons, MARIN – MSG, Wageningen, The Netherlands
H. J. Koelman, SARC BV, Bussum, The Netherlands
B.J. v. Oers, NL MOD, Defence Materiel Organisation, The Hague, The Netherlands
E.H. Takken, NL MOD, Defence Materiel Organisation, The Hague , The Netherlands

SUMMARY

This paper reviews the state-of-the-art Innovero design platform by means of a conceptual naval ship design case. The platform is a result of the three-year Dutch Maritime Innovation Platform project Innovero, which improves conceptual ship design by enabling the early stage application of advanced prediction tools.

The major components of this system are discussed in this paper. The first is the Innovero Design Server, which enables concurrent use of design tools and knowledge in discipline-oriented knowledge systems or 'agents'. The second component is a method for the modelling of the internal geometry of the ship. This tool bridges the duality between volumes (spaces, compartments) and planes (bulkheads, decks).

Both components were tested by The Defence Materiel Organisation to enable early stage assessment of intact and damaged stability, a crucial design driver for warships. The application illustrates how it improves the quality of the conceptual design process and reduces time-to-market for new designs.

1. INTRODUCTION

Being sufficiently stable in both intact and damaged condition is important for every floating structure, from the smallest dinghy to the largest ships and offshore platforms. The prediction of stability therefore receives ample attention and is one of the ship performances naval architects monitor throughout the lifecycle, from concept design to engineering, production and operations.

Still, even though proper stability is important in every design phase, it warrants particular attention during early stage ship design. Erroneous or inaccurate stability predictions introduce considerable risk in a project, for several reasons. First, stability requirements are typically non-negotiable, i.e., they must be met in order to obtain regulatory approval. Also, owners have a legal and moral obligation to crew and society to ensure the safe operation of their ships. Second, addressing insufficient stability requires considerable changes to the ship's design, changes that are expensive to make once the ship reaches the engineering phase, or even worse, the production phase. Hence, there is a considerable need to conduct accurate stability predictions during early stage design.

Unfortunately, making accurate stability predictions early on is not straightforward for both commercial ships and warships. Tendering for a commercial contract imposes considerable time-constraints during which elaborate and time-consuming stability predictions are difficult to carry out. For warships, a similar problem arises for a different reason. Early stage warship design balances naval ambition and affordability by considering a wide variety of alternative ship designs in an iterative process. Hence, time to investigate a single design is limited, making it difficult to execute accurate stability predictions even though damage stability requirements - especially damage size- are considerably more stringent for warships.

This paper presents an approach to improve the design process by means of applying the Innovero Design Server. This platform has been developed in the Dutch Maritime Innovation Platform (MIP) Innovero research project (www.cmti.nl) carried out in The Netherlands from 2008 to 2011 and offers users increased performance on data consistency, project manageability, and enables concurrent engineering [5]. The need for the design server is emphasised by the stability use case, which discusses the integration of a bespoke early stage ship design model GCD^2 [8] and QPIAS with proven stability prediction software PIAS (www.sarc.nl). Although this latter approach offers improvements on design speed, data consistency is not yet guaranteed. Within the Innovero context it was recognised that a promising opportunity for further improvement lay in the development of a method capable of converting a space-based ship compartment representation into a plane-based representation, and vice versa. Both representations are required, dependent of the different predictions tools used for the development of feasible concept designs.

The paper is structured in 7 sections. Section 2 establishes present problems with the usage of PIAS in the warship design process as applied by the Defence Materiel Organisation (DMO). Section 3 proposes three improvements which are subsequently elaborated on in detail in sections 4 and 6. Section 5 illustrates possibilities of the improved approach by showing results from an application to an early stage frigate design performed by Defence Materiel Organisation before Section 7 discusses the results and draws conclusions.

2. PRESENT STATUS OF DAMAGE STABILITY ASSESSMENT IN PIAS

Starting point for the improvements is the current workflow used to predict the intact and damaged stability. This workflow consists of three sequential steps briefly outlined below.

- First, stability prediction in intact and damaged condition requires the availability of a ship design defined in considerable detail. At DMO, balanced concept designs are generated by using GCD^2, which allows arriving at a balanced solution within a few working days and provides the majority of the required data for stability assessment.
- Second, stability predictions in PIAS require, as usual, a considerable amount of input, amongst others a geometrical description of hull , the definition of watertight compartments, their permeability, the location of openings, connections between compartments, loading conditions and lightship weight. Currently, all input is derived from the design model manually and transferred manually to PIAS.
- Third, the actual stability prediction is carried out in PIAS, leading to the familiar results.

As this process does not guarantee compliance to requirements, these labour intensive steps need to be traversed multiple times in the iterative ship design approach. Therefore, the manual approach represents a serious time investment to adapt the design in such a way that intact and damage stability requirements are met and introduces the risk of unacceptable inconsistencies and errors. Hence, improving the speed and quality of data-transfer was identified as crucial to improve the early stage design process.

3. PROPOSED PROCESS IMPROVEMENT

At DMO, balanced concept designs are generated by using several advanced prediction tools, such as 3D CAD programs, databases, spreadsheet, rules and formulas. GCD^2, developed to streamline this complex design process, uses MARIN's knowledge management framework Quaestor [4] to encapsulate these tools in a knowledge base and to execute calculations. Making Quaestor responsible for information exchange between design and analysis tools saves time and reduces the chance on errors.

DMO continuously improves early stage predictions by introducing new model fragments into their existing knowledge bases like GCD^2. One of the latest improvements is a newly developed knowledge base QPIAS around the stability software PIAS to assess damage stability in the early design stage. The QPIAS workflow system provides guidance and ease-of-use for complete chains of pre-processing, calculations and post-processing.

Because a GCD^2-model already contains a considerable amount of design data that can be used as input for QPIAS, integrating these two models is desirable. The interface between the knowledge bases QPIAS and GCD^2 can be controlled by the "Innovero Design Server", a new capability of the Quaestor framework as developed within the scope of the Innovero project. Innovero aims to improve the early design process in such way that existing advanced design tools can be applied already in early stages of design. The most important feature of the Design Server to DMO is its capability to combine multiple workflows on various Quaestor clients into a single 'super project' on the Design Server, while maintaining data consistency between these clients.

4. THE INNOVERO DESIGN SERVER

In early ship design, tasks are performed by using a variety of computer applications that are not always available on the same computer or even within the same organization. Often multiple specialists are involved in the design process using their proprietary tools. However, integrating all necessary tools needed into one single system is unattractive from a maintenance and reusability point of view. Therefore, DMO decided to develop a separate Quaestor based application QPIAS to perform the stability assessment task, instead of integrating this task within the existing GCD^2 design model. Distributing tasks over different applications has two main advantages. First, using different applications enables concurrent design and engineering between geographically distributed team members. Second, the maintenance of different design disciplines can be divided over several domain experts. Hence, distributing different tasks over different applications avoids extensive and inflexible design and engineering suites, which are hard to maintain.

Already for decades it is recognized that those different environments should preferably be interoperable. Therefore, in 2007, the Dutch research project Innovero, of the MIP, was initiated. First the backgrounds of the basic Quaestor concepts will be briefly discussed to provide more understanding on the present behaviour and possibilities of QPIAS, GCD^2 and the Innovero Design Server.

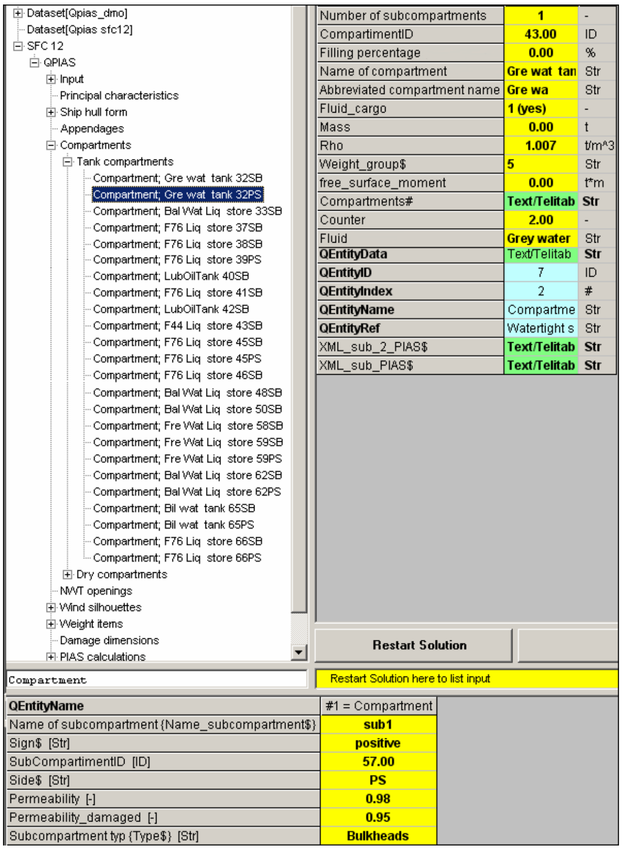
4.1 ABOUT QUAESTOR

The knowledge management system Quaestor is a computational modelling system applied in a variety of design and analysis applications in the shipbuilding and offshore industry. It is an integration platform to develop and use knowledge bases containing computational methods and data, which enable research, engineering and design departments to simplify the tasks of:

- Streamlining of complex analysis, design and engineering processes.
- Performing goal driven design, engineering and mathematical model development by using the powerful Quaestor reasoning mechanism.
- Developing workflows around existing software tools and integrating them as they are.
- Data and calculation management; all information from project data to calculation results and reports in one environment.

In addition, a developer of a knowledge base, a knowledge engineer, can create a workflow of a process (e.g. the design of a ship hull or the generation of a stability booklet) or product (e.g. ship structure), by means of the taxonomy-entity methodology [2, 5]. Taxonomy is an hierarchy of entities, each representing an object that contain data, goals and model fragments that are only applicable to that specific object. In other words, a taxonomy is the blue print of the process or product.

Moreover, it is possible to create complex applications which capture the workflow around existing design tools. As users are not controlling the execution of software by themselves anymore, because Quaestor will execute the software, they expect an evident sequence of process steps, more or less according the way they were used to. This can be realized by the knowledge engineer in the form of a taxonomy-based solution. As example the taxonomy of QPIAS is shown in figure 1.



```
⊞ Dataset[Qpias_dmo]
  Dataset[Qpias sfc12]
⊟ SFC 12
  ⊟ QPIAS
    ⊞ Input
      Principal characteristics
    ⊞ Ship hull form
      Appendages
    ⊟ Compartments
      ⊟ Tank compartments
          Compartment; Gre wat  tank 32SB
          Compartment; Gre wat  tank 32PS
          Compartment; Bal Wat Liq  store 33SB
          Compartment; F76 Liq  store 37SB
          Compartment; F76 Liq  store 38SB
          Compartment; F76 Liq  store 39PS
          Compartment; LubOilTank 40SB
          Compartment; F76 Liq  store 41SB
          Compartment; LubOilTank 42SB
          Compartment; F44 Liq  store 43SB
          Compartment; F76 Liq  store 45SB
          Compartment; F76 Liq  store 45PS
          Compartment; F76 Liq  store 46SB
          Compartment; Bal Wat Liq  store 48SB
          Compartment; Bal Wat Liq  store 50SB
          Compartment; Fre Wat Liq  store 58SB
          Compartment; Fre Wat Liq  store 59SB
          Compartment; Fre Wat Liq  store 59PS
          Compartment; Bal Wat Liq  store 62SB
          Compartment; Bal Wat Liq  store 62PS
          Compartment; Bil wat  tank 65SB
          Compartment; Bil wat  tank 65PS
          Compartment; F76 Liq  store 66SB
          Compartment; F76 Liq  store 66PS
      ⊞ Dry compartments
      NWT openings
    ⊞ Wind silhouettes
    ⊞ Weight items
      Damage dimensions
    ⊞ PIAS calculations
```

Number of subcompartments	**1**	-
CompartimentID	**43.00**	ID
Filling percentage	**0.00**	%
Name of compartment	**Gre wat tan**	Str
Abbreviated compartment name	**Gre wa**	Str
Fluid_cargo	**1 (yes)**	-
Mass	**0.00**	t
Rho	**1.007**	t/m^3
Weight_group$	**5**	Str
free_surface_moment	**0.00**	t*m
Compartments#	**Text/Telitab**	**Str**
Counter	**2.00**	-
Fluid	**Grey water**	Str
QEntityData	Text/Telitab	**Str**
QEntityID	7	ID
QEntityIndex	2	#
QEntityName	Compartme	Str
QEntityRef	Watertight s	Str
XML_sub_2_PIAS$	**Text/Telitab**	**Str**
XML_sub_PIAS$	**Text/Telitab**	**Str**

Compartment

Restart Solution here to list input

QEntityName	#1 = Compartment
Name of subcompartment {Name_subcompartment$}	**sub1**
Sign$ [Str]	**positive**
SubCompartimentID [ID]	**57.00**
Side$ [Str]	**PS**
Permeability [-]	**0.98**
Permeability_damaged [-]	**0.95**
Subcompartment typ {Type$} [Str]	**Bulkheads**

Figure 1: Screenshot of the QPIAS taxonomy.

The QPIAS workflow clearly distinguishes three process steps of pre-processing, calculations and post-processing. The "propagation engine" of Quaestor [5] is responsible for the data exchange between different entities and the propagation of input changes in parameter values through all calculations. Therefore, the consistency of data exchange is automatically maintained by Quaestor. For example, by changing the waterline length of a ship design in QPIAS, Quaestor automatically provides the user feedback by means of colours about which entities (representing process steps, calculations etc.) need to be recalculated. In this case the interfaced program PIAS will be executed again. The automated file format conversions and re-entering of input for each program prevents typing errors, saves time and ensures recalculation of dependent values.

Although Quaestor is used as an environment for streamlining design and engineering processes of DMO, the philosophy of DMO and MARIN remains that:

- The involvement of smart engineers will remain necessary for better designs: keep the specialist in control.
- Focus should be on optimization of information flow and execution order of analysis steps: spend less time on routine work and take more time for analysis and creative solutions.
- Focus on easy use of the expertise, while the responsibility of (knowledge) development remains with the expert(s).
- Make use of existing software, so domain experts can still maintain and further develop their own specialized tools and even more important, engineers still can work in the GUI of the software they are familiar with.

4.2 INTRODUCTION OF THE DESIGN SERVER

In an early or pre-contractual design stage, the ship designer's ability to concurrently apply design and simulation tools from different disciplines is limited. The Innovero Design Server provides an innovative conceptual ship design platform for use in a distributed multi-disciplinary environment consisting of a number of discipline-oriented knowledge systems or 'agents'. This paper reviews the state of the of the art of the Innovero Design Server by means of the cooperation between the agents GCD^2 and QPIAS, as schematic represented in figure 2.

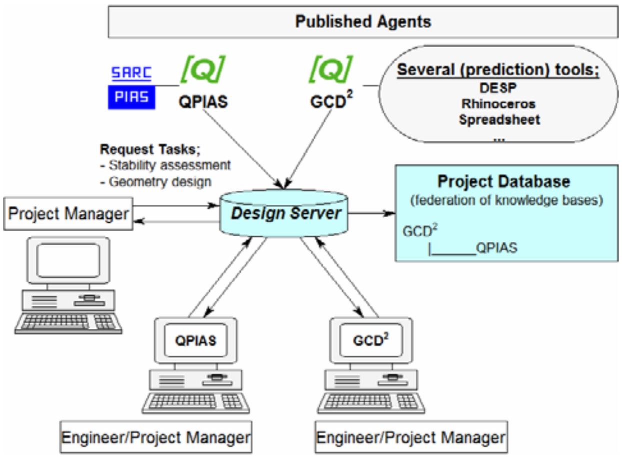


Figure 2: Cooperation between agents QPIAS and GCD^2 by mean of Design Server.

4.3 SET UP A DESIGN SERVER

The conceptual basis for the Innovero Design Server is the Taxonomy-Entity concept introduced above. In short, the Design Server has the following responsibilities:

- Knows the published (taxonomies of) agents.
- Knows the dependencies between the taxonomies of the published agents.
- Knows the members of the team of a combined project.
- Starts a project by opening the Project Database and contains the total solution.
- Knows and communicates about the status of a solution.

First the knowledge bases QPIAS and GCD^2, the agents, have to be published on the Design Server. After publishing, the list of parameters and the taxonomy's of both knowledge bases are added to the Design Server. Furthermore, the Design Server knows the manager, location and possible users of an agent.

Although it is possible to use both agents autonomous, dependencies between the agents can be created in two ways, namely hard-coded and with dynamic linking. A knowledge engineer can use a hard coded relation if he knows in advance that a value of a parameter in one agent is always equal to a parameter value in another agent. For instance, within the QPIAS taxonomy, a hard coded relation can be used for the calculation of parameter “Lwl”, by referring to a parameter “LengthWaterLine” in entity “main dimensions” of the agent GCD^2. Please note, using identical parameter names is not a necessity for linking parameters between different agents. However, as result of the Innovero project, there is a need for being able to link parameters to uniformly described parameters in a “dictionary” in support of knowledge engineers. In our example, the knowledge engineer of QPIAS had to find out that “Lwl” and “LengthWaterLine” are describing the same design variable.
The second way to realize dependencies between agents is through dynamic linking. Instead of a knowledge engineer determining a link between parameters, a user of an agent (designer or engineer) selects a parameter on the Design Server, which determines the value of a certain parameter.

4.4 COMBINED “SUPER” PROJECT ON THE DESIGN SERVER

How a combined project is carried out on the Design Server will be explained by means of the following simplified example of a frigate design project. In this example, it is assumed that only the main particulars of GCD^2 are required in QPIAS, which is arranged by hard coded relations.

A project manager creates a new combined project, e.g. by the name of *pr_frigate_design* on the Design Server and selects one or more available tasks. Each task corresponds with one of the published taxonomies, and has to be performed by one of the analysts that are marked as users of the corresponding agent (knowledge system). In our case, the tasks QPIAS (for damaged stability assessment) and GCD^2 (geometric design, weight estimate, resistance and propulsion predictions, etc.) are selected. The Design Server will sent requests to the engineers of the selected agents GCD^2 and QPIAS, to carry out a specific task for the project *pr_frigate_design*. Please note, a project manager of the combined project can also carry out one or more of the selected tasks by himself.
Next, the engineers of GCD^2 and QPIAS have to accept the requests and start performing their task as far as possible. The engineer of QPIAS is able to perform his task until he needs the main particulars of GCD^2. Meanwhile, the engineer of QPIAS can publish his intermediate results on the design server. Published results are visible for all team members of the combined project. At the moment the engineer of GCD^2 publishes his results on the Design Server, including the main particulars, the engineer of QPIAS will receive a message that the values for the main particulars are available, as calculated by agent GCD^2. Once available, the engineer of QPIAS is also able to finish his task and publish his end results on the Design Server.
The status of the total solution of the project *pr_frigate_design* is presented on the Design Server and is visible for all team members. In case the engineer of GCD^2 decides to change the geometry of the frigate design and publishes changed main particulars, the Design Server knows which other agents will be influenced. In our example the engineer of QPIAS will receive a message from the Design Server, that updated values for the main particulars are available. It is up the engineer of QPIAS to accept these updated values or not. If not, inconsistency will occur within the project *pr_frigate_design*. The Design Server informs the project

manager of the *pr_frigate_design* project, about the inconsistency by means of colour feedback. Hence, the client (the engineer of QPIAS) is responsible for the propagation of data and not the Design Server, which is only the messenger.

To summarize, the Innovero Design Server informs the project team members during the project about pending design tasks and tasks to be repeated due to changes made in any of the tasks of the project. The platform takes care of data exchange between design tasks and software tools, maintains data consistency and propagates changes in parameter values through all calculations in the project and keeps track of these changes.

The methodology and workflow described above have been demonstrated in a practical application, described in the section below. Please note that this proof of concept focuses on getting the process to work rather than implementing any and all developed innovations.

5. COOPERATION DESIGN TOOLS; GCD^2 – QPIAS – PIAS

The job of a concept designer is to demonstrate the consequences of design decisions on performance and cost. Therefore, one must integrate performance predictions in an efficient manner, a considerable challenge, which at DMO has been met by applying GCD^2. The aim of this design tool is to provide the designer with an integrated model capable of producing feasible concept designs within a few working days. As it is developed in Quaestor, use of the available interfacing possibilities with existing performance prediction programs could be made. This lead to an efficient and adaptable application with numerical representation handled in Quaestor and MS Excel™ and graphical representation handled in 3D-application Rhinoceros® (www.rhino3d.com).

Within GCD^2, all spaces on the design are defined as functional volumes blocks, which are developed from the building block methodology [1]. These blocks represents amongst others, space and weight reservations. A few exemptions are made where a surface description is strictly required, e.g. to describe a helicopter deck or to model required deck strengthening to support a sensor mast or a crane. This approach ensures easy and fast concept generation, alteration and analyzing, although no direct physical model description exists.

In order to facilitate usage of a volumetric, functional vessel description in stability computation in PIAS, a work flow as depicted in figure 3 and discussed below, has been developed.

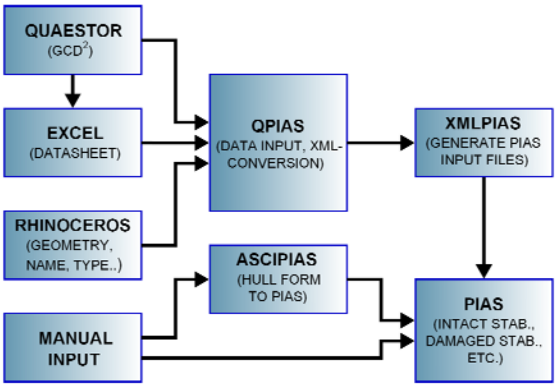


Figure 3: Work flow diagram of exporting GCD^2 data to PIAS.

As a functional volume block has no direct physical significance, it may stretch over multiple water-tight compartments or, alternatively, be merely a part of a water-tight compartment. Nevertheless, the volumetric model description largely corresponds to the sub-compartment description used in PIAS, although the description in GCD^2 cannot be used one-to-one in PIAS. This is due to the fact that GCD^2 requires a volumetrically and geometrically correct defined block, whereas PIAS needs a block with 8 corner points. For this reason, visualization scripts have been adjusted to automatically create blocks meeting GCD^2 as well as PIAS demands. Figure 4 depicts a conceptual design in GCD^2 functional blocks on the left and in PIAS blocks on the right, where you will notice that the blocks overlap the water-tight hull completely.

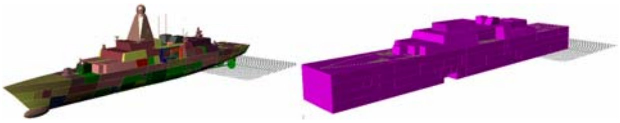


Figure 4: Conceptual design of a frigate. Left representation in GCD^2 functional building blocks and right blocks required for export to PIAS.

Besides the geometrical information, data on the general dimensions of the ship, its lightweight and its deadweight needs to be transferred from GCD^2 to PIAS. Deadweight information is gathered in GCD^2 for the fully loaded condition, which later will serve as starting point to manually define other loading conditions in PIAS, and transferred to a MS Excel™ sheet. Furthermore, the hull form of the vessel needs to be exported to PIAS. This is done by using a script which translates the hull sections to a point cloud. This point cloud is then translated to a PIAS hull form by a module of PIAS named ASCIPIAS. Figure 5 is a visualization of this process.

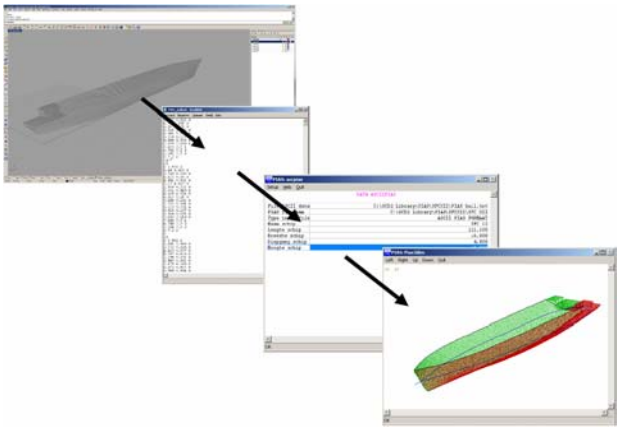


Figure 5: Exporting hull form data to PIAS.

As the script does not correct errors in the hull definition, once imported, it needs to be manually checked on consistency, such as the definition of double frames, the definition of the deck line, and the hull needs to be closed.

QPIAS – of which a taxonomy screenshot is shown in figure 1 - is used to import weight, loading, and vessel characteristics data from the GCD^2 datasheet, which ensures data consistency between the information used to create the conceptual design in GCD^2 and the information used to compute the intact and damaged stability of the vessel. Furthermore, design characteristics, lightweight and deadweight data is imported through earlier mentioned MS Excel™ sheet.

The volumetric description of blocks in GCD^2 and the differences in visualization needed to enable PIAS to use that data have been addressed earlier. For this proof of concept, all blocks are defined in such a way, that no blocks stretches over more than a single water-tight compartment. By applying a script, the geometrical information in Rhinoceros® is translated to 8 corner points PIAS requires. As PIAS cannot cope with negative y-positions, the script is set up in such a way that blocks symmetrical to the centre line are split into PIAS-double blocks and blocks asymmetrical to the centre line are split into two separate blocks, one defined on the starboard and one on the port side. Reading directly from the 3D application Rhinoceros® allows up-dating GCD^2 without the need of associated adaptations in QPIAS each time.

At present, the information is retrieved by QPIAS from template MS Excel™ sheets which previously are filled in by GCD^2. However, to improve data consistency, a design server should replace the exchange of data in the future, as demonstrated in Innovero. Once the necessary information has been imported, QPIAS begins to prepare the data export to PIAS. This is done by translating received data, such as weight items or main dimension, to XML-files readable for PIAS. Next, the location of the earlier made PIAS-hull file is defined. Then, the geometrical information defining the compartments are transferred to XML-files. During this step, the imported geometrical information is filtered to discriminate between tank compartments and dry compartments. This is done by combining data imported from the datasheet, which lists tank block types, and the information provided in the 3D model. This makes it possible to combine compartmentation data with geometrical hull form data.

Subsequently, a PIAS module named XMLPIAS is used to converse the above mentioned XML-files to PIAS-files and update the PIAS settings file. The input for the calculations is prepared and the heeling angles are defined. At present, the user needs to define these values manually, however, they should be imported from GCD^2, as it is available information. Also, QPIAS requires the user to define which stability criteria need to be employed. The criteria can be selected from a simple drop-down menu or be defined manually.

At this point in time, a PIAS calculation can be made. At present, QPIAS automatically computes hydrostatic data for the design draught as well as a tank list and exports the results to a MS Word™ design report. This step is important since it enables creating a tank filling list or to compute the damage stability. It is important to stress, that all compartments are now computed, both tanks and dry compartments. On a side note, it needs to be stated that all imported blocks are regarded as individual compartments. Therefore, the exported list needs to be manually processed in PIAS by combining all relevant sub-compartments to an actual watertight compartmentation. An example of PIAS and GCD^2 compartmentation is given in figure 6. PIAS provides the user with a “merge”-tool that allows combining these sub-compartments in a fast and simple way, although at present, it remains manual labour done by the designer.

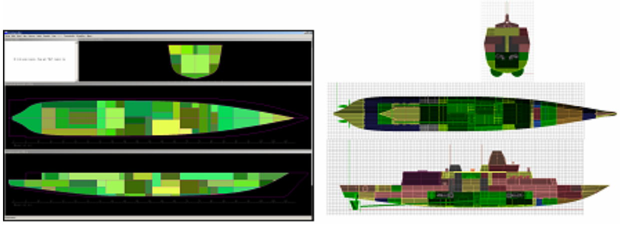


Figure 6: PIAS screenshot (left) and corresponding GCD^2 (right) visualization.

As the design is available in PIAS, the user no longer needs to run QPIAS, instead he can access the design directly from the PIAS interface and create different loading conditions or determine the stability of the vessel in intact or damaged condition. However, by applying QPIAS, the work load and overturn time of transferring a GCD^2 conceptual design to PIAS has been significantly reduced. This is due to the fact that compartmentalization is defined only once. A direct link between the concept design, its 3D representation and PIAS enabled increasing data consistency.

Once a concept design has been elaborated satisfactorily, a more detailed description of the vessel is required. This is done to carry out more performance analyses at a higher level of detail. To do so, a translation from volumetric design description to surface description is required. A promising way of doing so is by applying the Binary Space Partitioning method.

6. A NOVEL SUBDIVISION METHOD

6.1 THE BACKGROUND

Although the method described in section 5 is a significant process improvement, there was room for further improvement. As already indicated in that section, in data exchange matters on the internal subdivision of ships the duality between spaces and planes should be addressed. Consequently, within the Innovero context it was decided to develop a method which is capable to convert a space-based compartment representation into a plane-based fashion, and vice versa. Additional design requirements for such a method where:

- Compatible with (or convertible into a format compatible with) the applied naval architectural analysis software. If representation conversion is required, it should not lead to over-detailed models, because they could hamper calculation efficiency in case of lengthy calculations, such as damage stability.
- Ready to be applied in combination with various ship hull representations, e.g. a surface model, a solid model, or a wireframe model. The latter may even be rather sparse, if the hull is only defined upon cross sections.
- The applied method should be, as such, easy to understand by a ship designer and consequently be user-friendly. Underlying entities should be sufficiently easy to understand, and comprehensive, in order to be utilised with macros or scripts of general-purpose CAD software.

For this task several alternative modelling methods could be applied, an overview and a discussion of their merits is presented in [6]. Because the aim was set at an understandable underlying concept, from the options we have selected a space partitioning method. In particular in discussion in a ship-designer forum it became apparent that the 'space splitting idea' is considered to be rather intuitive; with this method an empty hull form is split in two by a plane, those two resulting spaces are subsequently split in two by other planes etc. etc., until the subdivision is obtained. If we look beyond its particular fields of implementation, this 'space splitting idea' is similar to the well-known Binary Space Partitioning (BSP) method, where a space is recursively split in two, resulting in closed cells, in which we see the ship's compartments. The BSP method (see http://en.wikipedia.org/wiki/Binary_space_partitioning for an introduction) originates from interactive computer games , but has also been used for modelling purposes.

Conventionally, the recursive subdivision is represented by a binary tree, which is also a suitable internal representation in a computer program. An example of a BSP-application in a plane is given in figure 7, where the shaded 2D figure on the right is recursively split by the planes a through f (which form the nodes of the tree at the right side of the figure) and the cells 1 through 7 (which form the leaves of the tree). In three dimensions, the splitting idea is equal, only now applied to the space instead of the plane.

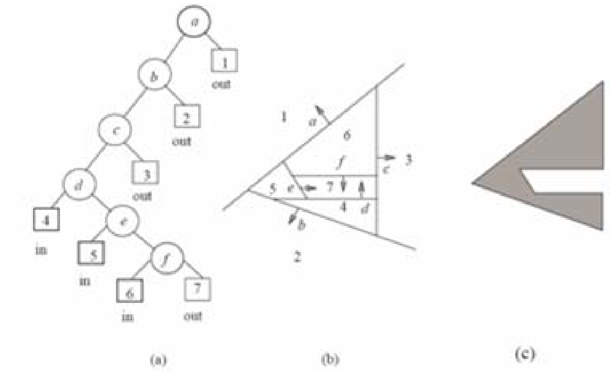


Figure 7: Two-dimensional example of the BSP tree (from [3]).

6.2 THE IMPLEMENTATION

We may conclude that the BSP approach may fit our requirements; it is capable of representing planes as well as volumes, supports the intuitive method of 'space splitting', and, most important, is conceptually simple. However, a native BSP representation would not always be the best entity to present the ship layout to the ship designer, for the reason that in a BSP tree a compartment or plane may be subdivided in many smaller sub-compartments or sub-planes, which hampers the grand overview of the design. For that reason, a data structure was designed where the program user, the ship designer, is working with the following familiar entities:

- The 'compartment', which is an enclosed space within the ship.
- The 'physical plane', which is a realistic plane within the ship, so a bulkhead or a deck. The physical plane may be bounded, which means it does not extend over the entire space in the ship hull.
- The reference plane, which is a virtual and unbounded plane, only intended to speed up modelling and modification action.

The BSP provides the 'glue' between those three entities, and is not available to the program user as a separate entity. This data structure is depicted in some more detail in figure 8.

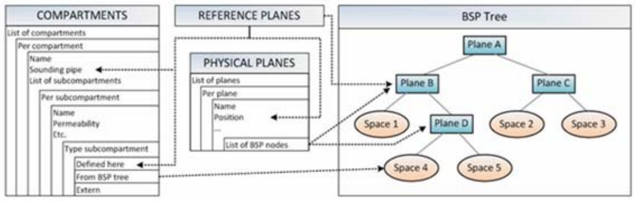


Figure 8: BSP tree as central representation for the compartment design.

Based on this design a computer program, a new module of the PIAS suite, was developed, which is aimed to play a central role in the Innovero context. For a list of program capabilities, and other details, the reader is again referred to [3]. In figure 9 program screenshots are presented (in this stage of development still in Dutch).

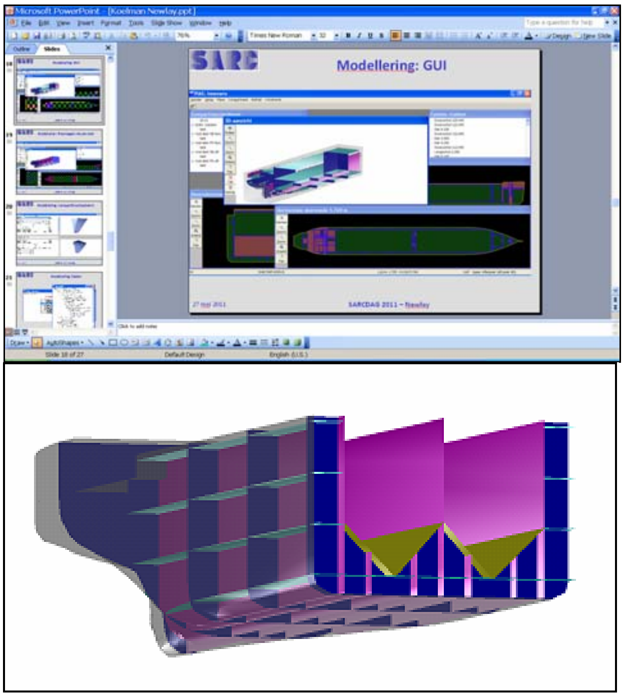


Figure 9: Screenshots of BSP-based PIAS module.

6.3 THE INTEGRATION INTO THE INNOVERO STRUCTURE

This new compartment representation and design program is capable to act as a stand-alone (PC-based) program, however, it is also ready to play the role of a subdivision shape and layout server in combination with the Innovero Design Server, which was discussed in section 4. With this combination the authors believe to have developed an efficient design framework and management tool for ship designs in the future of DMO.

Preliminary results indicate that the BSP could support and smoothen the integration of design steps at various levels of detail. By being capable of translating a volume to a surface representation and vice versa, more detail can be added easily, thereby improving scalability of the design tools and enabling the increase of result reliability. Furthermore, the integration of additional performance prediction tools, which are currently only manually accessible, is prepared. An overview is given in [7]. An example of DMO of a translation from a volume to a surface representation is given in figure 10.

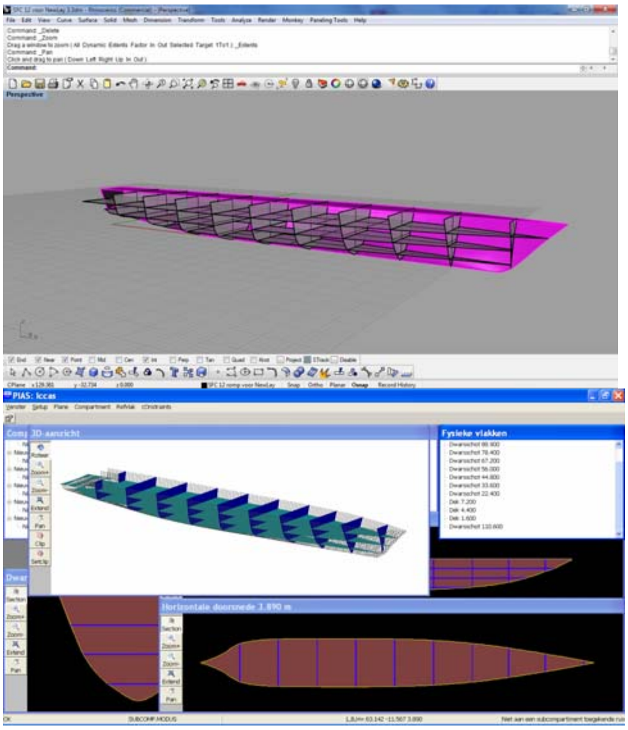


Figure 10: Example of a volume to surface conversion.

7. CONCLUSIONS

The aim of the work, reflected in this paper, was to reduce the amount of work involved in, and to improve the consistency of, some elementary aspects of the design of a ship, notably the hull, the subdivision, and the dependant analyses, such as intact stability and damage stability. It was concluded that the previous *modus operandi* involved quite some manual labour, and implied the probability of manual errors. An improved method was presented, which relied on a knowledge management system, Quaestor, to play a central role in the management and transfer of data. However, gradually it became clear that on the modelling side a gap had to be bridged; the duality between planes and spaces (of the internal subdivision of the ship). Subsequently, a BSP-based method was proposed to address this issue, and a provisional implementation proved to be promising. Finally, a first result of a preliminary implementation of all discussed tools and methods was presented.

In this work, the potential of the proposed method is illustrated and demonstrated. However, for an industrial implementation, some issues still have to be addressed or worked out in more detail:

- Extent the data IO; where this is currently limited to planes and property-lacking spaces, all relevant compartment or space properties should be included.
- Extent the analysis-suite with more functions, such as vulnerability and signature (for naval ships) and e.g. tonnage and freeboard calculations (for commercial vessels). Such tools are readily available, however, they should be included in the proposed infrastructure.
- Investigate the option to virtualise the CAD-system; currently the system is hard-coded linked to Rhinoceros, however, it could be beneficial to loosen this link, so that other CAD-systems could be integrated with relatively little effort.
- Improve the data transfer method; currently, in the experimental set-up it is file-based, while a network-based, or internet-based method would prevail. By the way, this subject was already addressed in *Innovero*, but a working implementation has not yet been produced.
- Integration of results of the translation from a volume to surface based design representation in the design process. This translation and its consequences require addressing many naval architectural challenges. One could think of adjusted weight estimation methods on base of the more detailed surface representation or the addition of more performance analysis tools.

8. ACKNOWLEDGEMENTS

The work presented in this paper is a result of the Innovero research project. The project was a collaboration of Conoship International, Damen Research, Damen Schelde Naval Shipbuilding, Imtech, Defence Materiel Organisation, MARIN, SARC, Delft University of Technology and Whale Informatics, directed by CMTI. The authors would like to thank our colleagues from the Innovero participants, with special thanks to Robin Brouwer (DMO), Martin van Hees (MARIN), Robert Hekkenberg (DUT) and Marcel Visser (DMO) for their support and expertise.

9. REFERENCES

1. ANDREWS, D., 'A creative approach to ship architecture', 2003.

2. BONS, A., 'QSHIP; Advanced Use of Hydrodynamics in Early Design', *ICCAS 2009 the 14th International Conference on Computer Applications in Shipbuilding, Shanghai, China,* 2009.

3. COMBA, J. L. D., NAYLOR, B. F., 'Conversion of Binary Space Partitioning Trees to Boundary Representation', *chapter Geometric Modelling: Theory and Practice: Chapter II - Representations, pages 286-301. Springer Verlag*, 1996.

4. HEES, M. TH. VAN, 'Knowledge-based Computational Model Assembling', *Proceedings of SCSC2003, Montreal, Canada,* July 2003.

5. HEES, M. TH. VAN, 'Quaestor: Taxonomy based compositional modelling and product configuration', *IMDC2009 10^th International Marine Design Conference, Trondheim, Norway,* May 2009.

6. KONING, D. DE, KOELMAN, H.J, HOPMAN, H., 'A Novel Ship Subdivision Method and its Application in Constraint Management of Ship Layout Design', *COMPIT'11 Proceedings of the 10th International Conference on Computer Applications and Information Technology in the Maritime Industries, May 2-4 2011, Berlin, Germany, pp.292-304,* 2011.

7. OERS, B.J., VAN, 'Designing the process and tools to design affordable warships', *NATO RTO-MP-AVT-173, May 16-18 2011, Sofia, Bulgaria,* 2011.

8. TAKKEN, E.H., 'Concept design by using functional volume blocks with variable resolution', *MSc Thesis, TU Delft*, 2008.

10. AUTHORS' BIOGRAPHIES

Arno Bons holds the current position of Project Manager Maritime Simulation and Software Group at MARIN. He is technically and financially responsible for projects related to streamlining of engineering and design processes at MARIN and other companies in the shipbuilding and offshore industry.

Graduated as Naval Architect at Delft University of Technology, 2006.

Herbert Koelman is founder and general manager of SARC, a consultancy and manufacturer of naval architectural software.

Graduated as Naval Architect at Delft University of Technology, 1985,
Ph.D. (on ship hull design and representation) at Delft in 1999.

Bart van Oers holds the current position of Senior Tool Developer at the Sea Systems Branch of the NL DMO. As a naval architect, he supports the design of future capabilities by developing and improving design tools as well as streamlining the early stage ship design process.

Graduated (Honours) in Ship Hydromechanics at Delft University of Technology, 2005.
Ph.D. Student (on the use of the vast capability of current first-principle tools during conceptual ship design) at Delft University of Technology, 2005-2009.

Erik Takken holds the current position of Senior Naval Architect at the Sea Systems Branch of the NL DMO. He is responsible for balancing ambition with available budget by producing conceptual designs, thus demonstrating technical and financial implications of requirement.

Graduated as Naval Architect at Delft University of Technology, 2008.