A Virtual Single Ship-Design System
Composed of Multiple Independent Components

Herbert Koelman, SARC, The Netherlands, h.j.koelman@sarc.nl
Jan van der Zee, Conoship International, jan.van.der.zee@conoship.com
Theodoor de Jonge, NCG, The Netherlands, jonge@ncg.nl

Abstract

This paper describes a ship design system originating from a collaborative effort in the Netherlands. Arising from a 2008-2011 Dutch development program, a pilot case was implemented where a general CAD program (Eagle, as used by Conoship) collaborated with a specific ship design program (PIAS by SARC). In 2013, this experiment was further enhanced by adding a CAE system (NUPAS-CADMATIC by NCG) into the loop. The paper explains the background, the results and envisioned future. Our experience indicates that coupling dedicated software packages is a better strategy than trying to develop monolithic "one code fits them all" ship design software. Or in short: cooperation beats integration.

1. Introduction

In our industry we may notice that no single CAD/CAE system has emerged which can, or might, serve in all aspects of ship design. So, in practice quite some independent tools are applied — including well-known commercial CAD and CAE systems, but also company-specific dedicated tools or spreadsheets — which happen to collaborate, usually in a rag tag fashion. For decades, effort has been dedicated to smoothen this collaboration by means of a centralized database — or product model. However, recent insights question the potential of such a centralized design. Furthermore, it has been suggested not only to include data in the communication, but also queries.

In close collaboration between the companies Conoship (ship designers, www.conoship.com), Numeriek Centrum Groningen (NCG, software house for ship engineering software (www.nupas-cadmatic.com), and SARC (software house for ship design software, www.sarc.nl), and with support of Netherlands Maritime Technology (http://www.maritimetechnology.nl), a pilot implementation was produced for a system which works on an alternative basis. As subject for this system the internal layout — compartments, decks and bulkheads — of a ship was chosen, for the reasons that:

- This is a rather complex entity, because it comprises spaces (compartments), planes (bulkheads and decks) as well as their mutual relations.
- In the design phase these elements are very frequently modified, in the process of obtaining or optimizing the trim, stability, deadweight, drafts, damage stability etc.
- The data of these elements are used in a 'round-trip' fashion, in other words, these elements are initially modelled in the design phase, then transferred to the engineering phase where some modifications can be applied, commonly not on the main elements, but smaller parts, such as a fuel oil day tank or the bulkhead of a generator room, and can be changed in order to adapt for last-minute requirement changes or equipment changes. Finally, the modified engineering data are transferred back to the design department for producing the final delivery documents, such as final stability booklet and probabilistic damage stability computations.
- Contrary to the hull form, for these entities no data exchange standard, or proposal for a standard, exists.

However, the hull form is not neglected; all planes and spaces are defined up to infinity, and Boolean intersected with the actual hull form. In this fashion hull form changes are immediately reflected in the internal layout.

This paper continues with a discussion of data management and communication, followed by a note on the applied representation of ship's internal geometry. The fourth section contains the core, with the implementation choices, followed by a section with application examples. As usual, in the final section the conclusion is drawn and future work is identified.

2. Data management and communication

In this section the communication aspects are discussed, where the communication is not limited to data, but includes commands (replies and requests) as well. We start with a short state-of-the-art overview,

2.1 You can't always get what you want

For the authors, the situation the Netherlands is the best known, where we have seen quite some endeavours in the field of Product Data Transfer over the past decades, such as CMO/TNO (1983), Hosdes/Mardes (1985), PITS (1988), L/Grand (Logos, 1990) and Open mind (2004). In all these projects it was postulated that a central product model would be the most efficient integration tool. This is demonstrated in figures such as fig. 1, where it is demonstrated that without a neutral model, with N components the required number of interfaces is N x (N-1), and with such a model only N.

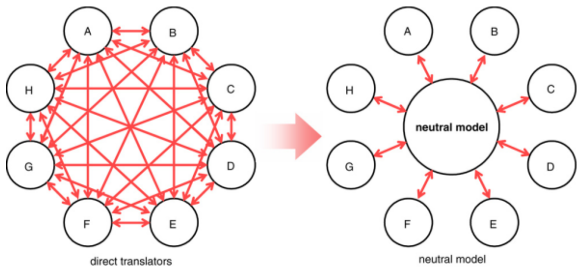


Fig.1: The neutral model saves on interfaces (from *Gieling (2008)*)

The paradigm of the neutral model is also reflected in several neutral file formats which are used for this purpose. According to the survey of *Srinivasan (2008)* the most frequently used data exchange standards appear to be DXF, IGES and STEP, with a combined utilization of 60%. Although quite widely used for data transfer, DXF is merely a drawing exchange and not specifically suitable for the exchange of product model data. IGES could be an option, but facilitates only the exchange of geometrical data, and has limited capabilities for other properties or design rules. That leaves STEP, which has specific naval architectural application protocols with AP215, AP216 and AP218, as the prevailing neutral model for the maritime industry, as also concluded by *Whitfield et al. (2003)*. Although a number of vivid STEP implementations are known to be used in industry[1], the application has a number of drawbacks, which are:

- The implementation requires quite some effort. For a collaborative environment with many constituting applications, some of which are only small `applets', this may turn out prohibitive, because for each of these applets a complete STEP interface must be built.
- The STEP standards themselves provide a number of alternative sub-standards, which hamper their generality. To quote the conclusion of *Gielingh (2008)*: "The neutral model doesn't really exist". In other words, in practice the idealized neutral model with N interfaces is replaced by a much larger number of sub-interfaces. A similar conclusion was drawn in *Whitfield (2011)*, see fig. 2.
- Another warning from *Gielingh (2008)*, where a remarkable conclusion is drawn on the basis of a practical experiment with STEP-based data exchange between CAD packages: "In all three cases significant differences were found: some entities disappeared, others appeared, and again others were changed".
- A final consideration in this respect is that, according to the cited survey, the acceptation of STEP within the industry is with a penetration grade of 15% not very high, so it is certainly not the de-facto standard we should adhere to.

[1] Such as LEAPS/ShipPDX of US NAVSEA.

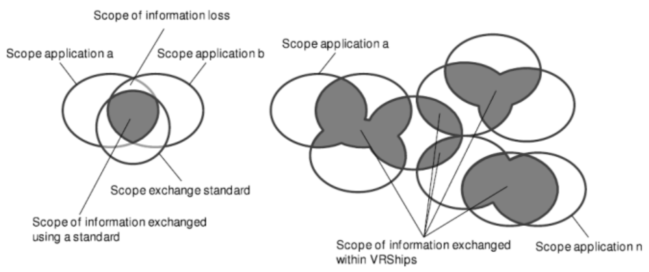

Scope of information loss
Scope application a
Scope application b
Scope exchange standard
Scope of information exchanged using a standard
Scope application a
Scope of information exchanged within VRShips
Scope application n

Fig. 2: Left the idealized neutral model, right the practice with multiple islands of data sharing.

These considerations made the developers decide to take a practical approach, and to apply an XML-based dictionary, which is comparable to the approach of *Whitfield et al. (2011)*, where a similar choice was made, and from which we quote: "Rather than using the STEP modelling language EXPRESS, the tool providers agreed that the eXtensible Markup Language (XML) would be used as the language upon which to base the storage of all data". This dictionary will be filled bottom-up; an entry will only be created when particular data or relations are required by an application. For the terminology of this dictionary, a close look has been taken to the relevant STEP application protocols, but we felt free to add or modify the dictionary.

2.2 Decentralized communication

Apart from choosing the modelling or communication language, choices have to be made on the data storage aspect, for example with a centralized data server or not. In *Li et al. (2005)* a useful distinction is made between *horizontal* and *hierarchical* collaborative CAD. In horizontal collaboration persons from the same discipline are co-designing in a parallel or serial way. In the hierarchical mode teams of different disciplines are involved. Examples of the latter are the collaborations between the design team and specialists in hydrodynamics, construction costs estimations, seakeeping and structural strength. Also the interaction between design and manufacturing falls within this category. The horizontal collaboration can further be subdivided into visualization-based systems and co-design systems. In the first category the emphasis is on a fast distribution of the visual design amongst the peer-designers, however, our goal is from the second category: co-design systems. According to *Fuh et al. (2005)* such systems can be classified into three types:

- Communication server + modelling client (thin server + strong client). In this variant the clients are equipped with full CAD functionality, while the server plays mainly a communication role.
- Modelling server + visualized-based manipulation client (strong server + thin client). In this fashion the main modelling activities are carried out in the server.
- Application or service sharing (peer-to-peer), where the different CAD systems have more or less the same 'weight', and share some of each other's facilities by means of an API.

In the previous sub-chapter it was motivated that a centralized product model appears in practice to be less potent than commonly thought. That makes a 'strong server + thin client' approach not an obvious choice. The 'thin server + strong client' model could be applied, but the drawback is that the clients are rather on their own, while there is a lot of required functionality that could beneficially be shared amongst clients. That brings us to the peer-to-peer model as the preferred one for horizontal collaboration. Concerning hierarchical CAD systems, according to *Li et al. (2005)* the focus should lie on bi-directional communications, instead of 'throwing over the wall', which is in practice frequently the case, still in 2015. As our target CAD system is hierarchical (CAD → CAE → CAD) we have taken this warning at heart, however, the peer-to-peer approach just seems quite suitable for this purpose.

The peer-to-peer model is not only applied for data, but also for requests from one system to another, on which the other gives the answer in a reply. In *Liu (2000)* this mechanism is called API-based communication. Recently, in *Hoffmann et al. (2014)* a similar analysis has been made[2], and similar solutions have been proposed, based on 'queries'. Essentially these are all the same. The advantage of the reply/request functionality is that partner systems can benefit of each other's capabilities, for example in applications as:

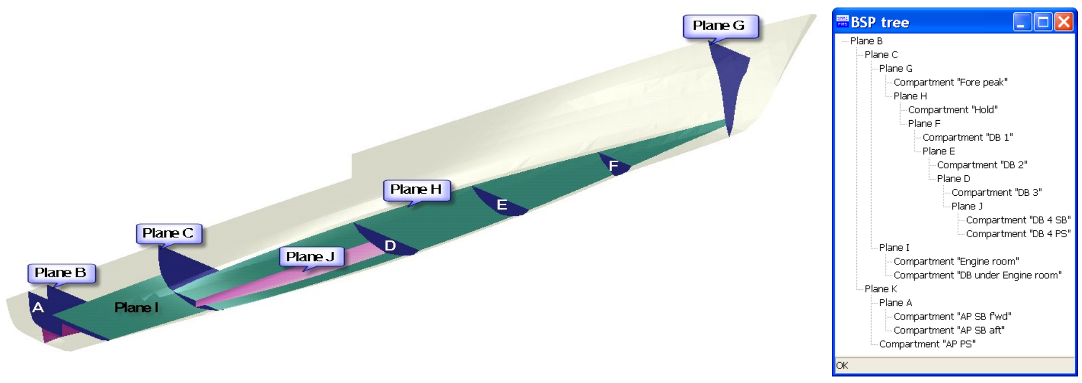
- If application A manages the shape data of hull form and compartments, then application B can request A for the shapes of intersections at different levels. In this case e.g. a general arrangement plan application can quickly be set up in a general CAD system without the need for the CAD system to maintain a full geometric model.
- Commonly, in a tank plan, in the right-upper corner, a list of tanks and their capacities and Centers of Gravity is included. Again, in order to save the tank plan application from the burden of volumetric computations, those parameters can be requested from a connected application which already has this capability, for example the tank sounding module of the hydrostatic package. These requested capacities are not stored in any way, neither local, nor central. They are simply printed in the tank plan, and never used again. The advantage is that we don't need to worry about the validity of stored data, if the tank plan is updated the capacities are simply requested again and recomputed.
- If an application has specified capabilities, say to enlist all compartments and bulkheads that are encountered by a pipe, then the other applications can use this capability without the need to replicate it.

3. Internal geometry modelling

Some years back a representation method for internal geometry was developed which covers the duality of planes (bulkheads and decks) and spaces (compartments). This method is based on the Binary Space Partitioning method (BSP), and has been introduced in *de Koningh et al. (2011)* and *Koelman (2012)*. Its properties are:

- The method is capable to assist in design evaluations and analyses.
- The method is suitable to be applied in combination with various ship hull representations.
- The BSP-concept is easy to understand, and comprehensive, so support in general CAD systems and other software is not expected to fail due to too much complexity.
- The concept is intuitive for the ship designer; it is only cutting spaces in half.
- Sufficient methods are available to convert the representation in other formats, such as B-rep solid models.

In order to illustrate the proposed approach in 3D, an example of a geometric model and the corresponding BSP data structure is depicted in fig. 3.



```
Plane B
  Plane C
    Plane G
      Compartment "Fore peak"
      Plane H
        Compartment "Hold"
        Plane F
          Compartment "DB 1"
          Plane E
            Compartment "DB 2"
            Plane D
              Compartment "DB 3"
              Plane J
                Compartment "DB 4 SB"
                Compartment "DB 4 PS"
    Plane I
      Compartment "Engine room"
      Compartment "DB under Engine room"
  Plane K
    Plane A
      Compartment "AP SB f'wd"
      Compartment "AP SB aft"
    Compartment "AP PS"
```

Fig. 3: Geometric model and corresponding BSP tree.

[2] The conclusion of this paper is "The advantages of our query-based approach include overcoming some of the thorniest obstacles to the data-centric approach to CAD interoperability, namely expressing, in a neutral format, the underlying (often incompatible) assumptions, proprietary algorithms, and heuristics used by the authoring system to interpret imperfect native CAD model data".

4. The implementation

Based on the tools and considerations of the previous sections, a system design was conceptualized with the following properties:

- Each application remains independent, although applications can use each other's facilities. So, applications cooperate, but are not integrated.
- Not only data are shared between applications, but also algorithms and processes. This is the reply/request or API style of communication as discussed before.
- Applications communicate peer-to-peer, without a central database or a central traffic manager.
- Communication directly over TCP/IP, coded in XML.
- If in one application the ship design is extended or modified, this action is directly codified in XML, and pushed to the other applications of the system. It is their task to interpret this message, to modify their local data structures accordingly, and to update the on-screen representation immediately. In other words, each design modification in one application is immediately processed in the other applications. In this sense the contributing applications collaborate as one single virtual ship design system.
- The XML dictionary is not written on forehand. It grows 'on demand', just like a natural language dictionary.
- Keep It Simple Stupid: the initial design addresses the core issues of ship design, without aspects of version control, rights management etc.
- WYSIWYG: keep in each development stage a real ship design in the picture. In order to show potential users or co-developers the real target instead of flow charts and fancy plans.

The first implementation contains functionality for the initial ship design process. This is listed below, where for each function the contribution application is also mentioned, however it should be clear by now that the system functions as a whole.

- Hullform design, for example with the Fairway hull design module of PIAS (see *Koelman et al. (2012)* for the most recent state-of-the-art overview on hull form topics) or any other hull design software that can export to IGES.
- Compartment definition (including sounding pipes, pressure sensors etc.) in PIAS. Or, alternatively bulkhead and deck definition in PIAS. Or, alternatively, bulkhead and deck definition in NUPAS-CADMATIC.
- Design visualization in all three contributing systems.
- Generation of tank plan and general arrangement plan, in Conoship's CAD system Eagle. Or, alternatively, in NUPAS-CADMATIC.
- Tools for equipment fitting, in Eagle.
- Computation of tank capacity tables. As well as other hydrostatic-related tasks, such as intact stability, (probabilistic) damage stability, grain stability etc., in PIAS.
- Scantling sizing, in NUPAS-CADMATIC.

5. Application example

The example is a recent design of an oil/chemical tanker at Conoship. As proof of concept, parallel to the conventional way of working, this project was also done with the aid of the tools and data transfer technology as described here. The project comprised hull form design, tank arrangement, general arrangement and the initial design of the midship section structural arrangement.

5.1 Hull form design

The hull was designed with PIAS' dedicated hull design module Fairway, fig. 4 shows the result. It was developed to such extent that basic requirements such as displacement, block coefficient and longitudinal center of buoyancy were met.

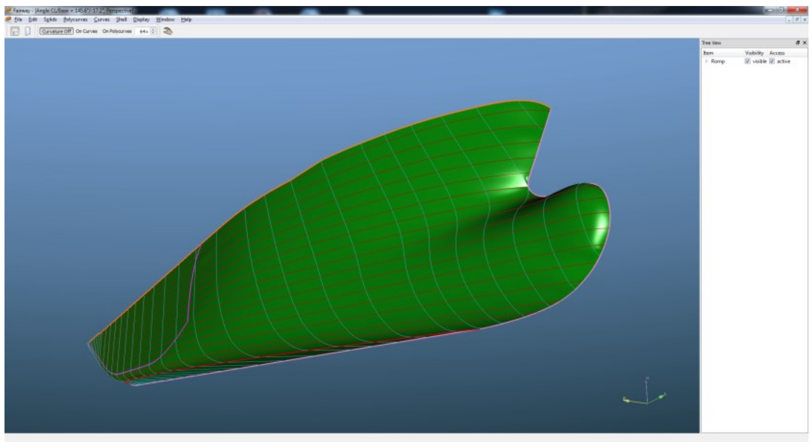


Fig. 4: Fairway hull form shape model.

5.2 Tank arrangement and General Arrangement

Once a satisfying hull was set-up, a start was made with the design of the tank arrangement. To be able to check for feasible solutions regarding the combination of double bottom height, ballast capacity, cargo tank size and capacity, the design was simultaneously developed in PIAS and Eagle. First, a basic 3D model of bulkheads, decks and compartments, which covers the most important compartments, was modelled in PIAS. This model was used to check the required ballast and cargo capacity. Simultaneously, the basics of the tank arrangement and General Arrangement were designed in Eagle, utilizing the available data from PIAS. First, the required hull lines were requested from PIAS, and subsequently, intersections of the watertight arrangement. With these data, the cargo tank arrangement was optimized in Eagle regarding cargo tank diameter, required inspection space, distance between tank and shell and other requirements. Figures 5 and 6 demonstrate this process.

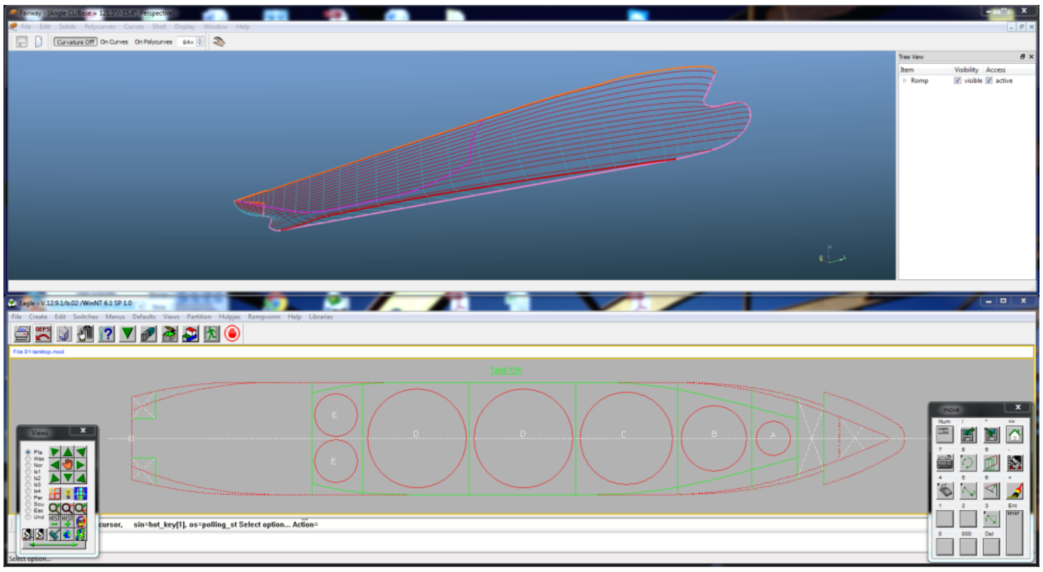
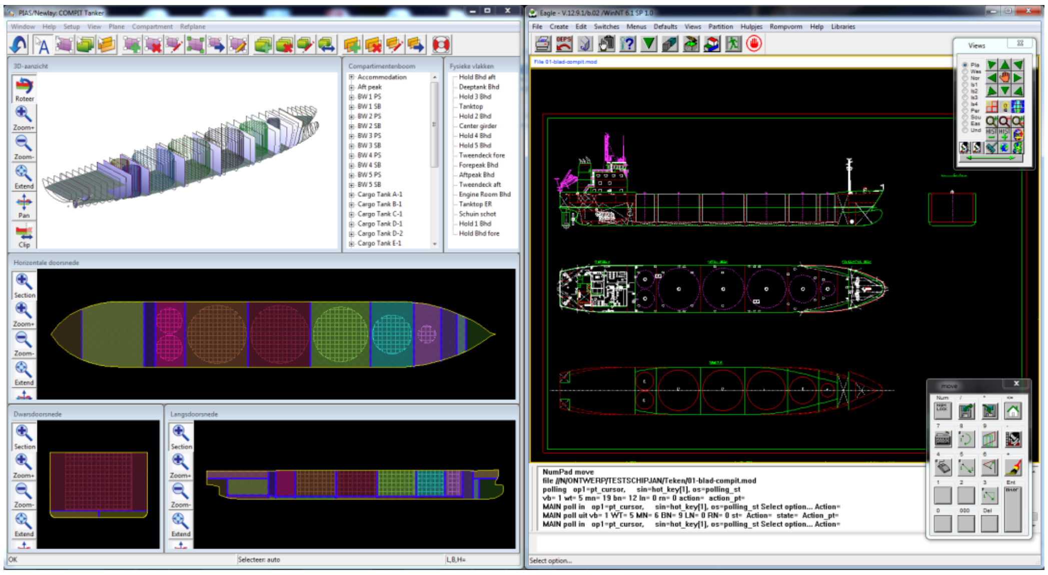


Fig. 5: Design of cargo tank arrangement in Eagle on basis of the waterlines as requested from Fairway.

5.3 Intact and damage stability

Once the arrangement of ballast tanks, cargo tanks and holds was sufficiently developed, the PIAS model was prepared for intact and damage stability calculations. Critical loading conditions and damage cases were determined and modelled. Subsequently, an optimization cycle was started in which the tank arrangement in Eagle and stability model in PIAS were further optimized regarding intact and damage stability requirements. This encompassed changes in the hull form, cargo tank and ballast tank arrangement, loading conditions and damage cases. Thanks to the instant data exchange, various arrangement alternatives could quickly be checked for compliance with the requirements.



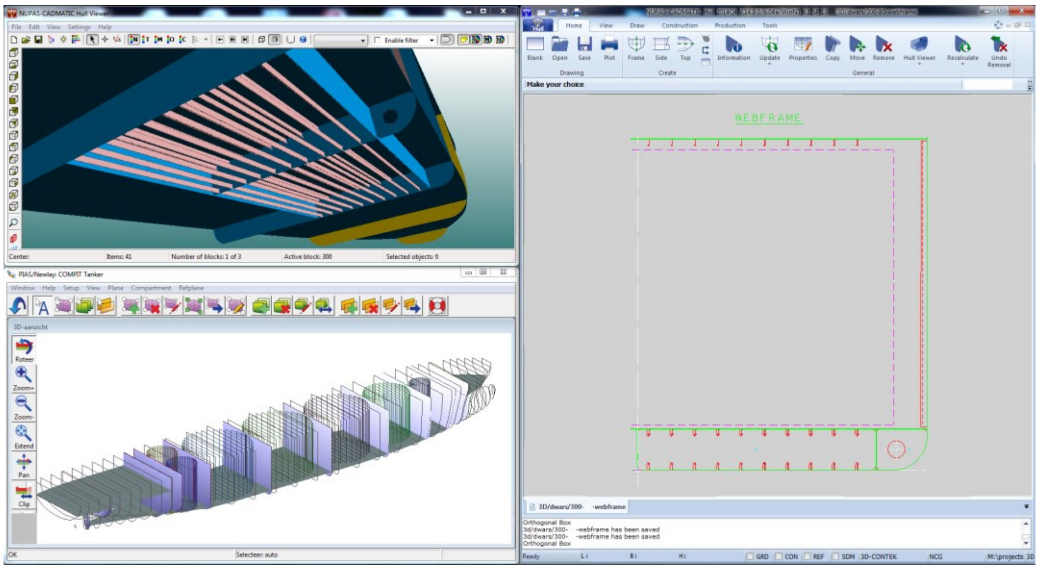
```
NumPad move
file //N/ONTWERP/TESTSCHIPJAN/Teken/01-blad-compit.mod
polling   op1=pt_cursor,    sin=hot_key[1], os=polling_st
vb= 1 wt= 5 mn= 19 bn= 12 ln= 0 rn= 0 action=  action_pt=
MAIN poll in   op1=pt_cursor,    sin=hot_key[1], os=polling_st Select option... Action=
MAIN poll uit vb= 1 WT= 5 MN= 6 BN= 9 LN= 0 RN= 0 st=  Action=  state=  Action_pt=
MAIN poll in   op1=pt_cursor,    sin=hot_key[1], os=polling_st Select option... Action=
```

Select option...

Fig. 6: Optimizing both the stability model and general/tank arrangement simultaneously, utilizing data exchange between PIAS and Eagle.

5.4 Midship section

To determine the initial scantlings of the midship and to elaborate a midship section drawing, the midship was set up in 3D in NUPAS-CADMATIC-Hull. Using the watertight bulkheads and decks delivered by PIAS as a basis, the construction parts were modelled and the sheet drawings were created (fig. 7). This was done with the aid of a classification society's scantling tool, for which the input had to be given manually because this tool is not connected to the data transfer facilities as described in this paper.



```
3D/dwars/300-   -webframe
Orthogonal Box
3d/dwars/300-   -webframe has been saved
3d/dwars/300-   -webframe has been saved
Orthogonal Box
```

Fig. 7: Watertight arrangement of bulkheads and decks is transferred from PIAS to NUPAS-CADMATIC-Hull as basis to elaborate the midship section in the latter.

6. Findings, future work and conclusion

At the moment of writing, in March 2015, a pilot implementation of this virtual single ship-design system is ready for practical, although still experimental, use. It has been used to produce the application examples of the previous section. These experiments have led to the following findings:

- The system works as anticipated. The smooth data sharing reduces ship design times, ensures model consistency and, as a result, reduced the failure probability.
- The ratio between implementation effort and gained results is remarkably low. An important reason is the reply/request mechanism, which enables applications to use other's facilities without having to replicate them.
- No performance degradation due to network traffic was experienced.
- The system has proved to maintain consistency over all applications.
- The system relies on instant messages from applications to each other. It will be obvious that all applications should be switched on, and attached to the current design project, in order for these messages to be processed correctly.

However, as the system has grown from bottom up, some important issues have not yet been addressed, and the time has more or less come to identify those subjects:

- The system has been designed without a central database or data flow manager. With three contributing applications, and benevolent users it works well, however, for more extended use some form of central background management could be necessary.
- Similarly, in more extended applications design history and rights management could be addressed.
- The XML dictionary is managed without a formal system (in Google docs now). Formalization could improve the coherence of the dictionary.

Apart from these potential extensions, for the near future the partners have identified the following tangible steps and developments:

- Stimulate other partners, with possibly other software, to share on this development.
- Include pipe geometry in the system. The reason is that a) pipes are already schematically defined in the design phase (e.g. de-airation), b) they play an important role in design and delivery documents (e.g. the effect of damage to pipes on the probabilistic damage stability), and c) their final routing is determined in the engineering phase. This certainly justifies a smooth sharing of piping data.
- Investigate the incorporation of a probabilistic steel weight estimation method into the system.
- Develop a more comprehensive pilot case, in the context of the EU-funded SMARTYards project.

References

DE KONINGH, D; KOELMAN, H; HOPMAN, J.J. (2011), *A Novel Ship Subdivision Method and its Application in Constraint Management of Ship Layout Design*, 10th Int. Conf. Computer and IT Appl. Maritime Ind., Berlin, pp. 292-304.

FUH, J. Y. H.; LI, W. D. (2005). Advances in collaborative cad: the-state-of-the art, Computer-Aided Design 37(5), pp. 571-581.

GIELINGH, W. (2008). *An assessment of the current state of product data technologies*. Computer-Aided Design, 40(7), pp. 750-759.

HOFFMANN, C.; SHAPIRO, V; SRINIVASAN, V. (2014). *Geometric interoperability via queries*. Computer-Aided Design 46(1), pp. 148-159.

KOELMAN, H.J. (2012). *An approach to modelling internal shapes of ships to support collaborative* *development.* Proc. TMCE 2012, May 7-11, 2012, Karlsruhe, Germany, pp. 685-696.

KOELMAN, H. J.; VEELO, B. N. (2013), *A technical note on the geometric representation of a ship hull form*. Computer-Aided Design 45(11), pp. 1378-1381.

LI, W. D.; LU, W. F.; FUH, J. Y. H.; WONG, Y. S. (2005). *Collaborative computer-aided design - research and development status*, Computer-Aided Design 37(9), pp. 931-940.

LIU, X. (2000). *Cfaca: component framework for feature-based design and process planning.* Computer-Aided Design 32(7), 397-408.

SRINIVASAN, V. (2008). *Standardizing the specification, verification, and exchange of product geometry: Research, status and trends*, Computer-Aided Design 40(7), pp. 738-749.

WHITFIELD, R. I.; DUFFY, A. H. B.; MEEHAN, J. (2003). *Ship Product Modeling*, Journal of Ship Production, 19 (4), pp. 230-245.

WHITFIELD, R.; DUFFY, A.; YORK, P.; VASSALOS, D.; KAKLIS, P. (2011). *Managing the exchange of engineering product data to support through life ship design*, Computer-Aided Design 43(5), pp. 516-532.