Proceedings of TMCE 2012, May 7–11, 2012, Karlsruhe, Germany, Edited by I. Horváth, Z. Rusák, A. Albers and M. Behrendt
© Organizing Committee of TMCE 2012, ISBN 978-90-5155-082-5

AN APPROACH TO MODELLING INTERNAL SHAPES OF SHIPS TO SUPPORT COLLABORATIVE DEVELOPMENT

Herbert J. Koelman
SARC BV
The Netherlands
H.J.Koelman@sarc.nl

ABSTRACT

In this paper the development and properties of a ship design system are addressed. This system is aimed at the preliminary design stage, and will consist of a confederation of heterogenous tools for subtasks. The idea is to let those tools to be as tiny as possible, in other words, to avoid monolithic solutions as much as possible. After a short introduction to the novel system, the necessity of a new method, and tool, for the internal shape of a ship is expressed. After a sub-sequent inventory of available tools and methods in this field, a Binary Space Partitioning (BSP) method was identified to be most promising. The implementation of this method is presented and evaluated, and auxiliary functionality is discussed, such as support for constraint management during design. A second line in this paper addresses the methods for inter- tool communication, and exchange of data and de- sign rules. Options for these issues are presented,after which it is concluded that a solution based on a) ad-hoc, XML-based data and rule exchange, b) a bus-like system design and c) the employment of a commercial data management system would be most practicable. Finally, a framework for system validation and evaluation is presented, conclusions are drawn, and suggestions for future developments aregiven.

KEYWORDS

BSP, internal shape modelling, ship design, product geometry exchange

1. INTRODUCTION

In the years 2008–2011, the Innovero project was commenced in The Netherlands. This project, where shipyards, ship designers, ship users, software developers and a university worked closely together, was focussed on the development of a software suite intended for the preliminary ship design stage. Key elements of this project are a) breaking down monolithic software suits into smaller entities, b) development of a data exchange language, c) project management by means of a knowledge management system, d) a flexible architecture, aimed at integration on the one hand, but allowing for a dispersedness of tools on the other hand, and, finally, e) not being tied to a particular software platform or tool.

The *Innovero* toolkit is designed to contain quite some specific design analysis functions, such as vulnerability analysis, data and power cable routing, hydrostatics and stability, resistance and propulsion, seakeeping behaviour, drafting on general purpose CAD-systems etc. etc. The ultimate goal is to provide cost and weight estimations, but the level of detail necessary for that task has not yet been reached. Some results of this project are presented in (Bons et al., 2011). In the course of the development of Innovero it appeared that existing methods, and consequently also tools, for the design and representation of the geometry and topology of the internal shape of the ship could not fulfill the requirements of the different analysis tools.

The background of this issue, as well as the design and realisation of a new method an tool is the subject of sections 2, 3 and 4. Furthermore, the issues of communication and management of data and rules play a paramount role in a environment such as Innovero, and these subjects are addressed in section 5. With the constituent as discussed until sofar a real-life confederation of tools, consisting of the internal shape tool, a general CAD system and the data and rule management system could be built, this is discussed shortly in section 6, while in section 7 some more general validation issues are addressed. Finally, conclusions are drawn and topics for further development are identified.

2. INTERNAL SHIP SHAPE MODELLING

2.1. Design requirements for the internal shape representation

The design evaluations where the Innovero project is focussed on, are dependent on the external and inter nal shape of the ship, and for those evaluations to be performed with sufficient accuracy, the shape geometry as well as the shape topology should be available at a detailed level. For the external shape quite some representation and modelling methods exist, and those methods are considered to be satisfactory for the Innovero purposes. In sub-section 4.1 some applicable methods will be introduced and demonstrated.

However, for the internal shape of the ship hull, its subdivision, the quest was for an effective and comprehensive method. As first step, an inventory of available methods was made, where, as a guideline for the selection of such a method, a number of implementation-related requirements have been formulated:

1. 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 by means of cross sections.
2. The applied method, and its underlying entities, should be sufficiently easy to understand, and comprehensive. The reason for this requirement is that general-purpose CAD software forms a major component of the Innovero system design, and it would hamper the acceptance of the Innovero system if for that task persons with a specialist level of knowledge of shape modelling issues would be indispensable.
3. Intuitive for the end-user, for if the method is intuitive, its implementation will probably also be. And this will make the implementation appealing to the designer.
4. Compatible with (or convertible into a format compatible with) the applied naval architectural analysis software, which is PIAS, by company SARC, of the Netherlands. 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.

However, the most important requirement was not related to software or implementation as such, but was included in order to assist the ship designer as much as possible. The issue is that for some design or definition tasks it is pleasant, for a human being, if the focus is on spaces, while for other tasks a plane-based focus is more desirable. And with ‘focus’ the basic modelling entity is addressed, so, summarised, one time it is easier to model a compartment directly by its boundaries, and in other cases modelling by means of internal subdividing planes (bulkheads and decks) is more appropriate. Obviously, planes and spaces are interrelated, so it is necessary to address their duality.

2.2. Methods for the representation of internal geometry

Representation methods, as applied in today’s popular naval architectural software tools, have been investigated in (Lee et al., 2003) and (Lee et al., 2009), where it is concluded that they fall apart in two categories:

- Those where spaces, or compartments1[1], are defined by their bounding planes. This method is utilized in the commercially available ship design packages NAPA and *Tribon design*[2].
- Those where a wireframe model of compartment boundaries is applied, a method which is reported to be applied in the *SIKOB* package.

Both methods are considered to be inadequate by (Lee et al., 2009), for the reason that they do not provide an integral modelling method. As a solution, a data structure was proposed which is based on a non-manifold solid model3[3], which gives the solid also an internal subdivision.

Another relevant paper is (Alonso et al., 2008), where many implementation details are unfortunately not disclosed, but from which it is apparent that the described software system allows for multiple representation methods. The conceptually simplest method is a compartment bounded by six planes, but there is also another method based on, quote, a “successive split of an initial space”, with a further detailing: “from the first level compartment definition the system allows the iterative subdivision

[1] In this paper the words *space* and *compartment* are considered to be synonymous.

[2] As well as *PIAS*, does the author wish to add.

[3] Contrary to a manifold solid model, a non-manifold has also *internal* faces, edges and vertices.

of the created compartments by user-defined planes. This process can be repeated as many times as needed to obtain the desired detail of subdivision". Certainly an interesting idea.

3. A NOVEL MODELLING METHOD

3.1. Evaluation of available methods

The available methods, as introduced in the previous sub-section can be compared with the requirements of sub-section 2.1:

- By spaces which are defined by their bounding planes. Such a method is not suitable for our purposes, because it lacks an explicit definition of, or reference to, realistic planes, so the duality between spaces and planes is not addressed.
- By a wireframe model of compartment boundaries. Although in this case some relationship between spaces and planes does exist, their reverse relationship does not, so the duality is not fully addressed.
- By a non-manifold solid model, according to the proposed data structure of fig. 1. Although this solution integrates the modelling of planes and spaces, a drawback can be found in the complexity of the implementation. The author of this paper has some experience in the implementation of a software system for the design of a ship hullform, based on a manifold solid model, see e.g. (Koelman et al., 2001), (Koelman, 2003) (and (Veelo & Koelman, 2011) for its latest incarnation), and it appeared to be a tedious task to implement all low-level operators precisely and flawlessly. And, with a non-manifold solid, an order of magnitude more complex than a manifold one4[4], for the non-manifold case the task would even be much greater.

  However, this is only a practical consideration, a more fundamental issue is that the simplicity requirement, the second from the list of subsection 2.1, is not met.
- The method of (Alonso et al., 2008) is not fully elaborated, so its applicability cannot be evaluated. However, we will encounter some notions of this paper in the next sub-section.

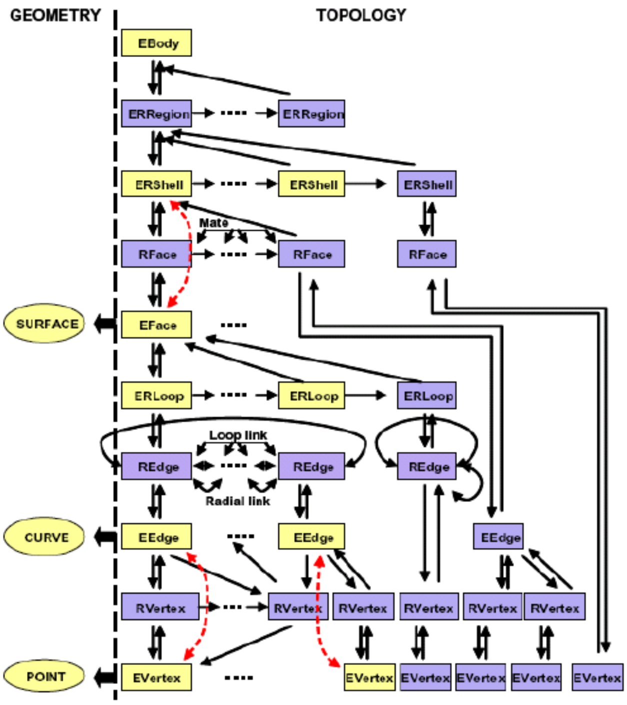


Figure 1 Non-manifold data structure

3.2. A proposal for a novel BSP-based method

Starting from the most important design requirement of sub-section 2.1, the capability to tackle the duality between spaces and planes, we may conclude that the spaces-part can be fulfilled by conventional methods. However, for the modelling of planes, realistic planes, the question arises what is the handiest method to represent and define those planes. And with handiest we mean which method intrinsically fits to the 'logic', the intuition and the expectation of the average ship designer. In discussions with ship designers and a designer's panel5[5] 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 may see the ship's compartments. Conventionally, the recursive subdivision is represented by a binary tree, which is also a suitable

[4] Which is illustrated by the fact that the non-manifold model of fig. 1 contains ten constituting topological entities, which can be compared to a number of five for a manifold solid.

[5] Withabut 12 participants, all experienced ship deginers, held in 2006

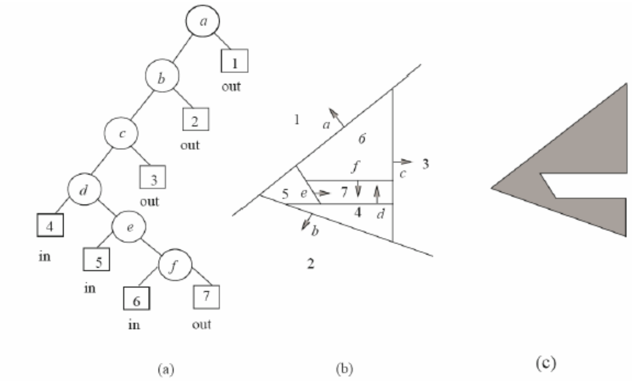


Figure 2 BSP tree (from Comba & Naylor, 1996)

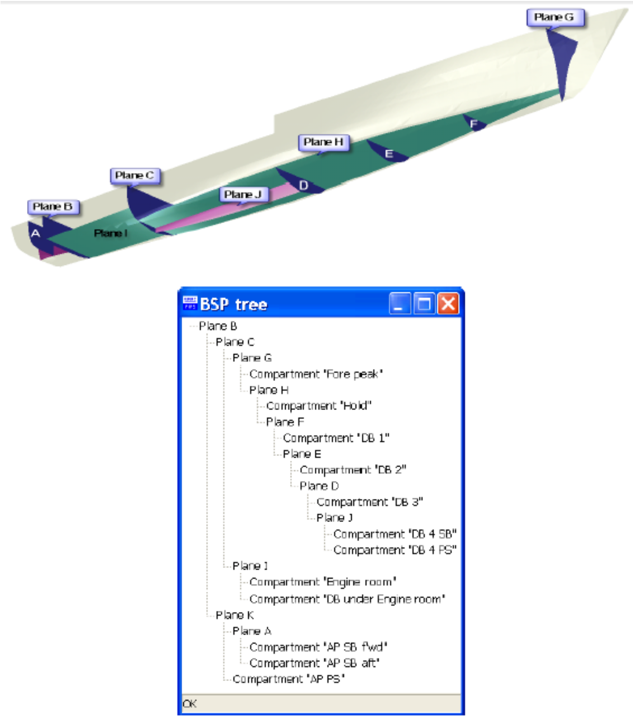
internal representation in a computer program. An example of a BSP- application in the plane is given in fig.2, where the shaded 2D figure on the right is recursively split by the planes a to f (which form the nodes of the tree, the partitioning planes) and the cells 1 to 7 (which form the leaves of the tree, the spaces). By the way, the phrases in (Alonso et al., 2008), as cited in sub-section 2.2, suggest that their method is somewhat similar to the BSP approach.

Our thesis is that this BSP concept could provide an adequate modelling method for our purposes, because in the BSP-tree both planes and spaces are included, so they both are included cooperatively. Its potential compared against the list of requirements of sub-section 2.1 delivers:

1. The method has no (apparent) limitation if applied in combination with various ship hull representations.
2. 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.
3. The concept is certainly intuitive for the ship designer; it is only cutting spaces in half.
4. Sufficient methods are available to convert the representation in other formats, either plane-based or space-oriented, see the list below.

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

To evaluate the BSP-based approach further, an experimental implementation has been produced, which is the subject of the next section. Fortunately, in the literature quite some methods have been presented, which assisted in the implementation and helped to gain insight:



```
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 fwd'
      Compartment 'AP SB aft'
    Compartment 'AP PS'
OK
```

Figure 3 Geometric model and corresponding BSP.

- A general description of the BSP, its properties and a number of basic algorithms have been presented in e.g. (Thibault, 1987), (de Berg et al., 1998), (Schneider & Eberly, 2003), (Naylor, 1998) en (van den Bergen, 1999). An application of the BSP in design and evaluation of buildings is presented in (Jeong & Ban, 2011).
- Conversion from B-rep to BSP, presented in (Thibault & Naylor, 1987), (Naylor, 1992), (Jiang, 1996), (Schneider & Eberly, 2003) and (Ghali, 2008). An integration between B-rep and BSP was proposed in (Vanecek, 1991).
- Conversion from BSP to B-rep, discussed in (Buchele & Roles, 2001) for objects with nonplanar faces, and in (Thibault & Naylor, 1987) and (Comba & Naylor, 1996) for polyhedrals. Ref. (Comba & Naylor, 1996) was commented by (Ghali & Smith, 2005) with the words “....that algorithm first constructs the boundary of the convex polyhedra at the leaves and then merges the resulting polyhedra. The algorithm also appears to be too involved to be practical.”, however, no alternative is provided, unfortunately.
- Boolean operations on BSP’s, in e.g. (Thibault & Naylor, 1987), (Naylor et al., 1990) and (Lysenko et al., 2008).

4. THE IMPLEMENTATION AND ITS FUNCTIONALITY

4.1. The implementation of the new method

For a BSP-based implementation it must be considered that by partitioning only convex spaces can be created, while in reality the ship’s compartments maybe concave. This asks for additional support. Furthermore, in the ship design practice it is considered to be advantageous if locations or coordinates can be defined relative to reference planes, because it reduces the amount of re-definition work in the case of design changes. For these reasons, 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, and may be concave. A compartment can be composed of one or more subcompartments.
- The subcompartment, which equals the BSP leaf, so it is always convex. Subcompartments are logical constituents of a compartment, they have no physical significance.
- The physical plane, which is a real plane within the ship, so a bulkhead or a deck. The physical plane may be bounded, which means it does not have to 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 actions.

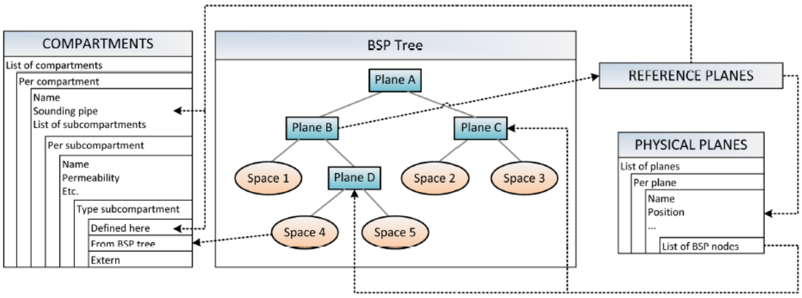


Figure 4 Proposed BSP-based data structure

The BSP forms the glue between these entities, and is not available to the program user as a separate entity, so the user, the ship designer, only works with familiar entities such as planes and compartments. This data structure is depicted in some more detail in fig. 4.

This method is implemented in a computer program, in such a way that first the BSP divides the infinite space into subcompartments, and subsequently these subcompartments are Boolean unioned with the hull form. Because in practice different representation methods for the hull shape are applied, preferable support for a multitude of methods should be provided. These particular hull shape representation methods are, in descending order of completeness (of the modelling method):

1. A (manifold) solid model. The BSP-leaf is completely bounded by planes from up-level nodes of the BSP, so it can be converted to a solid. So two complete solid models are available, upon which Boolean operations can be performed quite easily, e.g. as elaborated in (M¨antyl¨a, 1988), a method which, by the way, was commented in (Ghali & Smith, 2005) as follows: “Although such algorithms are at the outset simple, an implementation is usually quite intricate since neighborhoods must be maintained and that gives rise to many special cases”. NeverthelessM¨antyl¨a’s method was applied and indeed exceptions and special situations have been encountered, see (Koelman, 2006) for a report and discussion thereof. However, after

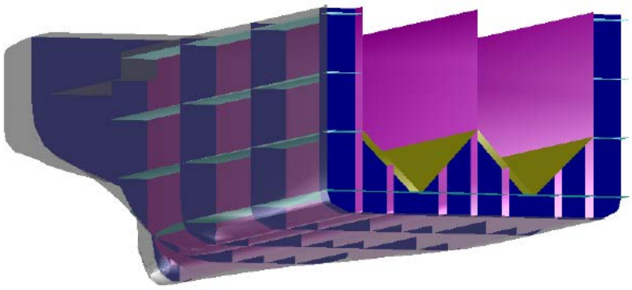


Figure 5 GUI with solid hull model

careful implementation this methods works satisfactorily, although it would be interesting to investigate whether an intrinsically more robust method, e.g. that of (Smith & Dodgson, 2007), would provide more stable or faster Boolean ops. Fig. 5 shows one window of the GUI with a ship hull as solid model, while fig. 6 shows a screenshot with the shape of the solid compartment, as unioned with a solid hull model[6].

2. A surface representation. Although B-spline or NURBS-based surface representations are common for hull shape representations, (currently) they are not supported in our implementation. One reason is that the necessity is low, because such representations can easily be converted into a wireframe, by making intersections. Another reason is that if there is a complex configuration of surfaces, the ‘inside’ and ‘outside’ parts are not unambiguously defined. Although particular design systems may have their own methods to indicate those sides, such methods are not generically applied.

3. A wireframe sectional representation, where only discrete sections of the hull form are defined. This situation is solved by making a section of the BSP leaf, and perform a 2D Boolean union of this section and the corresponding hull section. Fig. 7 shows the GUI with a wireframe hull as basis.

4.2. Functions of the computer program

In order to support an extensive evaluation by the (potential) users, the implementation was equipped with quite some functions, from which the core functions are:

- The program has two faces; it can either work as a stand-alone design tool with a Graphical User Interface (GUI), or it can act as a server, to serve other processes. In the latter fashion it can e.g. provide properties of the internal ship entities, such as the shape of decks or bulkheads, or the tank volumes and moments of inertia.

- Functions of the GUI: add/remove planes, change plane geometry, move plane, manage properties of planes and compartments. • Calculations, such as plane area, compartment volume, compartment surface area (= paint area), compartment free surface moments (which affect stability) etc.

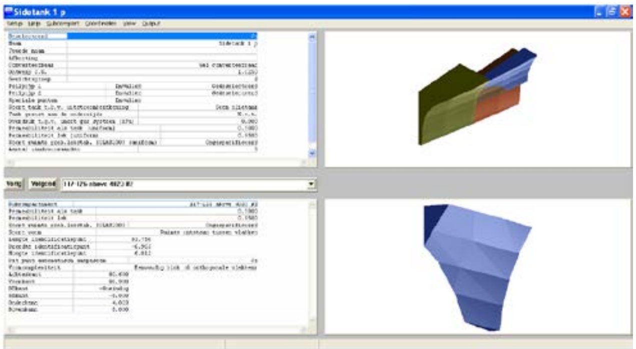


Figure 6 Compartment shape on basis of solid hull model

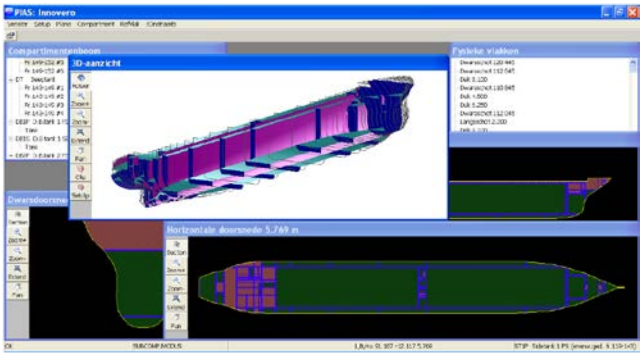

PIAS: Innovero
Venster Setup Plane Compartment Refdat Constraints
Compartimentenboom
3D-aanzicht
Fysieke vlakken
Dwarsdoorsnede
Horizontale doorsnede 5.769 m

Figure 7 GUI with wireframe hull model

- Import from the legacy PIAS format, which is essentially a representation by coordinates of vertices of frustums.

- Import of a set of planes, either extending over the entire ship hull, or bounded by other planes.

- Conversion from planes to spaces and vv.

- Export to PIAS, DXF and VRML. The latter can be synchronized via a FTP-server with a VRML viewer app[7] in the iPhone, see fig. 8.

- Multiple ways to communicate with other software, which is either the knowledge management system as applied in Innovero, or other end-user software, such as a general-purpose CAD system.

  Currently, implemented communication methods are file, named pipes and TCP/IP.

- Generation of a 2D subdivision plan, where the user can specify locations of cross sections, and their placement on a (virtual) sheet of paper. This subdivision plan, see fig. 9 for an example, can

[6] At the state of development of the software at the time of writing it is unfortunately still in Ducth

[7] In particular the *On-hand Viewer* by the company *MWF-Technology*

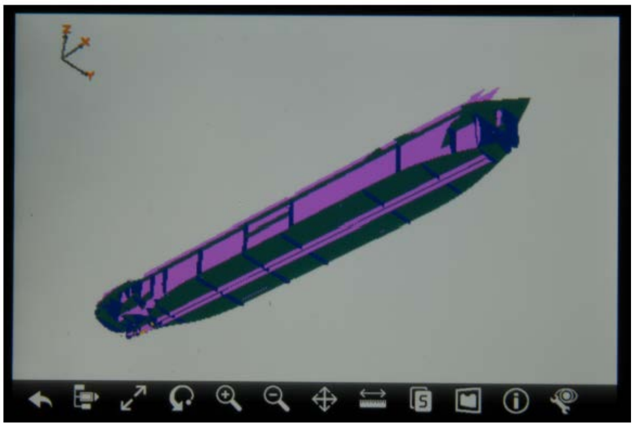


Figure 8 Model of internal plans, in VRML format, synchronized with an shown on the iPhone.

serve as a founding layer for a complete general arrangement plan, which may include details such as deck houses, life boats and winches.

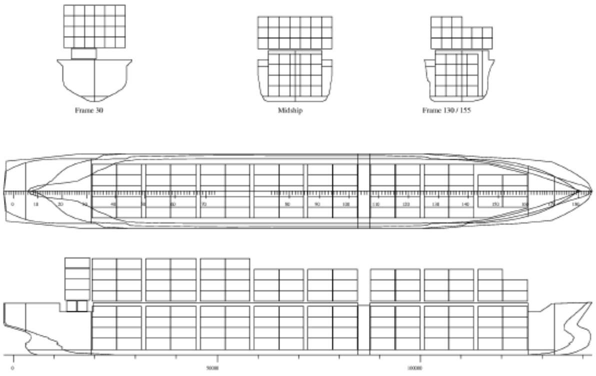


Figure 9 Subdivision plan.

Furthermore, the program has some supporting functionality, in the form of constraint management. This subsystem, which is described in more detail in (de Koningh et al., 2011), comprises the following parts:

- Identify design constraints. Currently four typesof constraints are recognized, a) amount (e.g. required number of transverse bulkheads), b) position of planes (decks, bulkheads), c) volume of tanks/ compartments or combinations of tanks/ compartments and d) areas of decks or combinations of decks.
- Each constraint can state a minimum required value, a maximum allowed value or a required interval.
- Each constraint is linked to a plane or to a compartment, or to a combination of planes or of compartments.
- The software checks whether all constraints are met. If not, planes are shifted by a minimal distance, just to let all constraints apply.
- If the total system of constraints is conflicting, a solution is proposed based on equal weighting of all constraints. If the user prefers, a window with slide bars appears, the so-called constraint equalizer, where the user can modify the (relative) weight of each constraint. This will result in a different design solution, which is immediately drawn at screen. Obviously, the conflicting system of constraints will not be solved by this solution, but at least the user can interactively choose the best compromise.

5. COMMUNICATION AND DATA MANAGEMENT

The presented internal ship shape modelling method will not be used on a stand-alone basis, but will be incorporated in an environment of collaborative tools. In order for these tools to collaborate properly, a communications framework has to be chosen or developed and the transfer of data and/or design rules between the tools should be managed in one way or another. These two subjects will be discussed in the next two sub-sections.

5.1. Management of data and design rules

An aim of the Innovero project is that the collaborating software tools not only share their design data, but also design rules as well as relations between data. For this purpose, the Quaestor program of MARIN, the Netherlands, was selected, see (van Hees, 2009) for a description of the methodology on which this program is built. In this central role Quaestor plays the role of a design server, which can serve all satellite applications with data and rules, as depicted in fig. 10. Tasks performed by the Design Server are amongst others:

- Supply applications with relevant data, and store possibly modified data as returned by applications.
- Track which data is used and returned by applications, and notify the project manager on invalid data (which can e.g. be the case if an analysis is dependant on other data, which was modified, so that the analysis should be re-done).
- Provide the project manager with an overview of the main data of the design at hand.

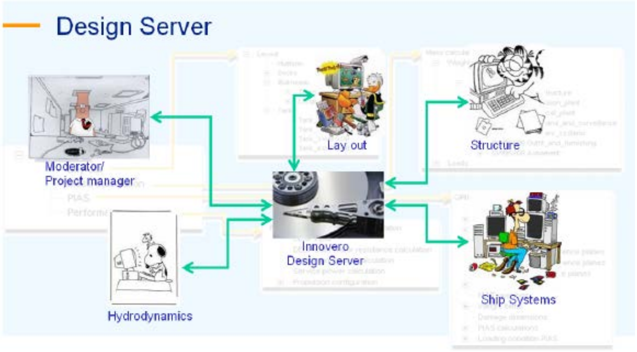


Figure 10 Central Role for the design server

- Evaluate relatively simple rules, for which no external application is required. Examples can be a single-line equation, or reading a lookup table.

5.2. Communication

Obviously, for the communications between applications and the Design Server, agreement on some standard was required. It was an option to apply a well-known standard, and because a recent survey ((Srinivasan, 2008)) indicated that the most frequently useddata exchange standards appear to be DXF, IGES and STEP, with a combined utilization of 60%, we will limit our considerations to those three.

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. However, the application of STEP has a number of drawbacks, which are:

- The implementation requires quite some effort. For a collaborative environment with many constituting applications, some 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 ref. (Gielingh, 2008): "The neutral model doesn't really exist".
- 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-factor standard we should adhere to.

These considerations made the consortium decide to take a practical approach, and to apply an XML-based dictionary8[8], which will only be filled 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 the consortium is free to add or modify the dictionary. It is of paramount importance that the materialisation of the dictionary (that is its physical existence, on paper or on computer file) is managed properly and kept up-to-date. For this purpose a dictionary format was proposed, which in itself is encoded in XML, with the idea that in this form it will be possible to convert it into any other format which may seem convenient at some future moment. The specification of this format, and its features, is currently under development.

In fig. 10 the initial system configuration is depicted, with the Design Server in the center. Such a system design implies that all communication is tunnelled through the Design Server, but in some occasions that would be sub-optimal. For example, if application A, say a general CAD system such as Autocad, is presenting a general arrangement plan of a ship for which it needs the shape of some intersections of decks or bulkheads with the ship hull, which it requests from a ship subdivision application (application B). This kind of data transfer between A and B is only for the moment; the requested data are completely derived from other data (specifically the position of the decks and bulkheads, and the hull shape) and there is no necessity to store or trace it.

So, A can easily ask this data from B on a bilateral basis, there is no need to involve the Design Server

[8] Comparable with the approach of ref. (Whitfield et al., 2011), where the same choice was made, quote: "Rather than using the STEP modelling language EXPRESS, the tool roviders agreed that the eXtensible Markup Language (XML) would be used as the language upon which to base the storage of all data".

in this case, in fact, using the Design Server would a) require specific functionality in the Design Server and b) slow down the data transfer process, which may lead to performance degradation in the case of highly data-intensive and interactive applications. For this reason, in a later system design a bus structure is conceptualised, see fig. 11, where the upper row contains domain-specific applications, and the lower row contains only the Design Server. However, there is no hierarchical relation between the rows or within a row. In this fashion, the Design Server is more or less eavesdropping the communication between the other applications, processing the information if relevant, and ignoring if not.

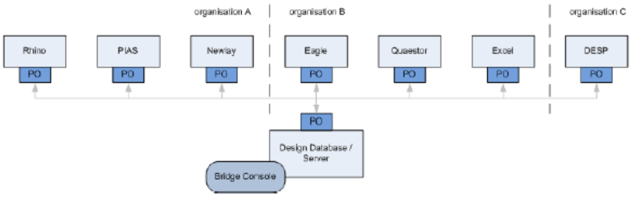


Figure 11 Applications organized in a bus structure

With such a structure applications can either communicate on a bilateral basis, or with the Design Server, all tailored for the specific particulars of the exchanged data, such as its volume and life span. In this setup the Design Server has become more an application on its own; ready to serve others, but not strictly mandatory to use.

Finally, for the actual transfer of data over the bus a dedicated XML-based format was developed which has been baptized the Digital Design Envelop, and from which details are depicted in fig. 12. However, at present such a DDE has not yet been completely specified, and its implementation will be a subject for future development.

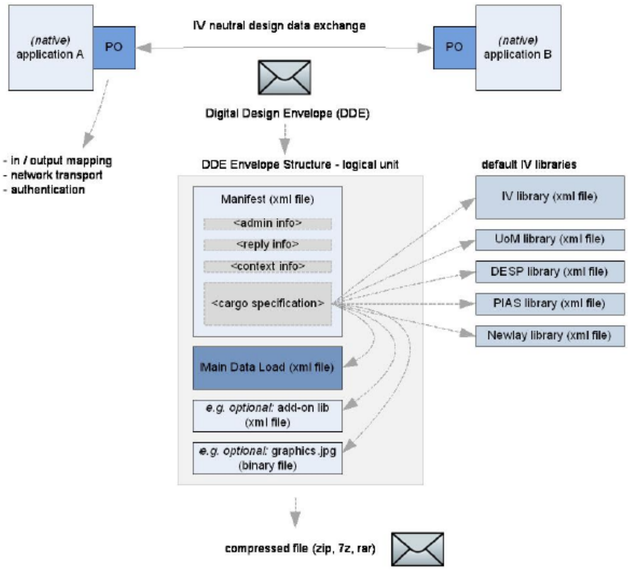


Figure 12 Digital design envelop

6. COOLABORATIVE APPLICATION

In collaboration with two partners[9] from the Innovero consortium, experimental software implementations have been produced, consisting of a) The BSP-based internal hull modelling tool, b) a provisional XML dictionary and c) a general CAD system. In this stage of development the Design Server was not yet employed for the purpose described in this section, but it will be the next development step to do so. Because the two partners use different CAD systems, this confederation of tools was prepared for two CAD systems; one for the Eagle system (http://macrovision.ie/), and one for Rhinoceros, www.rhino3d.com[10]. Although the functionalities of Eagle and Rhinoceros are certainly different, with both CAD systems a similar confederation of software systems was achieved, enabling transfer of the internal geometry from our BSP-based program to the CAD system, and vice versa, see fig. 13 for an example with Rhinoceros.

7. VALIDATION OF THE METHOD

Validation and testing is an important issue for each new method or software tool. However, the development stage of the Innovero platform as such has not yet reached the stage of maturity as required for end-user tests. But the BSP-based internal ship shape method is somewhat further; although its implementation is still too immature for application in a production environment, valuable lessons could be learned by incorporating end-users, ship designers, into the development process. For this purpose the following exercises were undertaken:

- A user workshop was held in February 2011, where the new implementation has been presented and discussed with some key potential users.
- The experimental software implementation has been distributed to the attendants of the

[9] Notably the design department of the Royal Netherlands Navy, and Conoship International, a designer of commercial vessels, www.conoship.com.

[10] The fact that this work was done for two CAD systems was born out of practical considerations, but it does underline the generality of our approach.

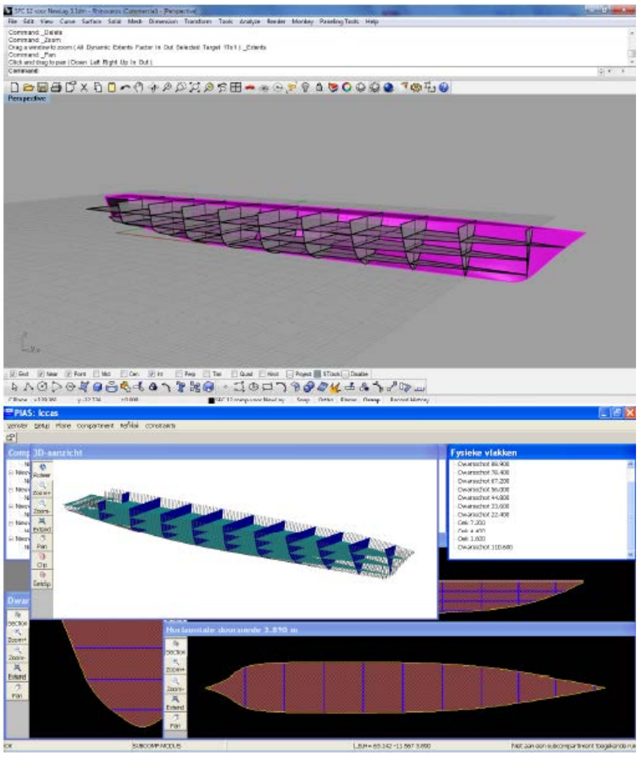


Figure 13 Rhino model (top) transferred to our BSP-based implementation (bottom).

workshop, so they could test the new approach with their specific ship designs.

- From a database of existing ship designs, some characteristic instances have been (automatically) converted to the new representation, thus verifying the stability of method and program.

In the first place, a vast number of programming bugs showed up from these experiments. Although annoying, these do not affect the method as such, and can be fixed with some programming effort. Other issues are the new ideas and enhanced functionality proposals which sprout of novice users when they are exposed to such a new software tool. For a software developer chasing after all and each of those ideas is impracticable, not only because the number is simply too large, but also because it would lead to an inconsistent software tool in the end. For this reason the adopted software enhancement procedure is:

- If anybody has a suggestion for enhanced functions, or an interface gadget, it can be put on a list.
- In a combined meeting of end-users and developers all these suggestions are presented and discussed. Generality of solutions, and merging of multiple proposals for functions into a single 'superfunction' are encouraged.
- All agreed enhancements are put on a list, and every person involved gets voting points, exactly as many as there are subjects on the list.
- Each person can distribute its voting points over all subjects, each subject can be allocated as many points as desired, but, obviously, the total number of allocated points should not exceed the maximum number[11].
- In the end, all voting points are added, and the subjects are ranked according to their points. Inprinciple the implementation will be performed in the sequence of this list. However, external factors may of course interfere with this idealistic scheme.....

8. CONCLUSIONS AND FUTURE WORK

In this paper the problem of the internal layout of a ship is addressed. In the first place, the need for a new modelling method is argued, requirements for such a method are postulated, and a solution is proposed. It appears that a Binary Space Partitioning method might provide a good solution, such a method was implemented and preliminary experiments suggest that it may work satisfactorily. Another important issue is the application of this subdivision method into a communication framework with other applications. For management of data and design rules the Design Server concept was proposed, implemented on a provisional basis, and found to be an attractive concept. The low-level communication was implemented based on an ad hoc XML standard, but this concept has been generalized to a structure for a dictionary. Finally, we learned the lesson that a bus-like structure of applications would provide more flexibility than one arranged around a central Design Server.

Although the proposed methods, and their cooperation, seem to satisfy the needs, a number of topics for future enhancements have been identified:

- Support for planes with a certain degree of nonplaneness (such as warped planes).
- Overlay a plane with regions of a specific property, such as plate thickness, watertightness, fireresistance etc.
- Re-use of a subdivision layout, or a part of it, of previous designs.

[11] In politics such a method has been proposed as a solution to improve the democratic outcome in a population which is more or less homogeneous, but also includes *passionate minorities*.

- Extend the XML dictionary.
- Finalise the specifications of the Digital Design Envelop, create a software interface for it, based on the proposed bus structure, and merge that with applications such as the CAD system and our BSP-based layout module.
- Import functional building blocks[12] and convert those, by means of the BSP-method, to planes & spaces. This may facilitate a system where the initial compartment topology is generated based on properties and requirements of functional building blocks (e.g. as proposed in (van Oers et al., 2009)), and which is subsequently converted to a conventional plane-space configuration.
- Apply the BSP structure to generate a lay-out on a statistical basis. In short, such a method could comprise the following elements: a) classify an internal topology, b) derive statistical topology rules and c) generate new designs.
- A recursive BSP, so that spaces can be subdivided further. This will allow for multiple levels of subdivision, not only watertight compartments, but also larger blocks, such as functional blocks extending over multiple compartments or smaller spaces, such as single cabins.

REFERENCES

[1] Alonso, F., Gonzalez, C., Pinto, A., & Zurdo, F. (2008). Integrated management of shipcompartments. In V. Bertram & P. Rhigo (Eds.), 7th International Conference on Computer Applications and Information Technology in the Maritime Industries COMPIT 08 Liege, Belgium.

[2] Bons, A., Koelman, H. J., van Oers, B. J., & Takken, E. H. (2011). Innovero design server applied on naval ship design. In Proceedings of the 15th International Conference on Computer Applications in Shipbuilding (ICCAS) Trieste, Italy.

[3] Buchele, S. F. & Roles, A. C. (2001). Binary space partition tree and constructive solid geometry tree representations for objects bounded by curved surfaces. In Proceedings of the Thirteenth Canadian Conference on Computational Geometry.

[4] Comba, J. L. D. & Naylor, B. F. (1996). Conversion of Binary Space Partitioning Trees to Boundary Representation, chapter Geometric Modeling: Theory and Practice: Chapter II - Representations, pp. 286–301. Springer Verlag.

[5] de Berg, M., van Kreveld, M., Overmars, M., & Schwarzkopf, O. (1998). Computational Geometry. Berlin, Germany: Springer.

[6] de Koningh, D., Koelman, H. J., & Hopman, J. J. (2011). A novel ship subdivision method and its application in constraint management of ship layout design. Journal of Ship Production and Design, 27(3), pp. 137–145.

[7] Ghali, S. (2008). Introduction to Geometric Computing. London, UK: Springer Publishing Company.

[8] Ghali, S. & Smith, C. (2005). Computing the boundary of a class of labeled-leaf bsp solids. In CCCG: pp. 159–162.

[9] Gielingh,W. (2008). An assessment of the current state of product data technologies. Computer-Aided Design, 40(7), pp. 750–759.

[10] Jeong, S. K. & Ban, Y. U. (2011). Developing a topological information extraction model for space syntax analysis. Building and Environment, 46(12), pp. 2442–2453.

[11] Jiang, Y. D. (1996). Block model conversion using binary space partitioning trees. Comput. Geosci., 22(4), pp. 409–417.

[12] Koelman, H. J. (2003). Application of the H-rep Ship Hull Modelling Concept. Ship Technology Research, 50(4), pp. 172–181.

[13] Koelman, H. J. (2006). Reflections on the implementation of boolean operations with polyhedral solids. In I. Horváth & J. Duhovnik (Eds.), the Sixth International Symposium on Tools and Methods of Competitive Engineering - TMCE 2006 Ljubljana, Slovenia: pp. 371–380.

[14] Koelman, H. J., Horv´ath, I., & Aalbers, A. (2001). Hybrid Representation of the Shape of Ship Hulls. International Shipbuilding Progress, 48(3), pp. 247–269.

[15] Lee, K.-Y., Lee, S.-U., Cho, D.-Y., Roh, M.-I., Kang, S.-C., & Seo, J.-W. (2003). An innovative compartment modeling and ship calculation system. In A. D. Papanikolaou (Ed.), Proc. 8th International Marine Design Conference Athens, Greece: pp. 683–694.

[16] Lee, S.-U., Roh, M.-I., Cha, J.-H., & Lee, K.-Y. (2009). Ship compartment modeling based on a non-manifold polyhedron modeling kernel. Advances in Engineering Software, 40(5), pp. 378–388.

[12] A *functional building block* is a 3D region of the ship that has one specific function. It does not necessarily have to coincide with a waterlight compartment, but may include multiple compartments. Examples are a “passenger space or a “machinery space”.

[17] Lysenko, M., D’Souza, R., & Shene, C.-K. (2008). Improved binary space partition merging. Computer-Aided Design, 40(12), pp. 1113–1120.

[18] Mäntylä, M. J. (1988). An Introduction to Solid Modeling. Rockville, MD, USA: Computer Science Press.

[19] Naylor, B. F. (1992). Partitioning tree image representation and generation from 3D geometric models. In Graphics Interface ’92”: pp. 201–212.

[20] Naylor, B. F. (1998). A tutorial on binary space partitioning trees. In Proceedings of Computer Games Developer Conference: pp. 433–457.

[21] Naylor, B. F., Amanatides, J., & Thibault, W. C. (1990). Merging bsp trees yields polyhedral set operations. In SIGGRAPH ’90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques New York, NY, USA: ACM pp. 115–124.

[22] Schneider, P. J. & Eberly, D. H. (2003). Geometric Tools for Computer Graphics. San Francico, CA, USA: Morgan Kaufmann.

[23] Smith, J. M. & Dodgson, N. A. (2007). A topologically robust algorithm for boolean operations on polyhedral shapes using approximate arithmetic. Computer- Aided Design, 39(2), pp. 149–163.

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

[25] Thibault, W. C. (1987). Application of binary space partitioning trees to geometric modeling and ray-tracing. PhD thesis, Atlanta, GA, USA.

[26] Thibault, W. C. & Naylor, B. F. (1987). Set operations on polyhedra using binary space partitioning trees. SIGGRAPH Comput. Graph., 21(4), pp. 153–162.

[27] van den Bergen, G. (1999). Collision Detection in Interactive 3D Computer Animation. PhD thesis, Eindhoven, The Netherlands.

[28] van Hees, M. (2009). Quaestor: taxonomy-based compositional modeling and product configuration. In S. E. Ovestad (Ed.), 10th International Marine Design Conference Trondheim, Norway: pp. 630–647.

[29] van Oers, B. J., Stapersma, D., & Hopman, J. J. (2009). An optimisation-based space allocation routine for the generation of feasible ship designs. Ship Technology Research, 56(1), pp. 31–48.

[30] Vanecek, G. (1991). Brep-index: A multidimensional space partitioning tree. In SMA ’91: Proceedings of the First Symposium on Solid Modeling Foundations and CAD/CAM Applications: pp. 35–44.

[31] Veelo, B. & Koelman, H. J. (2011). Non-disruptive development of a next-generation cad application program. In V. Bertram (Ed.), 10th International Conference on Computer Applications and Information Technology in the Maritime Industries COMPIT 11 Berlin, Germany: pp. 359–368.

[32] 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.