Modular design system re-galvanized

Programme structure, data exchange and pipe modelling are the main focus of SARC's major PIAS software update, writes Herbert Koelman, founder and managing director of SARC

A seminal version of PIAS was released in the last week of 2016, marking the beginning of the end for the ongoing large-scale renewal of the software, which started in 2011. The core design of PIAS originates from the early 1990s. While the core functionality of design and computation functions have frequently been enhanced and kept state-of-the-art, the user interface was deemed to require a major overhaul following over 20 years of service.

PIAS was originally subdivided into many distinct modules, each for a separate task, such as hydrostatics, cross curves, Bonjean, damage stability etc. The new programme structure contains a much more concentrated set of modules – shown in its main menu (Figure 1) – such as for hull design, design of internal layout (compartments, bulkheads and decks), and loading still & intact & deterministic damage stability. Beyond this core set, PIAS retains auxiliary modules, including for freeboard, tonnage, maximum allowable AHTS vessel anchor chain forces, and inclining test reports. Hopper dredger stability is still a separate module, but will be integrated into the intact stability module during Q1 of 2017, marking a final step in the software integration process.

However, why subdivide the software into modules? Why can't there be just one user interface, a single point of contact between user and software? The simple answer is that a program as comprehensive as PIAS contains so many features, functions and computation settings that it would be impossible to present them all on one monitor, and secondly that it would be utterly confusing for the user if it was achievable. After all, such a 'design' of the interface would include a blend of everything: surface modelling tools, sounding pipes, pendulum strokes during an inclining test, detailed settings of importing DXF files, specifics of intermediates stages of flooding, etc. So, although there is no software-related reason to organise the software in modules, for the benefit and overview of the user it is better to do so.

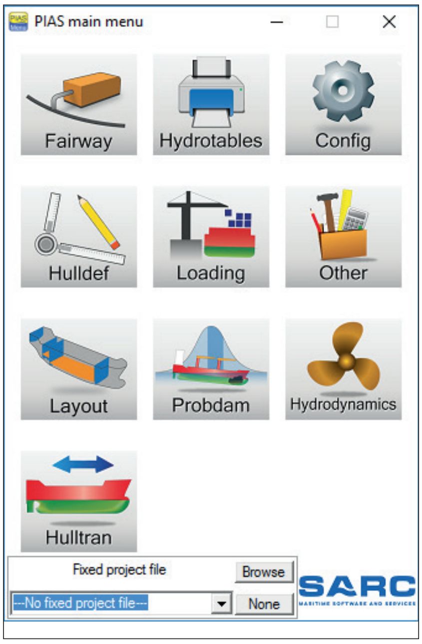


Figure 1: PIAS's new program structure as represented by its main menu

It could be argued that one drawback of such a modular design would be the lack of harmonisation between the modules, but PIAS tackles this by sharing all of a ship's design data between modules. This facility – baptised *local cloud*, and scheduled for release later in 2017 – enables a hull form modification to be directly (directly!) processed in the assessment of a loading condition against the applicable stability criteria. The local cloud permanently synchronises data in the background, hidden away from the user. This offers the best of both worlds: a single data model shared by the entire program *and* a well-arranged program focused on the main ship design tasks.

Apart from this structural reform, the opportunity has been used to enhance the software with:

- An integrated manual, which pops up with the corresponding section if the users hit F1 in a particular menu
- More use of dedicated Graphical User Interfaces
- A cleaner set of functions. In the past, SARC was occasionally seduced to include, on user request, some very specific functions that were sometimes applicable to only a single user or project. In hindsight, the program overview is sometimes better without, so some of them have been removed (or made invisible to the general user).

Conventional data exchange

The primal form of data exchange is by means of 'the file'. Although many data sharing alternatives exist, the file concept remains appealing for many people. One reason may be that a file is somewhat tangible; it can be copied, encrypted, stored on a USB stick or taken home. So, although file-based data exchange might not always be the most optimal alternative, it continues to be in vogue. PIAS already contains file-based import and export facilities, and this set-up is constantly being extended.

Recently, a pre-existing but separate function to import hull shapes from DXF or IGES formats was integrated into PIAS's Fairway hull form modeller. This is less trivial than it might seem; Fairway uses a genuine solid model, while DXF and IGES formats generally only contain curve or surface representations. This incompatibility has been solved by extending the capability of Fairway to manage unconnected curves, which are used to contain the imported curve shapes. If this set of curves has some coherence and connectivity, they can be converted into a solid model automatically so that the whole shape model is ready to be further processed by Fairway. The newly added functionality is also available without an imported hull shape, and offers more topological freedom at the very first hull design stage. Figure 2 shows an example of the Fairway GUI just after importing a set of IGES surfaces.

The export of surfaces to IGES has been the topic of some drama at SARC. After having reached the conclusion in the 1990s that NURBS surfaces were essentially unsuitable for effective ship hull design – the reasons for which have been duly reported in literature – SARC developed an improved method, which found its place in the Fairway modeller. However, a wide variety of tools are used in ship design, and so in areas such as engineering, CFD and FEM, importing in the NURBS format remains the preference. NURBS consequently re-enter through a backdoor, even though they are still sub-optimal. To cope with this, Fairway can convert its internal representation into a NURBS model and export it through the IGES standard.

This ultimately results in a vast number of small NURBS surfaces, and many software packages have problems with the quantity. Manufacturers play the ‘blame game’ – “you should accommodate an unlimited number of surfaces” vs. “the number of surfaces is too large you should reduce it” – while the issue hampers fluent data transfer. In order to ease this process, Fairway is at this moment being equipped with a postprocessor where the user can draw larger four-sided regions (a pre-requisite for NURBS) on the hull. These regions are being converted to NURBS by means of a special algorithm, which produces a set of NURBS surfaces guaranteed to be gap-free. An additional research project is being commenced where methods are evaluated for an automatic subdivision of the hull surface in four-sided regions. All these methods share the ultimate goal of making collaboration with other design-support software as hassle-free as possible.

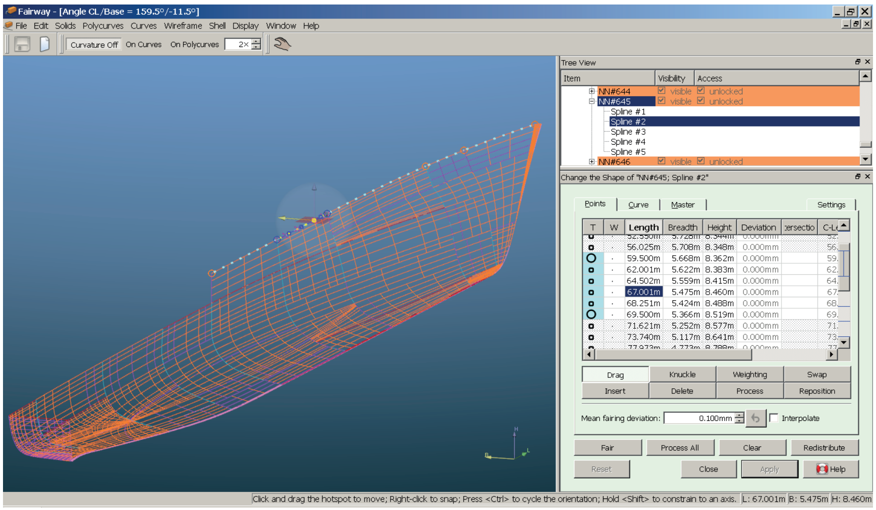


Figure 2: An example of the Fairway GUI just after importing a set of IGES surfaces

A third new export function is to Poseidon, DNV GL’s scantling program. This takes the PIAS hull form, decks, bulkheads, compartments, bending moments and additional data such as local loads and girders, and translates them into Poseidon import format. An example of a PIAS-originated model that has been visualised in Poseidon can be found in Figure 3.

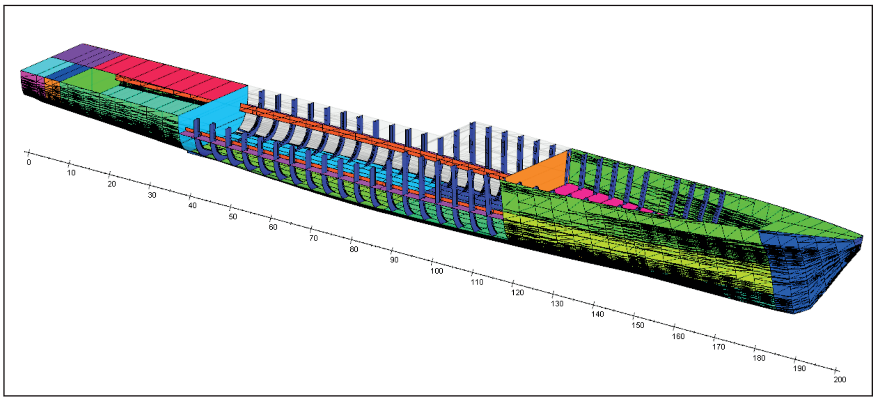
Non-conventional data exchange

As discussed in the *The Naval Architect*’s April 2015 issue (p.40-42), for decades the prevailing software architecture for collaboration has been a neutral data model, preferably stored in a central place that all participating computer programs can utilise. Unfortunately, such a design proved hard to conceive in practice. This is not an ideological statement, but simply the conclusion after decades of initiatives with this method and the fact it has not led to a prevailing standard. For example, the much-trumpeted STEP standard has achieved some endorsement, but it has not matured into *the* data exchange standard. A reason for this might be the so-called “representation variation”, which is the fact that geometric entities can be represented in multiple ways. For example, a circle can be described by centre and radius, or by three points lying on the circle. Likewise, a (hull) surface can be represented as a set of NURBSs, a point cloud or a set of curves.

Standards such as STEP or IGES support many of these alternative representations, which seem ‘friendly’ towards its users, but they require the importing programs to support all of those formats, and to convert them into internal representations. Where some variants can readily be converted, such as the circle example, others are non-trivial or cannot be done without loss of accuracy, such as from a point-cloud to a NURBS surface. Additionally, such standards have the habit of breaking down ship models into elementary constituents. The result of these two tendencies leads to an exponentially growing number of parts and representations, which requires a significant effort to manage.

It could be that the latest PLM solutions offered by major CAD vendors will provide

Figure 3: A PIAS-originated model that has been visualised in DNV GL’s Poseidon software



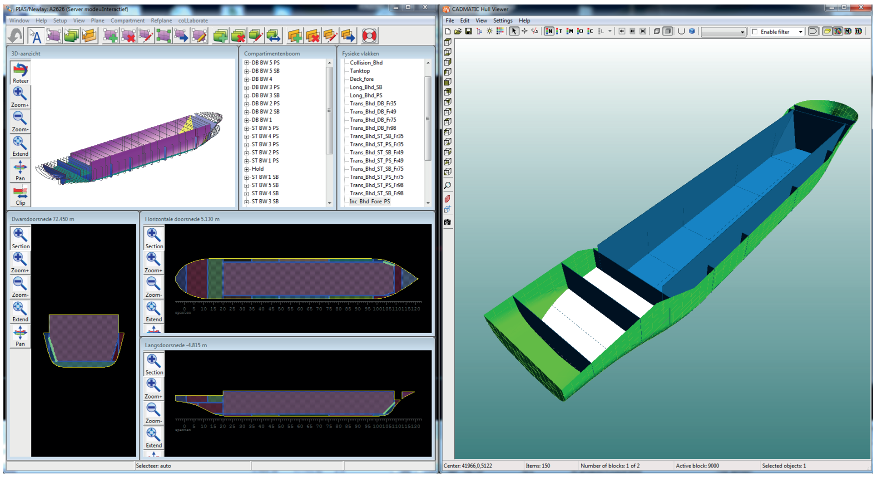
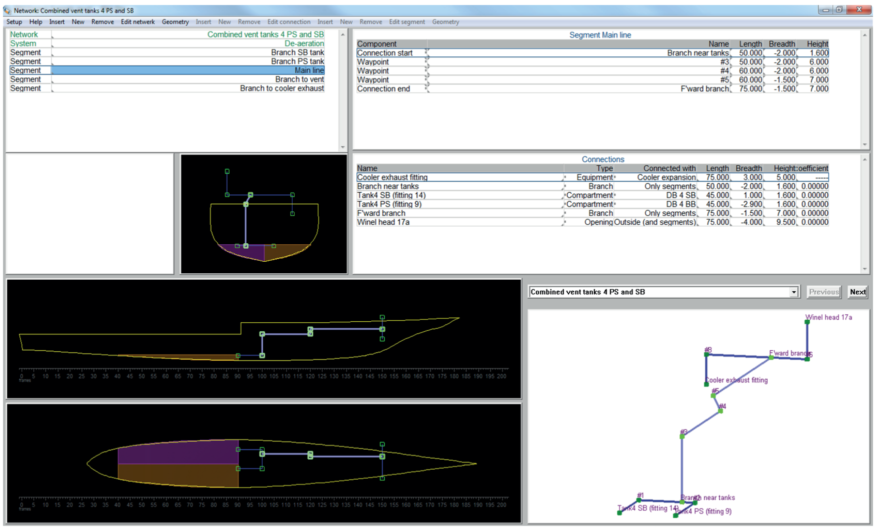


Figure 4: PIAS (left) and CADMATIC Hull (right) in joint collaboration

a solution to this pitfall, but SARC proposes an alternative based on three principles:

1. **Use of higher-level entities**
   In the conventional approach, a ship model is broken down rigorously, e.g. a 'bulkhead' will be broken down into plates, stiffeners, girders, brackets and welds, which is for design purposes much too detailed. Alternatively, by simply agreeing on the concept of a 'bulkhead', different software suppliers can exchange essential design particulars – such as its extent, position and watertightness – without sharing its internal representation.
2. **Sharing data, but tasks too**
   By this facility several applications can share their capabilities without sharing all underlying data. If, for example, a table of tank volumes has to be included into a general arrangement plan, the CAD system producing this plan could compute this table, but it would require a full geometric model of hull and compartments, as well as all computation logic. If, on the other hand, a hydrostatic software package is available as a collaborative partner, the CAD system can ask that package to perform this computation and just send the result. Such an approach will strongly reduce the software development effort.
3. **On the fly communication between the collaborating applications over the network**

Figure 5: Piping GUI under development in PIAS



At this moment a collaborative system based on this design is under development by Conoship International ship designers, CADMATIC and SARC. A specific feature of this system is that it deliberately does not produce a single common ship design model. The reason for this lies in the fact that many sidesteps are taken exploring the properties of design variants throughout a ship's design. For instance, moving a bulkhead in the stability software just to see the effect it has on probabilistic damage stability, does not need to be processed by all of the connected systems. In this way, the conceived system is designed to manage this diversity, and is, in fact, much more complicated than maintaining uniformity by means of rigorous permanent synchronisation.

Integration of piping in design

Connections between compartments, as well as internal openings, play an ever larger role in the assessment of damage stability. PIAS has been modelling this by so-called "compartment connections" for 20 years, but there has been a drawback: the real geometry of the connections is not available; they are more or less 'virtual' connections. This issue has led to a complete redesign, resulting in an implementation where the real shape and connection properties of the piping system are fully available in PIAS, including components such as pressure relief valves or vent check valves. Once finalised, the information is made available for three purposes:

- To be used in the calculation of deterministic and probabilistic damage. With this new data structure the effects of cross-flooding can be integrated into the probabilistic damage stability calculation.
- To be communicated with other computer programs, such as in the collaborative system with CADMATIC.
- To be utilised in the LOCOPIAS onboard loading software for enhanced damage assessment purposes and, in the future, time-to-evacuate analysis. *NA*