SIEMENS

Siemens PLM Software

Providing the next-generation design paradigm for shipbuilders

White Paper

Specialty marine computer-aided design (CAD) and mainstream product lifecycle management (PLM) systems have their respective strengths, but neither one is able to fully cover the needs of both design and configuration management for a massive product such as a ship. The 4th Generation Design (4GD) technology uses a component-based design approach to provide a design and data management paradigm that combines the best of both systems, enabling shipbuilders to meet the demand for increasingly complex vessels in an ever more competitive environment.

Answers for industry.

A white paper issued by: Siemens PLM Software.

Contents

Executive summary

Current commercial shipbuilding systems tend to fall into one of two categories. The first are specialty marine CAD systems that provide multidiscipline design tools for designing ships composed of millions of components, but do not provide a rich data lifecycle environment and have limited, if any, configuration management capabilities. The second are mainstream PLM systems that provide rich end-to-end product lifecycle solutions supporting robust configuration management and manufacturing integration, but are ill-equipped to handle the sheer size and data volume of ships. In this paper we will discuss 4th Generation Design (4GD), an advanced approach that when built into PLM systems exhibits the benefits of both categories, enabling an all-encompassing PLM shipbuilding environment for massive product design.

This paper is organized in the following manner: First, it is a discussion of the challenges faced by shipbuilders using systems in the two categories described above. Second, we present the new advanced approach that combines the richness of PLM data management with the effective design environment of specialty marine CAD. This approach requires a 4GD technology built into PLM systems to manage ship data at a component-based level. Finally, we discuss how each of the challenges is addressed with 4GD.

Facing the challenges of ship design without a rich data management backbone

The demands of ship design require a rich PLM data management backbone, including:

- End-to-end management of all PLM data
- Workflow automation and efficient change management
- Versioning and controlled evolution of product data
- Design re-use across ships
- Historical configuration
- Side studies
- Open format for exchange of product data

PLM challenge 1: end-to-end management of all product lifecycle data

One of the fundamental strengths of a rich PLM backbone is the ability to seamlessly support the evolution of the product definition across the product's entire lifecycle. Capabilities such as managing conceptual design and early requirements definition; project planning and tracking; change management; manufacturing planning and post-delivery and in-service product definition are all key PLM strengths.

Throughout the product lifecycle, the configuration of all product data and any other related information is managed in the database, governed by established processes and business rules rather than stored in an unmanaged environment with ad hoc processes or access controls. Specialty marine CAD systems typically have only limited support for such capabilities, which are foundational to a rich PLM backbone.

It is beyond the scope of this paper to discuss each of these PLM aspects in detail, so the following paragraphs discuss only one critical product definition phase: how to develop a seamless connection from design to manufacturing for product planning and production.

Typically, a design or engineering bill of materials (BOM) is organized in a way that is relevant and optimized for engineering. Often, there is a different organization of the product information that is optimal for manufacturing planning. Also, there is typically domain-specific information that applies only to design or to manufacturing (or other disciplines).

PLM systems give the necessary flexibility to the manufacturing planning organization to construct and manage a manufacturing planning bill of materials (PBOM) that is based on and associative to the engineering bill of materials (EBOM, which is essentially the CAD data). It allows the PBOM to be flexibly organized, and enables metadata to be added to capture manufacturing planning-specific information. Additionally, a PLM system provides accountability checking to visually validate that all required parts from the EBOM have been consumed into the PBOM.

The ability to seamlessly enable the product definition to be leveraged by both engineering/design and manufacturing planning while allowing both groups to organize the product and capture the information that is relevant to their domain without affecting the other, is based on a PLM backbone's ability to maintain associativity across the two domains as this parallel evolution takes place.

PLM challenge 2: workflow automation and efficient change management

Flexible workflow support ensures that validation, review and release processes are consistently followed throughout the enterprise. Performing these validations is executed by the system per the prescribed business process rather than performed manually by individual users.

Workflow in PLM systems provides a clear way to define flexible processes that are guaranteed to be applied consistently as the product data evolves. This includes ensuring that the correct authoring, review and approval steps are followed, that the appropriate users are involved in each, and that all of the data criteria are met before different milestones or maturity can be achieved.

Similarly, change management is a critical capability to ensure controlled evolution of product data and authorization of work according to business rules. Change management enables the business to control the issuance of change authorizations to perform work, and to release new product data. This, in turn, ensures that only a minimum amount of effort gets expended until the business validates the intent to proceed with the proposed design work. For example, if the need for a significant design change is identified, the business would like to have the proposed change and its expected impacts elaborated on and approved by key stakeholders before proceeding with the complete detailed design.

Leveraging change management in conjunction with workflow ensures a repeatable business process is followed with the appropriate amount of rigor and oversight based on the impact, cost, urgency, or other factors relevant for the work being done. Specialty marine CAD systems typically have limited support for robust workflow automation and change management.

PLM challenge 3: versioning and controlled evolution of product data

One key principle of configuration management is the ability to control the evolution of data as it changes over time. It is critical that a user is able to understand the lineage of an object across multiple iterations (called revisions or versions). Likewise, it is necessary to be able to clearly determine what iteration was relevant for a specified level of maturity at a given point in time or for a specific ship.

Versioning and the ability to configure data based on maturity or hull effectivity (effectivity is a common term in PLM that may not be widely known in shipbuilding. Specifying effectivity on a design enables it to be explicitly sanctioned for re-use on only the specified ships/hulls. This is a powerful method of enabling controlled re-use over cloning) is one of the powerful foundational elements of configuration management. It enables the ship design to evolve in a controlled way that facilitates the checking and approval processes and maintains a historical record of the design evolution. In addition, it forms the basis of flexible configuration management in which the ship can be configured for combinations of in-work, partially-approved and released designs. Specialty marine CAD systems typically lack the capability to evolve and re-use design data across multiple ships in a controlled way without data duplication.

PLM challenge 4: design re-use across ships

Although two ships are rarely if ever identical in every respect, a shipbuilder will try to leverage designs from previous ships as much as possible to reduce costs and risks. Historically, this re-use for shipbuilders has been limited, and typically involves an approach based on cloning rather than true re-use.

PLM configuration management capabilities support re-use by enabling a single design to be managed for an entire family of ships. Design content that is relevant for one or more (but not necessarily all) ships in the family can be readily qualified so only the relevant data is configured for each ship. For data that are in fact common and directly re-usable, no copying or cloning is required.

A rich PLM backbone supports starting a new product design configuration from a previous ship. Unlike cloning, which creates an entirely new copy of the entire ship's design data, carryover leverages hull effectivity and effectivity intents. This enables users to accept or extend a previous ship's design to the current in-work ship when appropriate, and to introduce variations only where needed. In this way, the system promotes design re-use, while still requiring explicit acceptance to certify if a previous design meets the needs of the new design.

Specialty marine CAD systems need the functionality to support multiple ships or a class of ships in a single definition that can be extended and modified as new ships are developed without requiring duplication of common data.

PLM challenge 5: historical configurations

The certification and delivery of a new ship often requires different subsystems or even the entire ship design to be reviewed and formally accepted at different design milestones. Capturing the precise state of the design at these business significant milestones are called configuration baselines.

By creating a baseline, a user or group of users can capture an immutable snapshot of the design exactly as it existed at a specific point in time for a given level of maturity for the ship. This baseline can be retrieved at any point in the future as a concrete record of what the product design was at that particular milestone. Multiple configuration baselines or iterations of a given configuration baseline can be tracked over time so you have a clear and unalterable record of each significant milestone.

Configuration baselines can be associated with contract milestones, formal design acceptance checkpoints or simply any point in time when a historical record of the ship design is needed. Specialty marine CAD solutions lack the ability to evolve design data over time and to capture a precise record of the design at given milestones for contract signoffs or stable historical records.

PLM challenge 6: side studies

As an organization identifies ways in which a ship design needs to evolve or be optimized, designers and engineers are tasked with proposing and evaluating alternatives to meet a ship's new or changing requirements.

There may be multiple users who are considering and proposing competing alternatives for a given requirement. Historically, there has not been a straightforward way for each user to work on their proposal while:

1. Maintaining awareness of any relevant surrounding design data as it evolves
2. Not exposing their preliminary work outside of their immediate workgroup. As a working practice, design groups have had to copy the relevant design data to a private workspace to insulate their preliminary/provisional work from other users.

This has several drawbacks, including:

1. Forcing designers to manually integrate their work into the master product definition later in the process
2. Forcing designers to manually update their private workspace with changes from the main ship design
3. Exposing other users and groups to immature or competing design alternatives to ensure that the designer has ready access to surrounding changes

With a rich PLM backbone, it is straightforward to allow one or more users or groups to work on provisional, possibly competing designs in a given area of the ship while ensuring that as each user or group is working on their design, they are always aware of any relevant surrounding reference data that may be evolving in parallel. Specialty marine CAD solutions can't be used to provide users with the ability to work collaboratively, or in parallel, on design alternatives without requiring the design data to be duplicated and the chosen design being manually introduced back into the product definition.

PLM challenge 7: open format for exchange of product data

A shipbuilder needs to exchange data reliably and flexibly with suppliers and partners, some of whom may use a different design authoring tool. Therefore, it is important to have a rich and open protocol for communicating design data in both directions. For example, Teamcenter® software from Siemens PLM Software offers the powerful open standard of JT™ for geometry management. Leveraging this open CAD-neutral and globally-accepted standard for data sharing and exchange, including geometry data, allows broad flexibility between a primary ship designer/manufacturer and its (likely numerous) suppliers. Leveraging JT along the exchange protocol, supported by what is called the Supplier Exchange Briefcase in Teamcenter allows suppliers to use their native design tools while still being able to share and exchange data (including supplier updates) with the primary ship designer. Specialty marine CAD solutions typically lack the functionality to accommodate multi-CAD design and flexible round-trip supplier data exchange.

PLM challenge summary

We can now list a set of PLM backbone capabilities that marine CAD systems typically lack when used for the design and manufacture of massive products like ships, including:

- Management of end-to-end product data across the product lifecycle
- Workflow automation and efficient change management
- Versioning and controlled evolution of product data
- Ship/hull effectivity/carry over within ship class or from sister ships
- Collaboration or working in parallel on alternative designs without duplication
- Stable product baselines for certification of contract milestones and support for historical configuration
- Support for multi-CAD design content and flexible round-trip supplier data exchange

Challenges of managing massive products in PLM systems

The massive amount of product data needed to represent a ship places some unique demands on PLM systems, including:

- Multiple data organizational breakdowns
- Efficient access to individual design elements
- Ad hoc design in context
- Concurrent access

The current approach for managing massive products in mainstream PLM/CAD systems relies on hierarchical product structures, also referred to as assembly structures. This approach provides management of individual parts and their usage in moderate-to-large products. However, using mainstream PLM/CAD systems for massively complex products like ships, which are composed of millions of parts and welds that must be managed individually, has revealed a significant number of data management challenges. These challenges have been addressed by successive generations of PLM/CAD technology as described below.

The first generation technology for managing design data used very coarse-grained collections of files, often referred to as globs. Each individual file contained all of the geometric models representing a section of the product, perhaps hundreds of models in a single file. However, as the scale and scope of products grew, so did the limitations of the method. Files couldn't be used to provide real re-use of standard geometry, offered limited concurrency for large design teams and had only very basic data management capabilities. For example, there was no efficient way of knowing that two objects in two different files actually represent the same design. This resulted in excess duplication of design data with no formal way to identify duplication or solve basic design problems, such as performing a global replace.

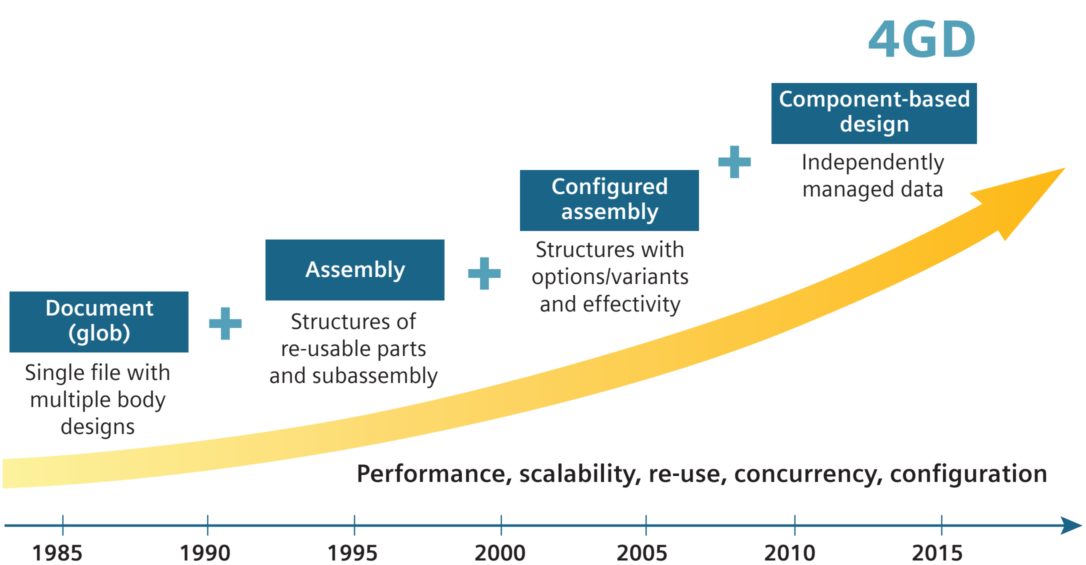

Figure 1: Evolution of large-scale design management.

The second generation technology introduced assemblies to overcome limitations of the glob approach. These assemblies could be used to enable commercial systems to successfully manage a much larger scale and scope of products than the initial glob approach. Assemblies encourage a single-part-per-file approach to design data. At the lowest level are unique part designs and re-usable parts that are instantiated into a hierarchy of assemblies and re-usable subassemblies.

As commercial product data management systems matured, they added another important dimension to assembly management: assemblies and subassemblies that could be revised under configuration control, and associated with configuration variability (such as options and effectivity). This third generation enabled the family of products to be defined and managed more efficiently. However, as the scale and scope of products continued to grow, several new challenges appeared in the assembly approach used in this third generation technology.

Assembly challenge 1: multiple data organizational breakdowns

Shipbuilders organize the contents of the ship according to several independent hierarchical breakdowns. Common data organizational breakdowns in shipbuilding include:

- System – An organizational breakdown of systems and subsystems, such as the fresh water system, electrical system and the heating, ventilation and air conditioning (HVAC) system
- Area – An organizational breakdown by location or zone of the ship, such as deck 1/room 1583
- Module – An organizational breakdown by modules and sub-modules. For ships that are constructed in a modular manner, modules are first constructed independently and then combined together in a final assembly process

Data organizational breakdowns provide alternative views into the same set of design data. For example, a holding tank may be simultaneously a member of the system breakdown (e.g. fresh water system), the area breakdown (e.g. room 2249) and the module breakdown (e.g. aft module).

Data organizational breakdowns are also used to browse or navigate to the data of interest, such as to find everything located in a certain room. They also help to provide context to data. For example, in a CAD application it is helpful to show the objects loaded into session in a tree view organized by the system, area or module hierarchy.

Unfortunately, the assemblies approach used by mainstream PLM/CAD systems supports a single organizational structure and does not easily adapt to the multiple organizational structures of data required by shipbuilders. This requires shipbuilders to compromise and define a single, primary organizational method and resort to workarounds for the others. As a result, shipbuilders manage and navigate data via an organization that is not optimal for everyone's needs. Mainstream PLM/CAD systems need a method that facilitates multiple organizational views without data duplication.

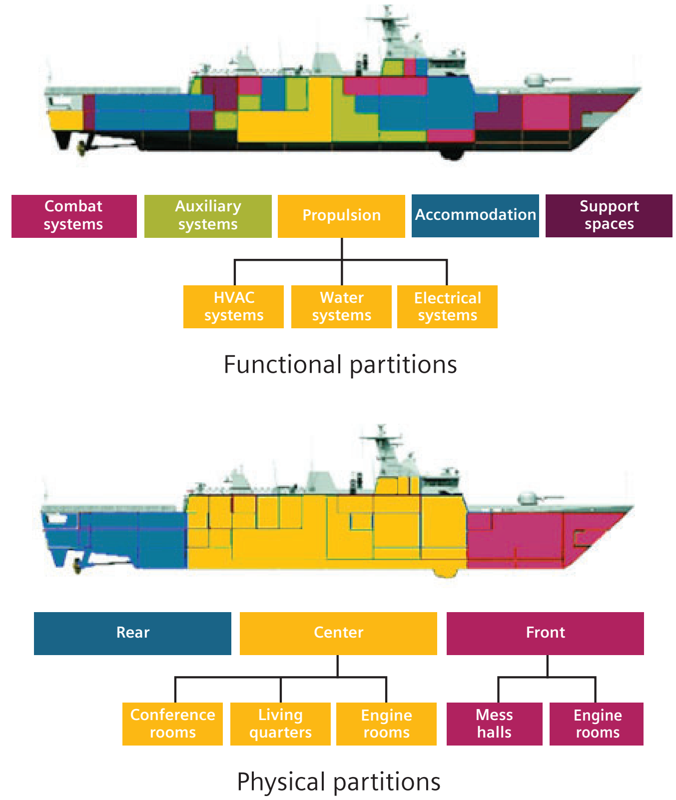

Figure 2: Example of multiple organizations of product data.

Assembly challenge 2: efficient access to individual design elements

Due to the massive amount of data comprising a ship, the hierarchical structure of assemblies and subassemblies in mainstream PLM systems becomes necessarily deep and broad. These huge assembly structures are also configurable (i.e. revision rules, baseline and unit effectivity). For PLM systems to give you the ability to access a particular design element in the product structure, it successively expands each level in the hierarchy, applying revision and configuration rules, and loading the children until all desired data is loaded. This approach works reasonably well for moderate-to-large products, but becomes problematic when applied to the scale and scope of data in a ship.

Fast searching based on spatial and/or attribute criteria across the entire ship is a critical need for shipbuilding design. And this searching must provide up-to-date results, as opposed to basing results on a cache that was harvested the night before. In the mainstream PLM system, up-to-date results are only achieved by configuring the assembly at the time the search is performed, which for a product the scale and size of a ship is an unwieldy and time-consuming task. This is a significant restriction on the viable use of mainstream PLM systems for day-to-day shipbuilding design needs, such as determining position, performing spatial searching and collision detection and overall query performance.

For example, to answer the question about the exact position of a component in ship coordinates, it is necessary to configure and load all structure nodes from the top-level node of the ship through all interim subassembly nodes, to the node of the component in question. All transformation matrices are then applied in order to arrive at the exact position of the component in ship coordinates. This method must be applied for every component that takes part in any ship query involving position. Given that a single hierarchy is used to organize design data, it is unlikely data that is functionally similar, spatially close in proximity and part of the same system will all be neatly grouped into a common branch of the assembly structure. The implication of spatial queries or collision detections and other operations that are critical for shipbuilding design is that very large portions of ship structure must be configured and loaded in order to perform fundamental ship design queries.

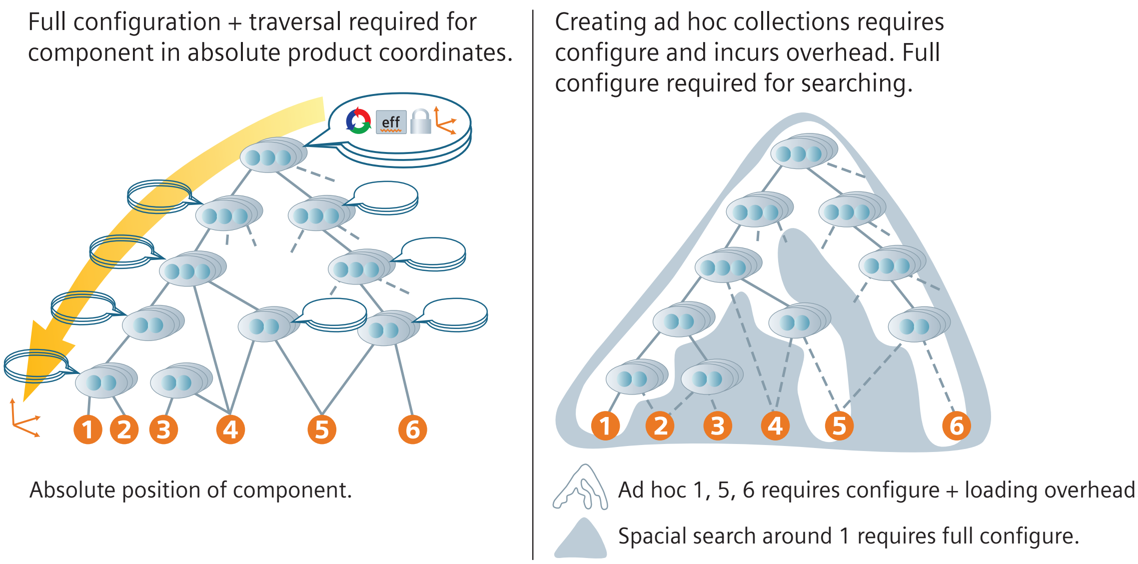

Figure 3: Implication of configurable structures on position and spatial searches.

Organizing by a single large structure also places restrictions on the ability to restructure, meaning moving a component from one branch of the assembly hierarchy to another. In shipbuilding, when top-down design is used, the structure must be defined well in advance of the actual design work. However, once design work is underway, sometimes it becomes necessary to introduce changes in that structure or move components from one branch of the structure to another. Restructuring, like computing position, must be done within the configured structure of the ship, and that means loading all nodes of the configured structure to the top-most common node of the restructuring operation.

Finally, query performance on configured hierarchical structures tends to degrade as the assembly structures become overly deep and broad. This is opposed to a flattened persistence of the data, which is more suited to leveraging optimized database retrieval techniques. For massive products, mainstream PLM/CAD systems need an alternative method for configuring large hierarchical product structures in order to perform common tasks.

Assembly challenge 3: ad hoc design in context

Although a ship is represented with a massive amount of data, only a fraction of that data is typically relevant for a user's particular design task. Even if desktop computing resources were powerful enough to load and display an entire ship instantaneously, users rarely need to do that. Instead, nearly every task requires only a subset of ship data needed to accomplish a given task, such as the detailed design, drawing creation, side studies or design reviews.

In addition, since the hierarchical structures of ships are very deep and broad, navigating these structures to find what data to load into a design session is impractical. Users need powerful tools that can quickly search across an entire ship and gather up for the user only what is needed. The search criteria must include a combination of object type, attribute value, zone (i.e., a fixed location like a room), proximity and more.

For example, to redesign the portion of a piping subsystem in a certain room, a user would likely request all objects comprising that piping subsystem plus any objects in that room within 1 meter of the subsystem. This subset of data could contain between 500 and 10,000 objects, which is large but certainly much smaller than the entire ship.

Once the user has found only the objects needed for the design task, the user wants to load these objects (and no others) into the CAD design session. However, in an assembly structure, these results may be from anywhere in the hierarchical structure and not necessarily neatly organized together, as mentioned above. So a much larger number of objects are loaded into the CAD session than are needed, including: all the desired objects, plus the co-objects of these objects beneath the same parent node, plus all interim sub-assembly nodes to the top-most common node, as well as other children of the subassembly nodes that are not necessarily relevant at all to the design task at hand.

Finally, since the design or analysis task can span many days or weeks, it is important to be able to refresh the ad hoc set of data to incorporate relevant changes made by other people. In the above search example (all objects within 1 meter of a certain piping system), the resulting subset of data should be updated if another user adds new objects in the vicinity of the piping system. But in assembly-based systems, it is challenging to differentiate between the subset of design data that is directly relevant for the design task, from the objects that were loaded as overhead due to the nature of the assembly structure itself. Thus, the updating process itself is prohibitively long, and spends resources updating data that is not relevant to the user.

Mainstream PLM/CAD systems need fast, robust query methods for identifying a precise collection of ship data needed to perform a design task in ship context, to load only that precise set of ship data into a client session and to quickly assess it if changes have been made to that set of data as time passes.

Assembly challenge 4: concurrent access

The nature of an assembly node in an assembly structure is to act as a container for all instances of its immediate children (pipes, welds, instances of standard parts, etc.). Therefore, an assembly node controls the ownership, position, lifecycle status, attributes and effectivity of all its instances, and more importantly, the instance cannot be managed independently of the assembly node. This poses challenges to collaboration and productivity for massive products like ships in the following ways:

- To modify an instance the user must lock the assembly node so another user cannot concurrently change anything else in the assembly node until the lock is released
- To approve and release a change to an instance requires that the assembly node also be approved and released, which results in an unnecessary amount of checking and reduced traceability into the exact nature of the change that was made
- To revise an instance, or apply unit effectivity to an instance, requires a change to the entire assembly, which results in a potentially large number of assembly revisions and a significant amount of redundant data in those revisions

Mainstream PLM/CAD systems need a method for managing ship design data so it can be accessed and controlled independently and individually. An independent object might be a structural steel member, a piece of pipe, a weld, an instance of a standard part (e.g., a flange), an individual length of HVAC duct or a specific wire hanger.

Assembly challenge summary

We can now list a set of needs that mainstream PLM/CAD systems require when used for the design and manufacture of massive products like ships:

- Facilitates multiple organizational views without data duplication
- Does not configure large hierarchical structures to perform common day-to-day tasks
- Facilitates fast, robust query methods for identifying a precise collection of design-in-context data
- Loads only a precise set of ship data into a client session
- Quickly assesses if changes have been made on that set of data as time passes
- Manages a granular level of design data

Challenge response: 4th generation design/ component-based design

Siemens PLM Software for Shipbuilding with 4th Generation Ship Design and Engineering (4GD) implements a new component-based design approach that overcomes the challenges discussed in the previous sections. And it delivers a PLM/CAD environment equitable to both categories discussed in the opening summary section: marine CAD for multidisciplinary teams with the rich/full data lifecycle environment of mainstream PLM systems; end-to-end PLM for massive and complex ships having millions of managed design data elements.

In component-based design, there is no product-level assembly structure. Instead, each managed object has a specific business purpose, is managed independently and has its own:

- Access privileges (i.e. read/write)
- Maturity status (i.e. in work, being checked, released)
- Position in product coordinates
- Set of attributes (e.g. pump operating pressure per each pump instance)
- Revision history
- Unit effectivity
- Locking status (checked out/checked in)

The Siemens PLM Software component-based design solution for marine shipbuilding is formally launched with the releases of Teamcenter 10.1 and NX™ software 9.0. By examining the 5 basic concepts of 4GD, it is possible to see how the Siemens PLM Software 4GD technology addresses each of the challenges identified in the previous sections. Note that it is beyond the scope of this paper to drill into a detailed discussion of 4GD concepts. Therefore, the level of 4GD explanations given is for the purpose of showing how the identified challenges have been met.

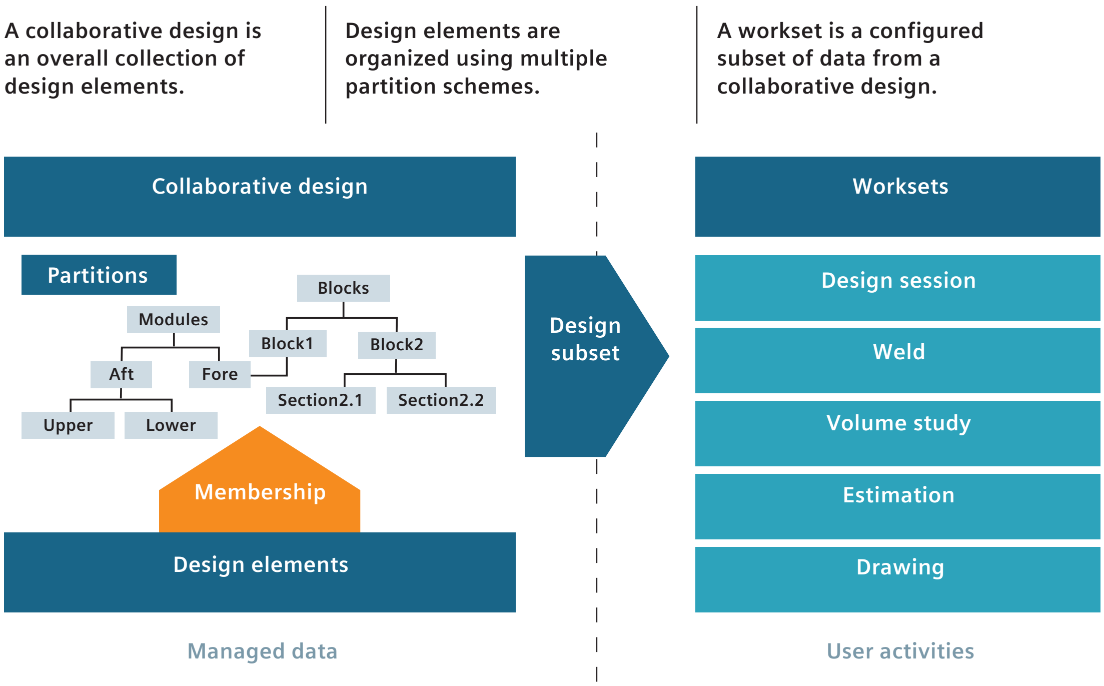

Figure 4: Main concepts of the Siemens PLM Software 4GD solution for component-based design.

Concept 1: collaborative design

A collaborative design is the single container for all the design data that comprises a ship or a class of ships. The design data is authored and modified by a multidisciplinary team of contributors, and is available for downstream use i.e., manufacturing.

Concept 2: design element

A design element is an independently managed occurrence of design data within a collaborative design. A ship has millions of design elements in which each design element might be a structural steel member, a piece of pipe, a weld, an instance of a standard part (e.g. flange), an individual length of HVAC duct, or a specific wire hanger. A design element is a declared member of a collaborative design – it is not a child of a collaborative design.

Each design element provides a specific business purpose, and satisfies the needs of a component-based design solution because it has its own:

- Access privileges
- Maturity status
- Position in ship coordinates
- Set of attributes
- Revision history
- Unit effectivity
- Locking status

Concept 3: partition

Partition hierarchies provide a logical way to organize the millions of design elements in multiple ways. They can represent functional organizations of the product (i.e. system/subsystem breakdown), physical divisions of the product (i.e. grand block/block breakdown, fire zone allocation, compartment breakdown) or any other desirable way that the shipbuilder wants to organize the design elements, such as equipment, task breakdown structure.

A design element may be a member of multiple partition hierarchies, and may also appear in more than one partition node within a single partition hierarchy. For example, in a system-based partition hierarchy, a generator is part of an electrical power system, generator cooling system and the fuel system. The same generator is also in the engine compartment zone in a zone-based partition hierarchy.

Partitions do not interfere with the design data of the design elements. Partitions are used only to organize the design elements into multiple views. In addition, partitions can be populated manually (i.e. user assigns a design element as a member of a specific partition), or partitions can be populated dynamically (i.e. a partition declares a design element to be its member because the design element lies within a specified volume area, or the design element has a certain attribute value defined).

Partition hierarchies are navigated as an indented tree, which provides a convenient and natural way to browse and navigate product data.

Concept 4: subset

Component-based product design enables searching and retrieving design data from across the breadth of the product without requiring top-down configuration and loading the entire product. The collector that is used to capture and persist the design data from the collaborative design that meets the user-applied criteria for a given design task is called a subset.

A subset defines a collection of design elements from a single collaborative design. Subsets are used to define what set of design elements are required in order to perform a task. The data might be used in any type of task, such as to author new design data; modify existing design data; perform an analysis or create a detailed drawing.

A subset uses recipe statements to define exactly what is needed by the user. The subset's recipe statement might define a single specific design element, such as bring the pump design element from deck 4, compartment 20. Or the subset recipe statement might define an ad hoc collection of design elements, such as bring all HVAC design elements located in compartment 121, or, bring all design elements of any type that are within 25 centimeters (cm) of the hydraulic system.

A subset also defines the configuration scope to use when searching for data based on the above recipe statements. The configuration scope determines which revision of a design element to return, and which unit effectivity is relevant. For example, bring the latest design revisions for ship number 3 (unit effectivity = 3).

Concept 5: workset

The workset is the environment where users within an application do their actual work in the context of the ship. Each workset contains one or more subsets, and is the place where users directly access those design elements that were captured by the subset. The workset is also the way that users define new design elements to add to a ship. In the workset, users manipulate, modify, navigate and visualize the design elements of a ship.

The workset supports individual authoring assignments, analyses assignments and review activities. Since the workset is a persisted object in the database that has its own lifecycle, it is also used to create drawings of single design elements or ad hoc collections of design elements such as systems.

Inside the workset, the collection of design elements returned by a subset does not automatically update. The user controls when to 'replay' the recipe terms of the subset, which in turn may add more design elements to the workset, or even remove some. For example, for a recipe term based on proximity, a new design element created by a different user would appear in the workset if it was positioned within the proximity defined.

4GD delivers a rich PLM data management backbone for ship design

The Siemens PLM Software 4GD component-based design solution is fully immersed in the Teamcenter PLM backbone. This effectively integrates shipbuilding-capable design tools within a rich PLM backbone, and provides the ability to effectively manage not only the design information for a given ship, but the whole product development process, including the design itself and all related information across the lifecycle for families of ships as they evolve. The advantages of the Siemens PLM Software 4GD component-based design solution are as follows:

- Management of end-to-end product data across the product lifecycle
- Workflow automation and change management
- Versioning and controlled evolution of product data
- Ship/hull effectivity/carry over within ship class or from sister ships
- Enable users collaborating or working in parallel on alternative designs without duplication
- Capture of stable product baselines for certification of contract milestones and support for historical configuration
- Support for multi-CAD design content and flexible round-trip supplier data exchange

4GD surpasses assembly structure-based design for shipbuilding

In the following sections, we revisit the needs that mainstream PLM/CAD systems require when attempting to manage massive assembly structures like ships, and discuss how the Siemens PLM Software 4GD solution for component-based design provides clear responses to the identified set of assembly challenges.

Managing a granular level of design data independently

The 4GD design element is an independently managed occurrence of design data within a collaborative design. The design element is a direct member of a collaborative design, and it contains all knowledge about itself (lifecycle state, position, effectivity, etc.). There is no overarching assembly structure to traverse in order to view, configure, access, manage or control its contained design data.

The design element is also a versatile object that is used by the business to decide the level of detail needed for each type of design data in the overall ship design. For example, each pump assembly unit placed on different ship decks can be its own design element; all steel plates throughout the entire ship can each be a design element; each weld can be a design element and even the surface hull definition can be managed as a design element. The shipbuilder has the option to decide the level of granularity for independently managing critical design data, and defines these as the design elements in the ship collaborative design.

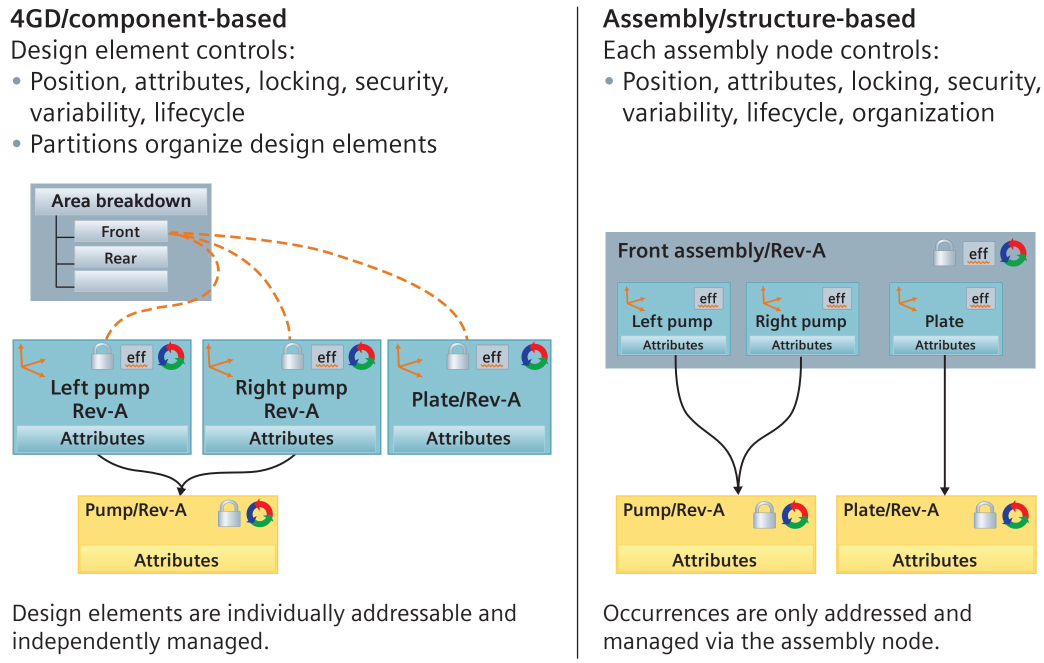

Figure 5: Component-based design enables managing occurrences independently.

Facilitating multiple organizational views without data duplication

4GD partitions provide a logical way to organize the millions of design elements in multiple ways. As mentioned above, they can represent functional organizations of the product (i.e. system/subsystem breakdown), physical divisions of the product (i.e. grand block/block breakdown, fire zone allocation, compartment breakdown), or any other desirable way to organize design elements.

More importantly, the customer is not required to define partition hierarchies prior to the start of a project or program. While some partition hierarchies may be fully understood at the start of a program, there is no penalty or design impact for defining a new partition while a program is already in progress, and no problem with extending and/or altering an existing partition hierarchy.

Does not configure large hierarchical structures to perform common day-to-day tasks

The millions of 4GD design elements that comprise the members of a collaborative design are essentially a large flat collection of design data. There is no configuring of an assembly structure or traversal of an assembly structure in order to understand the position and bounding box for a design element. Spatial searches have no need to configure an assembly structure prior to performing the spatial search itself.

Also, since partitions are completely separate from the design data of the design elements, changing the partition assignment of a design element does not require configuring and/or loading an overarching assembly structure, the opposite of what is required in a restructure operation in assemblies.

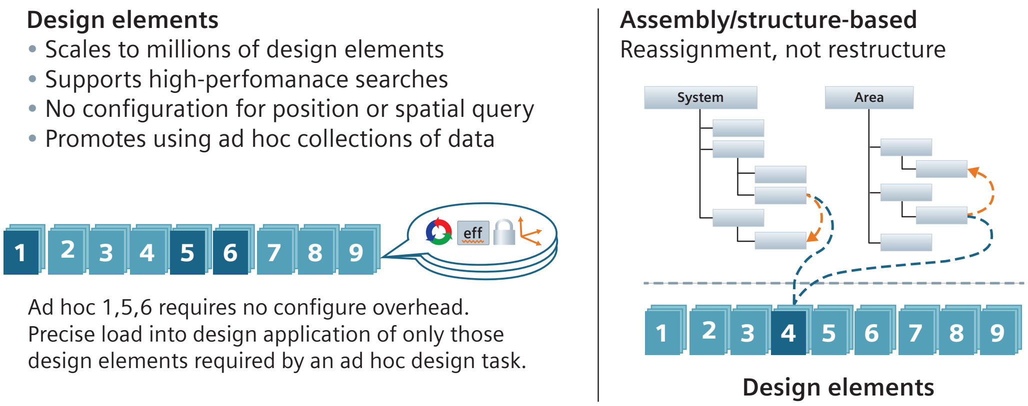

Figure 6: Component-based design in 4GD has no assembly configuration overhead.

Robust methods for providing design-in-context data

The 4GD subset provides a way for users to define the precise set of design data (design elements) required to fulfil a particular task. No more, no less. Furthermore, when inside a design application, the design elements captured by the subset definition are the only design data that are loaded. There is no assembly structure overhead associated with the loading of design data that might degrade the performance of the loading operation, or incur the loading of more data than the design application can reasonably handle.

Conclusion

As stated in the abstract, specialty marine CAD systems provide efficient multidiscipline design tools for millions of ship components, but do not have a rich/full data lifecycle environment. Mainstream PLM systems provide rich end-to-end product lifecycle solutions, but rely on assembly structures that are ill-equipped to handle the sheer size and data volume of ships. The component-based design approach as implemented in Siemens PLM Software 4GD combines the best of both, and provides a single all-encompassing PLM shipbuilding environment with rich, full lifecycle data management capabilities that is well-suited for managing massive products like ships.

Siemens Industry Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software

Siemens PLM Software, a business unit of the Siemens Industry Automation Division, is a world-leading provider of product lifecycle management (PLM) software and services with nine million licensed seats and 77,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software helps thousands of companies make great products by optimizing their lifecycle processes, from planning and development through manufacturing and support. Our HD-PLM vision is to give everyone involved in making a product the information they need, when they need it, to make the smartest decision. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

Y13 37145 9/13 C