CAPTURING AND EXPLOITING KNOWLEDGE IN A COMPONENT-BASED SHIPBUILDING PRODUCT MODEL

Michael A. Polini, Newport News Shipbuilding, Newport News, Virginia
Patrice Blanchard, Intergraph Corporation, Paris, France

Abstract

During the past five years, significant research has been undertaken by the shipbuilding industry to apply object-oriented technology to product modeling information systems so as to capture and associate behavior with shipbuilding objects. The primary driver behind these efforts has been the need to become globally competitive in the face of a shrinking industrial base. Challenges facing the industry that must be addressed by the next-generation information systems include an aging workforce, the need to standardize not only the design but the information derived from product models, and the realization that automating the ship design process contributes to compressed acquisition schedules and lower overall costs. Advances in Information Technologies are rapidly changing the application development landscape and the current trend is towards user-friendly applications assembled from modular software components that adhere to documented protocols. In addition to the promise of a high level of re-usability, these components are appealing due to their ability to function in distributed, network environments—a necessity for scalable systems and virtual collaborative design environments. This paper will discuss the development of an integrated suite of shipbuilding applications developed using Microsoft’s Component Object Model (COM) running under Windows NT on high-end PC’s. It will address how knowledge is captured in the system, how the system can be customized to meet varied enterprise needs, how this knowledge must be managed in a collaborative, multi-discipline environment, and how this knowledge can be exploited to automate the ship design process.

Introduction

At one extreme, computer scientists think of knowledge as something commonly associated with Artificial Intelligence (AI) technology and expert systems and their research concentrates on expanding the ways computers can use problem-solving knowledge to perform tasks that seem to require intelligence. From a different perspective, a mother would say that knowledge is not going out in the cold without a coat on. The Random House College Dictionary identifies information, knowledge, and wisdom as “terms for human acquirements through reading, study and practical experience”. It goes further to define information as “that which applies to facts told, read, or communicated that may be unorganized and even unrelated.” Knowledge “is an organized body of information, or the comprehension and understanding consequent on having acquired and organized a body of facts”. Pragmatically, from the shipbuilder’s viewpoint, knowledge represents the human elements that transform the abstraction of a ship concept into the multitude of parts that, when assembled, produce a vessel that meets its functional, financial, and schedule objectives. In a simplistic approach, shipbuilding knowledge can be represented by the “rules and standards” employed by an organization through the course of a design.

Newport News Shipbuilding (NNS) has been devoting a significant number of resources over the past several years to studying and optimizing the activities associated with the shipbuilding process across the entire enterprise. These business process re-engineering efforts have led to numerous and
far-reaching changes to the shipyard—the most notable being the establishment of the “Shared Data Environment (SDE)” project. The objective of this effort is to link the individuals throughout the shipyard through an information network and provide integrated business and engineering applications that eliminate redundancy, centralize information, and minimize “paper-based” data exchanges. One of the key components of SDE is a next-generation shipbuilding product modeling system. Using state-of-the-art technology, this system is being designed to support the optimized shipbuilding processes being developed (i.e. the “best practices”) and to provide a flexible mechanism for allowing the shipbuilder to define and embed knowledge—in the form of rules and standards—into the system. This paper will address technology that is the enabler for this knowledge capture and is deemed to be fundamental to achieving the productivity gains necessary to attain world class competitiveness.

The issues that will be discussed are symptomatic of the entire shipbuilding industry, as evidenced by the recent study sponsored by the National Shipbuilding Research Project (NSRP) in support of its Maritech Advanced Shipbuilding Enterprise (ASE) Program. The findings are documented in a Strategic Investment Plan (SIP) for the industry [reference 1] that identifies six major initiative areas that must be addressed to enhance the global competitiveness of the industry. The topic of this paper and the underlying research is germane to the “Advanced Design, Simulation, Estimating, and Analysis” and the “Component-based Software” sub-initiatives within the Systems Technologies major initiative.

**Background**

Newport News Shipbuilding has been developing proprietary shipbuilding design systems for over a decade. Beginning in 1996, it built on the research efforts NNS had performed in support of the DARPA Simulation-based Design Program with respect to an object-oriented product modeling system and developed a design tool that was commonly referred to as the “smart product model”. This UNIX-based system, known as workstation VIVID®, was put into production use for structural design at the shipyard in the fall of that year. It was a very advanced system that had a rich set of behaviors associated with a broad spectrum of shipbuilding objects and it prototyped many of the re-engineered processes within the engineering community as well as many of the leading-edge technologies in the computer science field. Significant progress was made in the areas of design standardization and automation through the use of rules.

In 1997, the team of Newport News Shipbuilding and Intergraph Corporation began working together to develop and deploy systems technology under the DARPA Maritech Program based on Microsoft’s Windows NT operating system, their Component Object Model (COM, today called COM+), and their Distributed interNet Architecture (DNA) technology. This project was aptly named the Common Object Model of Products/Processes for an Advanced Shipbuilding System (COMPASS). It was believed that ground breaking shipbuilding systems—based on modern information systems technology—were required to achieve drastic productivity gains. In addition, empowerment at the worker and manager levels would be enabled when these new systems were financially attractive, intuitive, familiar, easy to use, and available at their desktops. Many vendors of commercial shipbuilding systems are faced with these same challenges and there is a variety of avenues that can be pursued to achieving these goals. This strategic partnership conscious choice to approach a next-generation shipbuilding design system by

- provide a core set of shipbuilding components to the industry
- supply a suite of end-user applications in the form of customized User Environments tailored to focused tasks in the shipbuilding process
- define processes that reflect the “best practices” associated with these tools

**Figure 1 - COMPASS Benefits**
developing a new architecture that utilized the experiences gained from their legacy products. The ramifications of such a decision led them away from having to worry about porting existing applications to new platforms, compromising the architecture to protect a large installed user-base, or considering putting a modern Graphical User Interface (GUI) veneer on top of legacy code. Instead, the team began building, from the ground up, a solid and sound system for shipbuilders that represented the first major technological innovation for shipbuilding information systems in the past decade. Figure 1 illustrates some of the objectives that the COMPASS program will fulfill.

**Component Technology Overview**

It is outside the scope of this paper to detail all the technical innovations and issues underlying the COMPASS program, however, an overview of the technology is warranted. This team is committed to component technology that is based on a 3-tier client-server architecture. These layers are commonly referred to as the "client" tier, the "business" tier, and the "data services" tier. This team follows a formal object-oriented development methodology that begins with Use Cases (the requirements as specified by the application domain experts) and progresses in an iterative manner through the software development activities of Analysis, Design, Code, and Test. For deployable products such as this, two additional activities are required, namely Training, and Implementation. These activities produce a variety of interim products (e.g. schedules, architecture and design documents, data models, test plans, implementation plans, and training materials) that culminate in the software components put in the hands of the shipbuilder.

Terminology in the software industry changes at a pace nearly as rapid as the underlying technology and those not immersed in the details—shipbuilders among them—tend to banter these terms about without understanding the subtleties associated with them. The case with which some vendors changed their systems from "CAD tools" to "product modeling tools"—with no underlying architectural changes—as the term came into vogue is a case in point. A similar phenomenon can be observed with the terminology surrounding software development methodologies and frameworks, specifically procedural development, object-oriented development and component-based development. What are the fine distinctions between these seemingly innocuous terms? Taking a definition from an article in the October 1998 issue of the "Rose Architect" published by Rational Software, a component is "a non-trivial, nearly independent, and replaceable part of a system that fulfills a clear function in the context of a well-defined architecture". Components are about interfaces—documented, immutable specifications to enable consistent and dependable communication channels. Object-oriented programming focuses on the development of object-based components. Component development is concerned with deploying these components in such a manner that they can interoperate with one another, can be assembled to form more complex, integrated systems, and that they can be widely distributed in client-server environments with no knowledge of who their clients are. As implied by the name, it is the software equivalent to the electronic industry's "black box" which, given some explicitly defined inputs, produces some defined outputs. What happens internally within the black box is of no concern and if one replaces this black box with another that has the same inputs and outputs, the result will be the same.

Component technologies can be traced back to a common origin—the Common Object Request Broker Architecture (CORBA) standard developed by a consortium of industry members known as the Object Management Group (OMG). Shortly after this standard was published, Microsoft created its own competing standard called the Component Object Model (COM+) and more recently, Sun Microsystems and IBM joined forces to develop their own competing standard called Enterprise JavaBeans (EJB). Each of these standards address the basic problems facing today’s application developers—namely language-independent object definitions and transparent distribution of objects within a single process, between multiple processes, and remotely over a network (reference 2 provides
an in-depth treatment of this technology from the COM perspective). It is the distribution capabilities of these technologies that has the biggest impact on the application architecture separating computer roles into "client", "business", and "data" categories (i.e. multi-tiered systems) and having enormous consequences for performance, scalability, reliability, security, and cost.

Intergraph was first in the engineering design tools industry to understand and appreciate the importance of Microsoft, Windows NT, COM+ and component-based interoperability standards, as well as the general effectiveness of the Windows development environment. Consequently, they moved quickly to create Windows-based graphics capabilities and implemented several user interface innovations specifically for graphics environments.

Intergraph further undertook definition of an extension of Object Linking and Embedding (OLE), called OLE for Design and Modeling (OLE for D&M) that permits engineering graphics tools to inter-operate with one another and with office automation (OA) tools such as word processors, spreadsheets, and presentation graphics applications. More recently, Intergraph has created a new, component-based Enterprise Engineering Environment (EEE) for application development and engineering data management. This EEE is designed to address the problems with today's CAD engine-based applications. It is the infrastructure, integrated tools, reusable software components, interoperability standards, and other capabilities needed to improve design efficiency, to facilitate early stage design optimization, and to integrate engineering systems with both the enterprise and operation and maintenance environments. Intergraph's EEE system will minimize overall information technology costs by leveraging Microsoft's Windows NT operating system, Windows user interface and application design standards, the COM+-based DNA architecture, incorporation of OLE-compliant OA applications, and COM+ compliant development tools and capabilities.

The EEE also includes user applications. Windows NT and the DNA infrastructure are foundational elements; EEE framework components layer upon the operating system and distributed computing infrastructure: application components layer on the framework; and numerous small

![Figure 2 - EEE Framework](image-url)
Applications use the services of the components and underlying infrastructure (Figure 2). Applications, known as User Environments (UE), developed on this architecture are small, task-focused, and easy-to-use. Intergraph, along with its shipbuilding development partners, envisions a broad slate of these task-focused and easy-to-use ship design and production applications.

In summary, the importance of the preceding section is to highlight the breadth of new technology that can be brought to bear on the problems facing the shipbuilding industry.

**Elements of Knowledge Capture**

The ultimate question from the shipbuilder is “what does all this mean to me in terms of my day-to-day business of building ships?” To answer this question, the Molded Form User Environment (UE) will be studied in detail. This UE has been specifically tailored to enable the creation and modification of objects that represent the zero-thickness curves and surfaces that give the ship its basic form and will be used by shipbuilders in the early stages of design. These objects live on the middle-tier of the architecture with numerous objects from other UE’s. Collectively, these are commonly referred to as “business objects”. While there are many objects within the system at this level the majority are necessary to support the application developer and only a few are meaningful and exposed to the shipbuilding end-user. In order to effectively create and interact with the business objects, software must be written at the client-tier level. This software takes the form of the commands and user-interface components necessary to effect the task-focused UE that reflects the domain of interest. Microsoft’s Visual Basic (VB) language is commonly used to develop client-tier applications and components. In addition, VB can be used to define “rules” that can be invoked by business objects to modify and/or create other business objects. Each of these three areas—User Environments, Business Objects, and Rules—will be discussed in more detail in the following paragraphs.

![Diagram of User Environments](image)

*Figure 3 - User Environments*
**User Environments**

The 3-tier architecture represents a layered system approach whereby some services build on other services. Those at the lower level are typically referred to as core services. There is another partitioning that can take place that subsets the application domain space rather than the technology domain space. This sub-setting corresponds to the User Environments described above. At the most elementary level, these UE's can be thought of as the most rudimentary form of knowledge since they organize the vast amount of data (i.e. information) about the ship into small, well-focused, partitioned subsets. Figure 3 illustrates this concept.

**Business Objects**

Business Objects are another major element necessary to enable knowledge capture within the system. There is a close coupling between the business objects in the middle-tier and the software written for the client-tier. Trying to deal with enforcing rules at the client-tier means that each client must understand the intricacies of the architecture and implement the rules on a case by case basis. Obviously, this approach is unlikely to succeed because it will be difficult if not impossible to educate all potential clients on the architecture and impossible to validate whether all client applications correctly implemented the rules. Even if it were possible to overcome these obstacles, the resulting system would be prohibitively expensive to maintain as changes to the rules would have to be accompanied by changes to each of the clients. Concerning the underlying knowledge, there is a significant advantage in embedding this information into the business objects themselves as opposed to handling it at the client-tier level. At the business object level, this knowledge need only be captured once and it is guaranteed to result in consistent behavior for each and every client.

![Diagram of UML Data Model](image)

**Figure 4 - UML Data Model**

280
Data models of the components are developed as part of the software engineering process. These data models define the attributes (i.e. properties) and relationships between the various objects. The Unified Model Language (UML) is used to formally capture this information. Figure 4 represents a sample UML data model. Business objects can be categorized by their expected behavior within the system and can either be persistent—that is saved from session to session—or non-persistent living only for a brief period of time within a single session.

Passive entities are the persistent business objects that represent the physical things making up the ship. These entities store the static attributes appropriate to the object. A “plate system” will be used as an example of a passive entity in our scenario, however, profile systems, seams, openings, connections, and structural specifications are other business objects within the Molded Form UE. The business objects in this category are those typically presented to the application user.

![Figure 5 - Split Operation](image)

Active entities are the persistent business objects that represent the operations that manipulate and transform the passive entities. These entities are responsible for identifying the inputs and outputs of the operation as reflected by relationships with other passive entities. Again, within the Molded Form UEs.
Form UE, create, split, cut, and bound are operations represented as active entities. These entities are usually presented indirectly to the application end-user in the form of commands associated with a toolbar or a menu choice and usually only directly addressed by application developers. Figure 5 is an example of the split operation.

Semantics are the non-persistent objects that perform the update to the passive entities based on the relationships defined by the active entities they are associated with. These semantics are triggered by the relationship support component when any of the inputs to the active entity are modified. These semantics make use of utilities that perform much of the actual modifications to the passive entity. These objects are never exposed to the end-user and are created and maintained by the application developer.

![Visual Basic Example](image)

Figure 6 - Visual Basic Example

Through careful analysis of end-user requirements, appropriate business objects can be designed and implemented that result in components suitable for the task at hand. These components can be presented to the user through a narrowly focused, highly customized user interface that has been optimized for a particular work flow within some aspect of the ship design process. These business objects capture and record—either through attribution or via explicit relationships—information as the design process progresses. This information embodies knowledge transferred from the mind of the designer into the product model database in a manner that is suitable for exploitation by other aspects of system.

**Rules**

One means of supporting rule definition that is both flexible and extensible is through Visual Basic (VB) programs. Some number of these programs could be supplied by the system developers and serve as templates for users wishing to customize their system. Because the data models are captured within the repository, the business objects are capable of automatically exposing their properties and interfaces from within the development environment. Figure 6 is a screen capture of such a situation.

Given that there exists a means to define a rule, the challenge is then to provide a mechanism by which the rules can be invoked. The authors propose the concept of a "structural specification"
business object that can serve as the “dispatcher” for the default rule behavior. This structural specification can be associated with the plate and profile system business objects and can be divided into categories that correspond with the common operations acting on the objects. Within these categories may be any number of situations where a specific rule would be needed. As an operation is executed, the specification would be queried for the known configuration and the rule invoked to supply the default behavior. Several categories and common situations are outlined below:

Naming Rules
- Plate systems
- Profile systems
- Plate parts
- Profile parts
- Seams
- Openings
- Connections
- Features

Connections Rules
- Continuity
  - Profile end to profile end
    - aligned
    - knuckled landing curve
    - orthogonal landing curve
- Profile end to profile edge
  - aligned
  - orthogonal
- Profile edge to profile edge
- Profile edge to profile face
- Profile end to plate face
- Profile edge to plate face
- Profile edge to plate edge
- Plate edge to plate edge
- Plate edge to plate face

Penetration Rules
- Tightness
- Profile penetrating plate
- Profile penetrating profile
- Seam penetrating plate
- Seam penetrating profile
- Opening penetrating plate
  - Normal Access (e.g. doors)
  - Limited Access (e.g. manhole)
  - Lightening
- Pipe penetrating plate
- Pipe penetrating profile
- Duct penetrating plate
- Duct penetrating profile
- Electrical penetrating plate
- Electrical penetrating profile
- Opening penetrating profile

Knowledge Exploitation

Application data models are publicly exposed within the Microsoft Repository and they may be created and/or modified with graphical editing tools such as Rational Rose or Microsoft Visual Modeler. These tools present data models as Unified Modeling Language (UML) diagrams. This eliminates the use of obtuse ASCII file-based “meta-data” definitions typical of many current generation applications. Application data models may be extended with new properties, which will display in property dialog boxes. Or, they may be extended with new objects and relationships that may be implemented as COM objects identified within the meta-data. These can be manipulated with either modified application components or new commands in client applications. Likewise, new relationship semantics can be added to implement specialized rule-based behaviors triggered by the framework. Thus, the Repository supports a completely unprecedented level of open applications.
In addition to the application components described above, a host of framework-supplied reusable software components are available or under development and include:

- ActiveX controls for generic application commands (e.g., view manipulation commands, printing and plotting, etc.)
- A Relation Support component for change notification and propagation
- Query and Versioning components
- A persistent data manager component to provide database isolation
- A variety of miscellaneous framework services.

As mentioned earlier, the "relation support" component is the key to enabling the exploitation of the knowledge embedded into the system in the form of all three categories of business objects.

In order to support rule-based design, a "relation support" component has been provided and is responsible for reading application meta-data (i.e., the information model or schema) out of the Repository and using it to intelligently propagate the impact of user actions and software-induced changes to the data. The meta-data contains knowledge of application logic associated with relationships between application entities. These so-called "relationship semantics" are intelligently triggered for execution by the Relation Support component, thereby providing the generic mechanism required to solve the problems of application editing, reuse of historical data, and change management described earlier.

![Figure 7 - Relationship Graph](image)

Figure 7 is an example of what the internal state of the product model may look like after a series of user actions. A passive plate system object was created and at some point, a "create" operation resulted in some geometry being associated with the plate system. At some later point, a "bound" operation was invoked and the geometry from other plate systems was recorded as input. The semantics associated with this active entity caused the geometry of the original plate system to be trimmed by the geometry of the bounding plate systems. In a similar manner, downstream design activities can result in cut and split operations that recorded the relationships with the various business objects. What is important to point out is that it is possible to control the degree of propagation of change with such a mechanism. For instance, should an attribute on one of the bounding plate systems
be modified, the relationship support component would not trigger the semantic associated with the bound active entity because the plate system was not an input. Only when a change to the geometry of the bounding plate system occurs will the update take place.

**Business Case**

To round out the discussions related to embedding knowledge into product modeling systems, a high-level overview of the business case behind this effort will be presented. As stated earlier, NNS has been studying and refining the activities performed by the engineers and planners over the course of the ship design lifecycle. These studies have identified three major issues related to product modeling that have a significant impact on productivity. The first is the need to begin product modeling in the earlier stages of the lifecycle, the second is the need to standardize the design products, and the third is to defer the actual creation of the parts for as long as possible. Using the illustration in Figure 8, each of these will be discussed in more detail below.

![Figure 8 - Product Lifecycle](image)

**Product Models in Conceptual Design**

Most traditional CAD modeling systems use geometry as the underlying representation of the product and then associate attributes with it to enhance its definition. While this approach can be used to build ships, it comes with one major drawback—one cannot begin to build the CAD model until the geometry is known. In the early stages of design, much of the information is not geometric in nature. One may know very early that there will be five decks on the vessel, but they may not know where they are located, whether they will be planar or not, or what their exact shape will be. For this reason, shipbuilders who use these traditional CAD modeling tools, tend to think of them as detail and production design tools and use other means to develop the concept and contract design products. Should the shipyard be successful in winning the work, the CAD modeling effort begins anew with the detail design activities. Since the concept and contract products are the means by which the price and
schedule are negotiated and established, any discrepancies between the model developed in these early stages and the CAD model developed during the downstream detail and production design stages can lead to cost and schedule overruns.

Using an object-oriented product model, geometry is treated as an optional attribute of the object. With this approach, the shipbuilder works at a higher level of abstraction with objects that closely represent the physical aspects of the ship. The objects can be created early in the lifecycle, with or without geometry as appropriate. Relationships can be established between the objects at this stage that are independent of the underlying geometry. As the design progresses, other non-geometric attributes can be defined and the product model undergoes a gradual refinement throughout the lifecycle until it fully matures into the finished product.

Standard Design Products

The term “standard design products” is used to refer to the re-usability aspect within or across ship design projects. Several studies have cited the fact that U.S. shipbuilders re-engineer a much higher percentage of the ship than our foreign counterparts. In many ways, these efforts could be simply viewed as wasting engineering time and money, however, in the broader sense, they can lead to serious downstream problems since these newly engineered products have never been validated in a production setting. NNS has found that one of the biggest factors affecting the quality of the design and productivity of the end-user was embedding rules into the product modeling system. One of the most critical activities under a “design for production” approach is that of lofting—the art or science of adjusting the part geometry to account for fit-up clearances, weld shrinkage, and/or added material. This role was traditionally reserved for the most experienced and skilled designers and engineers who took the detailed parts when they were completed and performed their magic on them before sending them off to be manufactured. In many cases, the adjustments were made in the 1/16\textsuperscript{th} of an inch range and were not obvious to the naked eye. More importantly, there was no consistency between one loftsmen and another and there was nothing but the individual’s personal discipline that ensured that all parts had been lofted correctly. This resulted in processes that included activities for checking and validating the parts on the engineering side, activities for checking the parts at the manufacturing site, and accuracy control activities for checking and validating the parts after they were produced. Indirectly, the processes for procuring material had “safety valves” built into them in the form of excess material to re-do parts found to be in error and the assembly schedules were padded to allow for the extra time necessary to correct these errors.

Once the rules were embedded into the product modeling system and validated by the downstream users, many of these extraneous activities became unnecessary. The need for constant checking at each stage of the design process could be eliminated, and the focus of accuracy control shifted from a piece part mentality to an overall process quality mode. The products coming out of the engineering organization were standardized and consistent with the expectations of the production organization, were feasible for manufacturing within the confines of the company’s facilities, and represented the best practice, lowest cost alternative. Where there was once a critical skill shortage in the engineering process concerning lofting, the product modeling tool turned every user into a top-notch loftsmen. The black magic that was once associated with these activities became a well understood, documented set of rules that became part of the corporate knowledge-base and could be monitored and maintained at a strategic level.

Deferred Part Creation

There are two other observations that are important about these early design stage activities based on the figure. The frequency of change is very high early in the lifecycle, however, the number of “parts” is very low. The term parts is used liberally in this context and an argument can be made
that there is a distinction between the business objects that represent the early stages of the ship product model and the business objects that represent the later stages. The authors have chosen to address this distinction by referring to the early stage objects as "systems" and the downstream objects as "parts". Systems can be equated with the logical or functional definition of the ship, while parts correspond to the physical definition. Many idealizations are possible at the system level which simplify greatly the internal data definitions necessary to create an early lifecycle model of the vessel. Experience with such concepts on a product model of a 40,000 DWT commercial tanker resulted in several thousand structural plate and profile systems and about 50,000 plate and profile parts—an order of magnitude difference! The number of geometry objects (i.e. curves and surfaces) representing these parts numbered about 350,000, again another order of magnitude difference. While it may be possible to effectively keep up with changes in the early stages of design, it becomes increasingly more expensive and time consuming to do so in the later stages of the lifecycle. Even though the frequency of change has slowed significantly, the sheer number of parts that may need to be updated can be overwhelming and configuration management becomes the governing factor. Without the ability to capture the knowledge that went into the design beginning in the early stages, it becomes a purely manual, labor-intensive exercise to incorporate change after the bulk of the parts have been created.

Conclusion

The authors have had considerable experience with object-oriented product modeling systems and the concepts described in this paper. While commercial products built on this technology are not anticipated to be available in the marketplace until the year 2000, advanced prototypes and internal pilot implementations have shown great promise in reducing design time, improving the final product quality, and lowering the required skill level of the user. By capturing knowledge through the use of active entities and semantics at the business object level, and by supporting a mechanism for automation using rules, the actual creation of parts can be deferred until very late in the design cycle. The domain experts can shift from a piece-part mentality to a systems engineering mode where they 1) concentrate their efforts on creating the system business objects; 2) establish the appropriate relationships between objects; and 3) associate rules with the business objects to drive automation. The tedious part creation activities can be automatically performed by the system following a validated set of rules, thereby guaranteeing product quality and significantly compressed design periods.

References

AN INTEGRATED DESIGN AND PRODUCTION ENVIRONMENT FOR SHIP MACHINERY SYSTEMS

Robert Bronsart and Steffen Gau, University of Rostock, Institute of Maritime Systems and Fluid Engineering, Rostock, Germany

Abstract

The paper describes the ship machinery systems design and production process chain. Different partner involved are identified and their specific function is explained. Communication processes forming the base for the concurrent and distributed production environment are thoroughly analysed with respect to who exchanges what kind of information and their related interdependencies. Existing standards, especially those under development for the exchange of product data relevant to the worked out communication scenario e.g. STEP AP 217, 226 and 227 are discussed. It is shown that the scope of these APs as defined today overlap considerably. The process integration strategy followed in an ongoing R&D project of a German shipyard and a consortium consisting of engineering subcontractor, classification society, communication technology provider, module manufacturer is described.

Introduction

Until some years ago, almost the complete design and production of ship machinery systems was carried out at the shipyard building the vessel. Today the situation has changed as shipbuilders world wide try to stay competitive. In this context, the same strategic approach as in other industries, like the car industry, is followed by many shipyards: more and more activities of different process steps are performed by subcontractors. Whereas some traditional relationships exist like ship model basin doing the power prediction, the new strategy results in a design and production environment, in which engineering subcontractors are assigned overall responsibility for complete onboard systems e.g. heavy fuel oil, lubrication oil or freshwater system. Based on the functional specification laid down in piping and instrumentation diagrams, the makers list and additional necessary information on the general arrangement as well as the steel structure layout supplied by the shipyard, engineering subcontractors perform the detailed design like routing of pipes, ducts and cables and by this generate fabrication information to be used in the downstream process steps. Beyond this design task, engineering subcontractors also have to take care for the approval by classification societies, the manufacture of components and modules by subcontractors as well as the installation of the systems onboard the vessel at the shipyard.

This "virtual enterprise environment" for design and production of one-of-a-kind products which consist of many and complex systems and the very short time to market has resulted into problems not known before. The necessary transactions of today manly paper based information between the partners involved has partially led in a slow down of the processes and the necessity to control different product descriptions in different versions at different sites. To overcome these problems, the definition of a commonly used product model and the usage of state of the art information technology are regarded as key factors. The following principal benefits are identified by the maritime industry:

- reduction of design and production time which in turn results in the reduction of costs,
- elimination of errors due to inconsistency problems caused by multiple product definitions used by the partners involved,
- support of version control for design variants.
Many research and development resources were and are still allocated for the definition of product models suitable to meet the formulated communication requirements. Kendall and Hasund [1] give a short overview of the STEP application protocols with an application domain related to shipbuilding. Langbecker and Rabien [2] describe the activities under the umbrella of the European Maritime STEP Association (EMSA). The defined business cases worked on in the R&D projects mainly focus on the exchange of hull form and ship structural data.

In this paper, the design and production process chain of ship machinery systems is analysed to identify those business cases with a great potential for an increase of the overall productivity. The technology used will principally be the same as for the above mentioned business cases but based on different product model scope.

Ship Machinery Systems – Design and Production Process

The design and production process of ship machinery systems today is performed in a concurrent and distributed working environment. This overall situation is better described when focusing on a configuration as seen in practice. A consortium consisting of shipyard, engineering subcontractor, classification society and module manufacturer serves as an example. These four partners are involved in a co-operation scenario working on the fuel oil supply system and major components thereof. The entire process chain comprises pre-design, schema approval, detailed design including generation of manufacturing information, material and component ordering and logistics, parts manufacturing and assembly and final approval as well as the installation of complete modules onboard the vessel. Within this situation each partner serves clearly described functions and has to take different responsibilities.

A module in this context is an assembly of components which in general belong to one piping system. The prefabricated unit is installed onboard and connected via pre-defined interfaces. For an example of a module, in this case part of the fuel oil system, please see Figure 1.

In the following, the today’s situation is described to some detail, please refer also to Figure 2.

Shipyard

In accordance with the owner requirements and rules and regulations to be observed the shipyard creates early stage product information like general arrangement, engine room layout and principle piping diagrams of the most relevant systems. These and some other documents form the information set which is called ‘classification project’ and are sent to the classification society for approval.

The approved documents are sent back to the shipyard where they are stored, copied and transferred together with additional information to the engineering subcontractor. All of these documents may be called the ‘technical project’.

Figure 1. Example of a HFO module (courtesy of MTE)
**Classification society**

At the beginning the classification society receives a classification inquiry issued by the shipyard. This results in prescribing a unique registration number to the ship to be built.

The classification project supplied by the shipyard is directed to the departments responsible. An analysis and approval follows. The classification society is utilising its specific regulations as well as internationally valid laws and restrictions. When the classification project documents do not comply with the rules the classification society is requesting changes and/or improvements from the shipyard. This results in an extended communication between classification society and shipyard. In case all documents comply with the rules the classification society will document it’s approval by adding the approval stamp and signature to the relevant documents. This stage marks the milestone for further detail work.

![Diagram](image)

**Figure 2.** Design and Production Process for Ship Machinery Systems

**Engineering subcontractor**

Provided with the approved documents and other shipyard specific information as well as owner specific requirements like the makes lists, the engineering subcontractor is carrying out the detailed design of the systems. These activities are done taking classification and as necessary further regulations into account and using additional information from the shipyard regarding other specific systems. A 3-D model of all components and pipes is generated. In co-operation with the shipyard module parameter and interim results are exchanged. Another major task is the generation of fabrication information. This includes pipe sketches, piece lists and even control data for (C)NC machines.
used by the module manufacturer. Generated data are processed and delivered to the shipyard and module manufacturer and kept in the own project archive.

**Module manufacturer**

The fabrication of piece parts and assembly of complete modules starts on the basis of the results of the upstream process steps. Out of numerous information sources all necessary documents are to be selected, materials and components in accordance to the makers lists and required specifications have to be ordered. In parallel the existing resources are checked and capacities are assigned to specific fabrication steps in the assembly tree (production planning).

The task do deliver a fully prefabricated ready-to-operate module asks for direct contact between manufacturer and classification society. Several built-in components as well as the completed module need to be approved according to classification society rules and regulations. These approvals have to be carried out by classification surveyors at the manufacturers site, often before the module is completed (without insulation, painting,...).

The delivery of modules and components as well as the complete product documentation including all classification certificates has to be agreed upon between module manufacturer and shipyard. All hardware is installed onboard the vessel at the shipyard, the related documents like approval certificates, operation manuals etc. are handed over to the shipyard to either be kept in the files or put onboard.

When looking at the chain of activities it can been seen that different types of co-operation have to be distinguished. Each is representing a specific configuration, see Table 1.

### Table 1. Partners and Communication Links

<table>
<thead>
<tr>
<th>Communication Link</th>
<th>Partners Participating</th>
<th>Business Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC CS 1</td>
<td>Shipyard ↔ Classification society</td>
<td>I + IV</td>
</tr>
<tr>
<td>SE ES 2</td>
<td>Shipyard ↔ Engineering subcontractor</td>
<td>II</td>
</tr>
<tr>
<td>EM ME 3</td>
<td>Engineering subcontractor ↔ Module manufacturer</td>
<td>III</td>
</tr>
<tr>
<td>MC CM 4</td>
<td>Module manufacturer ↔ Classification society</td>
<td>IV</td>
</tr>
<tr>
<td>MS SM 5</td>
<td>Module manufacturer ↔ Shipyard</td>
<td>-</td>
</tr>
<tr>
<td>EC CE 6</td>
<td>Engineering subcontractor ↔ Classification society</td>
<td>I</td>
</tr>
</tbody>
</table>

According to Table 1, 12 communication links exist, looking at each communication from both partners involved. In order to generalise 6 different communication scenarios are identified.
Comparing these scenarios with the process chain depicted in Figure 2, it can be seen that link no. 6 does not occur in the actual co-operation activities. Communication links are determined by the participating partners and their communication requirements. These information exchange scenarios may be regarded as business bases which are characterised by the parameter described below.

![Diagram showing co-operation network and business cases]

**Figure 3. Co-operation Network and Business Cases**

**Communication Requirements**

Business cases as defined above and depicted in Figure 3 represent inter-company communication processes. In the following each of the identified business cases is described by

- the principle partner involved,
- the activities to be supported,
- the information representation in the data exchange as used today,
- the amount of data (qualitative statement),
- the "direction" of information exchange, if applicable,
- the communication technology used today for the information exchange, if applicable.

**BC I Design and approval of ship machinery systems**

Example link: Shipyard ↔ Classification society

<table>
<thead>
<tr>
<th>Task</th>
<th>Information representation</th>
<th>Data volume</th>
<th>Data flow</th>
<th>CT used today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional design of ship machinery systems</td>
<td>CAD database, P&amp;ID</td>
<td>very high</td>
<td>S internal</td>
<td>-</td>
</tr>
<tr>
<td>Exchange of classification relevant documents</td>
<td>text, drawings</td>
<td>high</td>
<td>S → C</td>
<td>mail</td>
</tr>
<tr>
<td>Approval of systems</td>
<td>text, drawings</td>
<td>high</td>
<td>C internal</td>
<td>-</td>
</tr>
<tr>
<td>Change requests Communication</td>
<td>oral discussion, text, drawings</td>
<td>low - high</td>
<td>C ↔ S</td>
<td>mail, fax, phone, email</td>
</tr>
<tr>
<td>Transfer of certificate (sign and stamp)</td>
<td>text, drawings</td>
<td>low</td>
<td>C → S</td>
<td>mail</td>
</tr>
</tbody>
</table>
### BC II  Ship machinery module detailed design

Example link: Shipyard ↔ Engineering subcontractor

<table>
<thead>
<tr>
<th>Task</th>
<th>Information representation</th>
<th>Data volume</th>
<th>Data flow</th>
<th>CT used today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer of specification and approved documents</td>
<td>text, drawings</td>
<td>high</td>
<td>S → E</td>
<td>mail, fax, email</td>
</tr>
<tr>
<td>Generation of detailed module design data</td>
<td>CAD data, text, drawings</td>
<td>very high</td>
<td>E internal</td>
<td></td>
</tr>
<tr>
<td>Communication on interfaces to other systems</td>
<td>oral discussion, text, drawings</td>
<td>low - medium</td>
<td>E ↔ S</td>
<td>phone, fax, mail, email</td>
</tr>
<tr>
<td><strong>Delivery of design information</strong></td>
<td>text, drawings</td>
<td>very high</td>
<td>E → S</td>
<td>mail, email</td>
</tr>
</tbody>
</table>

### BC III  Ship machinery production engineering

Example link: Engineering subcontractor ↔ Module manufacturer

<table>
<thead>
<tr>
<th>Task</th>
<th>Information representation</th>
<th>Data volume</th>
<th>Data flow</th>
<th>CT used today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation of fabrication and assembly data</td>
<td>text, drawings, NC-files</td>
<td>very high</td>
<td>E internal</td>
<td></td>
</tr>
<tr>
<td><strong>Exchange of design and manufacturing data</strong></td>
<td>text, drawings, NC-files</td>
<td>high</td>
<td>E → M</td>
<td>mail, email</td>
</tr>
<tr>
<td>Communication on manufacturing details</td>
<td>oral discussion, text, drawings</td>
<td>low - medium</td>
<td>M ↔ E</td>
<td>phone, fax, mail, email</td>
</tr>
<tr>
<td>Transfer of as-built data</td>
<td>text, drawings</td>
<td>low</td>
<td>M → E</td>
<td>mail, fax, email</td>
</tr>
</tbody>
</table>

### BC IV  Survey and approval of ship machinery systems

Example link: Module manufacturer ↔ Classification Society

<table>
<thead>
<tr>
<th>Task</th>
<th>Information representation</th>
<th>Data volume</th>
<th>Data flow</th>
<th>CT used today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request for survey and approval</td>
<td>text, drawings</td>
<td>low</td>
<td>M → C</td>
<td>mail, fax</td>
</tr>
<tr>
<td>Communication on survey details</td>
<td>oral, text</td>
<td>very low</td>
<td>C ↔ M</td>
<td>phone, fax</td>
</tr>
<tr>
<td>On-site survey of components / module</td>
<td>text, drawings, survey form</td>
<td>very low</td>
<td>C</td>
<td>on-site discussion</td>
</tr>
<tr>
<td>Transfer of classification Certificates</td>
<td>certificate</td>
<td>very low</td>
<td>C → M</td>
<td>mail</td>
</tr>
</tbody>
</table>

294
The primary information elements of the four business cases are listed in Table 2. Apart from the exchange of high volume data, an informal communication exists in many cases. Especially in those situations when additional ad hoc information is needed by any of the partners involved, the personal communication is regarded as the most efficient one. This is important to keep in mind when trying to formalise the communication in a "virtual enterprise" which is of highly dynamic nature.

Table 2. Primary Information Elements

<table>
<thead>
<tr>
<th>Business case</th>
<th>Information elements in scope of data exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design and approval of ship machinery systems</td>
<td>- Ship machinery specification</td>
</tr>
<tr>
<td></td>
<td>- Makers list</td>
</tr>
<tr>
<td></td>
<td>- Functional system design (P&amp;ID)</td>
</tr>
<tr>
<td></td>
<td>- Connection of components</td>
</tr>
<tr>
<td></td>
<td>- Instrumentation and control</td>
</tr>
<tr>
<td></td>
<td>- Material, stream</td>
</tr>
<tr>
<td></td>
<td>- Catalogue</td>
</tr>
<tr>
<td></td>
<td>- Approval/change</td>
</tr>
<tr>
<td></td>
<td>- External references</td>
</tr>
<tr>
<td>Ship machinery module detailed design</td>
<td>- Ship machinery specification</td>
</tr>
<tr>
<td></td>
<td>- Makers list</td>
</tr>
<tr>
<td></td>
<td>- Functional system design</td>
</tr>
<tr>
<td></td>
<td>- Connection of components</td>
</tr>
<tr>
<td></td>
<td>- Piece parts, assembly</td>
</tr>
<tr>
<td></td>
<td>- Shape representation 3-D, location</td>
</tr>
<tr>
<td></td>
<td>- Catalogue</td>
</tr>
<tr>
<td></td>
<td>- Approval/change</td>
</tr>
<tr>
<td>Ship machinery production engineering</td>
<td>- Piece parts, assembly</td>
</tr>
<tr>
<td></td>
<td>- Shape representation 3-D</td>
</tr>
<tr>
<td></td>
<td>- Fabrication data (NC)</td>
</tr>
<tr>
<td></td>
<td>- BOM</td>
</tr>
<tr>
<td></td>
<td>- Catalogue</td>
</tr>
<tr>
<td>Survey and approval of ship machinery systems</td>
<td>- Ship machinery specification</td>
</tr>
<tr>
<td></td>
<td>- Functional system design</td>
</tr>
<tr>
<td></td>
<td>- Instrumentation and control</td>
</tr>
<tr>
<td></td>
<td>- Material</td>
</tr>
<tr>
<td></td>
<td>- Approval/change</td>
</tr>
<tr>
<td></td>
<td>- History</td>
</tr>
</tbody>
</table>

Business Cases versus STEP Application Protocols

Currently there are five application protocols under development which focus on the exchange of information in shipbuilding: Ship Arrangements (AP215), Ship Moulded Forms (AP216), Ship Piping (AP217), Ship Structures (AP218), and Ship Mechanical Systems (AP226). In the context of a ship machinery systems application domain, the application protocol "Plant Spatial Configuration" (AP227) which relates to "Functional data and their schematic representation for process plant"
(AP221) have to be looked at as well. In the following an overview of the scope of the APs is given.

**Ship Piping (AP217)**

The scope of the application protocol for ship piping is defined to cover the life-cycle phases: functional design, detailed design, production engineering, fabrication and assembly as well as testing, see Figure 4. According to the committee draft for comments [3], the following are within the scope:

- data required to support the definition of the operating flow states of a piping system for the purpose of analyzing flow conditions, computing required pipe sizes, and for documenting the operational conditions of piping systems for shipboard personnel;
- data required to support the definition of the geometry and rigidity of a piping system for the purpose of evaluating stresses in the system obtained by applying loads;
- data sufficient to describe the geometry and location of equipment connected by piping and other distributive systems;
- data that defines the geometry of piping components, the equipment to which they attach, and envelopes surrounding these objects sufficient to enable interference analysis;
- data that defines the sequence of bending operations needed to bend a fabricated pipe;
- data describing the assembly operations necessary to assemble a piping assembly;
- data that defines the test procedures that evaluate the proper operation of a piping system or subsystem;
- data necessary to support the extraction of a bill of material data for a piping system or piping assembly;
- data necessary to document the configuration status of one or more piping components;
- data that defines the maintenance requirements, history and status of a piping system or collection of piping components.

The conformance classes defined are grouped into the subsets:

- functional piping design,
- detailed piping design,
- production engineering design,
- piping test data,
- piping maintenance and repair data

with additional variations in each subset with respect to shape representation and configuration management information.

Compared to the other APs described below, this is a very broad scope definition.

**Ship Mechanical Systems (AP226)**

In the working draft of this application protocol [4], the scope is defined as follows. Lifecycle phases to support are specification, design/selection, approval, installation, commissioning, acceptance, operation, in-service inspection and maintenance, decommissioning and disposal.

Systems and components to support are air supply to the engine room, exhaust gas, fuel oil treatment, lubrication oil and engine cooling, propulsion drive line, main and auxiliary engines, thruster units, pumps, heat exchangers, air compressors, boilers, deck machinery, ... For each of these systems and components, the product definition information is in scope:

- functional and physical connectivity including connectivity to ship structure
- functional description such as performance and operational characteristics
- geometric representation
- technological information such as material, tolerance, ...
- data necessary to track lifecycle and operational history such as specification, inspection and maintenance data.

Declared out of scope are: piping arrangements, electrical distribution systems and control systems not integral to the machinery unit, ship's heating, ventilation and air conditioning (HVAC), cargo refrigeration, ..., data related to the manufacturing of systems and components.

Figure 4. Life Cycle Activity Coverage

**Plant Spatial Configuration (AP227)**

According to the introduction in the draft international standard (DIS) document, this application protocol [5] is to be used "for the exchange of the spatial configuration information of process plants". The information includes the shape and spatial arrangement characteristics of piping system components as well as the shape and spatial arrangement characteristics of other related plant systems (i.e. electrical, instrumentation and controls, and structural systems) that impact the design and layout of piping systems. In the design and fabrication of a piping system, the piping layout must be evaluated with respect to the spatial characteristics and arrangement of these related plant systems, and the requirements for clearances between systems. The complete specification of these other systems is not needed, but enough spatial information is needed to support the layout of the piping system.

This AP specifies additional requirements for the exchange of information required for the design and fabrication of a piping system. This includes information on the piping material, process stream fluid, and the piping system functional characteristics. A process and system design specifies process requirements for a piping system that includes pipe size, design temperatures and pressures, and insulation class. The physical design uses these process requirements for the design of the piping system.

The application protocol also identifies and provides a functional specification of the components of the plant piping system. The design information for a piping system may specify a pump capable of maintaining a pressure and flow rate. The design will also specify the shape limitations or requirements and the location of the pump in the system, but not sufficient information for the fabrica-

---

1 A process plant is defined as "an assembly of one or more plant systems and plant items that can, or is intended to perform, a chemical, physical or transport process. A process plant is identified as a single unit for the purpose of management and ownership. A process plant has both physical and functional aspects." (definition 3.3.37 in [5]).
tion of the pump. The principle focus of the AP is on piping systems and the shape and spatial arrangement of systems including plant items required to ensure the physical integrity of piping systems."

Furthermore, the functional view on the piping systems, traditionally documented in the piping and instrumentation diagrams (P&IDs) is supported. The representation of these diagrams (drawings) however are not supported by the AP. The shape of items making up the plant may be represented at various levels of abstraction, i.e. from an encompassing envelope to a detailed design description. The requirements of the following business cases are to be satisfied by this AP:
1. Exchange of requirements from a plant owner to an architectural engineering (AE) firm;
2. Exchange of process requirements for the plant piping system from a process engineer to a system design engineer;
3. Integration of designs created by different engineers;
4. Detection of physical interferences of plant piping system components with components of other plant systems;
5. Exchange of construction specifications between AE and construction firms.

The conformance classes are defined:
- Class 1 - Provides piping system functional information,
- Class 2 - Provides equipment and component spatial information,
- Class 3 - Provides plant layout and piping design information,
- Class 4 - Provides piping fabrication and installation information.

It has to be noted, that compared to the shipbuilding application protocols, this AP has already DIS status and though can be regarded fairly stable with respect to its contents.

![Figure 5. Scope Overlap Between AP 217 and AP 227, BC 1](image)

The application protocol "Functional data and their schematic representation for process plant" (AP221) focuses on the exchange of functional data and their schematic representation for process plants. It complements AP 227 in that P&IDs in form of a graphical representation, are in scope whereas the spatial configuration and shape as well as production information of pipe parts and
components are out of scope. Apart from this major difference, the large overlap in scope will result in a decision to be made on which AP implementations for an information exchange will be realised.

Comparison of Life-cycle Coverage and Scope

Figure 4 shows how the different life-cycle phases are supported by the APs. Whereas the usage of the application protocol for "Plant Spatial Configuration" is principally restricted to the plant engineering life-cycle phases, the AP for "Ship Piping" also supports information exchange in the production and testing phases. AP 226 however is planned to support the whole life-cycle of the vessel. According to the above outlined scope definition, the application protocol for "Ship Mechanical Systems" differs from the other two in that the level of detail for the product definition is far less. Another important difference is to be seen in the focus on the equipment rather than on the piping components. In Figure 5 the two APs with mainly focus on piping systems are depicted, indicating the concepts defined part of the corresponding scope. It can be seen, that there is a considerable overlap between AP 217 and AP 227. As one example, the concepts to be used in the first business case, "design and approval of ship machinery systems" are highlighted.

Conclusion

The analysis of the design and production process chain for ship machinery systems in a distributed environment results in four inter-company communication scenarios with different information exchange contents. The existing application protocols, namely AP 217 and AP 227, are able to fulfil (most of) the formulated requirements. The considerable overlap in scope of the two APs and the DIS status of the application protocol "Plant spatial configuration" might lead to the decision to make use of AP 227 for the exchange of data of ship machinery systems. A thorough analysis of this AP with respect to the requirements of the above outlined communication links will be performed. This approach is also stimulated by the experience gained in the implementation of AP 217 (ARM level). The work done by KCS in the MariSTEP project has shown that major modifications to the ship piping AP are necessary as the existing version does not allow for efficient data exchange of pipe components.

According to the results of the ongoing R&D project, it has to be kept in mind that for the realisation of an IT based communication infrastructure, the highly dynamic nature of the overall scenario has to be taken care for. The potential for an increase of productivity is dependent on the amount of information to be exchanged between the companies involved as well as on the frequency of the communication.

Acknowledgement

The work presented in this paper is supported by the Federal German Ministry of Education, Science and Research and Technology. The authors acknowledge the valuable discussions with the members of the project team from the shipyard Aker MTW, from the engineering subcontractor MTE, from Germanischer Lloyd, from the module manufacturer mas and from the information technology supplier Scheller.

References


   ➤ www.nist.gov/sc4/step/parts/part217

   ➤ www.nist.gov/sc4/step/parts/part226

   ➤ www.nist.gov/sc4/step/parts/part227

   ➤ www.nist.gov/sc4/step/parts/part221

The authors' email are: robert.bronsart@mbst.uni-rostock.de
steffen.gau@mbst.uni-rostock.de