Emerging Information Technologies
AN IMPLEMENTATION OF LARGE-SCALE PRODUCT MODEL VISUALIZATION IN SHIPBUILDING

Bryan Marz, Newport News Shipbuilding, Newport News, VA
Rob Lisle, Newport News Shipbuilding, Newport News, VA

Abstract

Newport News Shipbuilding (NNS) has investigated the latest emerging visualization technologies for interfacing with large product model datasets. From this investigative research, a solution has been reached for implementation at NNS to address the multi-operational needs of shipbuilding, including but not limited to operational analysis, preliminary and detailed design review, manufacturing, and training. The solution was designed for collaboration efforts, between local and off-site participants, and includes immersive capabilities. This paper will provide an overview of the visualization efforts undertaken at NNS and how visualization is being implemented within the shipbuilding process.

Disclaimer

Reference herein to any specific commercial products, process, or service by trade name, trademark manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Newport News Shipbuilding. Statements and opinions expressed in this paper are those of the co-authors and do not necessarily state or reflect those of Newport News Shipbuilding.

Introduction

In the area of computer science, designers and engineers have always been faced with the problem of inefficient or lack of computational capabilities to help in design and analysis. Rapid changes in computer hardware technology are now allowing software engineers to greatly improve the functionality and usability of applications on a daily basis. One area in particular that is beginning to show great strides in improvement is the graphical visualization of simulation and analysis projects. In the past, NNS has used VIVID® for performing large scale visualization. VIVID® is a proprietary product modeling system developed at NNS specifically to handle the requirements of solid modeling for ship design and manufacturing. During the design of the US Navy's new fast attack submarine, SEAWOLF, VIVID® was successfully used as a virtual prototype to replace the traditional full-scale physical mock-up. Because VIVID® was specifically designed to facilitate concurrent system development in a solid modeling environment, it enables the designer to load and visualize an entire ship's compartment. This capability is fundamental for an efficient product model centric design process. Many products that compete with VIVID are now using third party visualization products to bring together an entire view of the composite design. This is extremely inefficient for product
modeling on a day to day basis since the designer must continuously switch between applications. VIVID® offers a much better solution for ship compositing since the designer can always see the entire design. However, VIVID® is essentially a static environment. As the utilization of product modeling has grown, so also have the requirements to have the product model more representative of all aspects of the virtual ship. For this reason, NNS has undertaken an implementation of third party visualization applications to be used in the shipbuilding enterprise. The objective in this implementation effort is to augment the existing legacy systems with a dynamic environment. This dynamic environment will allow everyone in the enterprise to interact with or within the synthetic shipbuilding environment. This paper will discuss the process that was used to determine the appropriate technologies to achieve these objectives.

Approach

To begin the effort, a business model was developed to associated industry efforts, NNS user requirements and areas with the greatest potential return on investment. From this document, NNS developed a high-level business case for visualization implementation. During the industry research phase the following areas were studied:

- Business or user requirements
- Display Devices
- Software Options
- Hardware Options
- Infrastructure Requirements
- Security Issues
- Implementation Strategies

Business Requirements and Implementation Strategies

The top-level requirements for visualization can be easily stated as the technology that can enable products to be developed better, faster and cheaper. In order to understand how these technologies can be implemented one must first have a good understanding of your business processes and where visualization can enable these efforts. Essentially visualization technology is best deployed as a means of fundamental communication. Most people will agree that their primary sense of communication is visual. Consequently, NNS feels that the greatest return on investment (ROI) is in implementing visualization technology to facilitate your organizations processes for communicating product data information. The NNS business model is comprised of three primary tiers as shown in figure 1.
Figure 1. Visualization Tiers

The highest tier, which we call large-scale visualization, presents the biggest challenge. Large-scale visualization in this model is defined as the interaction of large volumes of ship product model data for large design and design review teams. Design review teams may range typically from around 12 to 60 people. Shipbuilding presents a scale problem like no other industry. A typical ship's machinery space may contain 10,000 parts equating to upwards of 10 million polygons. NNS has estimated that to visualize a complete NIMITZ class aircraft carrier the technology would have to support models of greater than 10 million parts and files on the order of 200 gigabytes in size. Navigation speed is also an issue when working with large teams. If the system's performance is slow and there is significant time spent waiting for the display, teams tend to lose their focus and side meetings begin to occur. Therefore we can derive several requirements for the large-scale tier; accommodate very large models and large design teams. These requirements then decompose into specific requirements for facilities, display devices, hardware performance and software functionality. The middle tier, which we call the engineering tier, represents the personnel who are doing the day to day design tasks working in a product-modeling environment. Model sizes will be smaller and collaborative efforts may usually involve only 2 to 12 people. Consequently there is a different set of requirements for display technologies, hardware performance and software functionality. In the lowest tier, which we call the enterprise tier, our goal is to provide some collaborative visualization technology to anyone operating in the business enterprise.

Display Devices
There are a number of possible display mechanisms for visualization solutions. The options run from personal computer monitors to fully immersive virtual reality headgear. The following list provides a range of potential options:

- Workstation Desktop Viewing
- Large Flat Screen Viewing
- Large Desk Projection Systems
- Large Wrapped Screen or Immersive Viewing
- Stereoscopic Viewing (desktop, flat or wrapped screen)
- CAVES
- Helmet Mounted Display Systems

Increasing Levels of Immersion
The most pressing business need for new visualization technology at NNS will be in the area of large volume visualization for electronic mockups with our customers. This particular area will most likely drive the performance requirements for both hardware and software. Very few people in industry are trying to visualize datasets as large as what is required for shipbuilding. The industry leaders in this area are the aerospace and auto industries. Our datasets will most likely be 1000X + the size of their typical datasets. NNS has considerable business experience in performing electronic mockup reviews. This approach was pioneered by NNS on the Seawolf program during the mid to late 1980's. During the Seawolf program, there were typically two levels of design reviews preformed: large audience space reviews and individual system reviews. Consequently, our primary driving visualization display requirement is to be capable of large audience viewing (12-60 people). Additionally, immersion technology provides a significant advantage in that it allows the viewers to be more enveloped in the visualization environment; and, thus, the effect or sense of spatial realism is increased.

For NNS, we have selected to implement large wrapped screen centers as well as flat screen centers depending on the need for immersion. All sites will be capable of stereoscopic viewing. We have chosen the Panoram System (http://www.panoramtech.com/) as the typical implementation. Figure 2 depicts a typical center.

![Panoram Display System](image)

Figure 2. Panoram Display System

For the engineering tier we are implementing smaller flat screen single projection systems. Personnel compositing in the product model will also have workstations (both NT and Unix) that will be able to display large volumes of geometry. The enterprise tier will utilize common network PCs for displaying web based visualization applications. There are also a host of other display technologies that may be implemented; however, the intent of this paper is to focus on our core shipbuilding requirements.
through the ship interactively. This is needed in case anyone within the review wants to take a closer look at a compartment or piece of equipment. The benchmark that we are striving for is a minimum continuous refresh rate of 10 Hz.

**Collaborative Review**

A collaborative review is the ability of multiple users in multiple locations to review and manipulate a common model. This is becoming increasingly important with travel budgets dwindling and the desire to keep design team members informed with regularly scheduled meetings. Ideally, these reviews should require a minimal amount of preparation and special software, and should perform adequately on a mid-range engineering workstation. The objective is to be able to perform interactive distributed design reviews within a central model in conjunction with video teleconferencing technology. Requirements address areas such as cost, platform, ease of use for casual users, avatar technology, telecommunications technology, ability to coordinate effort, ability to create and collect comments as well as many others.

**Equipment Removal**

An important consideration of ship design is ensuring equipment can be removed with minimal impact to other ship components. Creating, maintaining, and presenting these equipment removal paths should be easy and time effective. The paths created should be able to be saved and used in conjunction with collision detection. The software should be able to simulate the removal process of ship's equipment. There should also be mechanisms for capturing these simulations in video format for use later in training and life cycle support.

**Collision Detection**

The ability to perform analysis to determine location and extent of interference and collisions between components describes collision detection. This process should be capable of being performed parallel to other capabilities. Methods for collision detection should be a core feature of the software and contain some method of compiling and retaining collision data.

**Manikin Technology**

Manikin technology includes the use of manikins to obtain detailed information on human interaction with the product model. This includes items such as adding scale to the product model, logistics functionality (e.g. How effective are the planned routes from general quarters to battle stations?), ergonomics functionality (What can this person see and reach?) and physical stress functionality (Is this task physically possible and/or what are the limitations?). The manikin technology should address the multi-level requirements of different users. For example, still images of static manikins may be needed for technical manuals, simple human interface simulation may be needed by design teams and detailed anthropomorphic ergonomic analysis may be needed for safety and man-machine interface design. Each of these levels requires a different set of functionality. For example, the design teams working on human simulation within the product model do not necessarily need all the overhead of a full-featured ergonomics model. The design team needs quick straightforward answers whereas the ergonomics analysis needs very detailed answers. The two different applications will most likely result in different levels of time performance.

**Desktop Visualization**

Many hours of preparation are typically needed to design and build a large-scale presentation for use on a high-end computer. This preparation will likely be performed on desktop machines. Thus,
Hardware and Software Options

Initial research indicated that there were two potential options in the market for large scale visualization: SGI’s Onyx 2 Infinite Reality Supercomputers and HP’s new Pixel Flow Technology Machine being developed in conjunction with the University of North Carolina. However, at the time of our investigation, HP had decided to abandon the Pixel Flow Machine because there was not a sufficient business case. Therefore, the SGI Onyx2 Infinite Reality was chosen for our large-scale tier. There are many different visualization applications on the market today. It should be noted that the performance and functionality requirements for every organization might be different depending on the nuances of the business requirements that need to be filled. In general, NNS looked at a variety of different applications. For our business requirements, we grouped the applications into several categories as shown below:

- Large Scale Mockup and Simulation
- Virtual Human Simulation
- Enterprise Collaboration

Later in the paper, we will discuss detailed requirements for evaluating various software alternatives.

Infrastructure and Security Requirements

For NNS our primary requirements are centered around the design and manufacturing of aircraft carriers and submarines. These involve very large enterprises that must deal with both unclassified and classified information. Again each organization will have its own unique requirements. For this discussion we would like to simply point out that a fundamental piece of implementing visualization for many companies will require research, planning and development in network technologies and their associated security issues. These technologies may include:

- Computer Hardware Requirements
- Secondary Hardware Requirements
- Signal Processing Requirements
- Encryption Devices
- Internal and External Networks (Servers, Hubs, Switches, Fiber, etc.)

Software Requirements and Testing Criterion

In the process of determining visualization software requirements, a quality function deployment process was utilized to drive software requirements from core business needs. The following section will briefly discuss some of the major requirements for NNS. The software was graded based on twelve of NNS’ most important visualization requirements. Each category had a weighted value and was given a score of 1-10. Short explanations of these requirements and the criterion on how they were scored are listed below.

High End Visualization (Real Time Fly-through)

The ability to smoothly fly through and manipulate large data sets constitutes a primary goal of visualization. The software must be able to move through large portions of ships in an impressive, real-time manner with little or no hesitation. The intent of NNS is to construct several visualization rooms throughout the yard for reviews with integrated product teams. Many of these reviews will be upper-level management meetings where a review of the whole ship will occur. Being able to load a large portion of the ship, if not the whole ship at once, is important. The user should be capable of flying
a requirement of desktop visualization is the ability to adequately run the software on a typical desktop machine with a large volume of data loaded.

**User Interface**

The user interface of the software can either guide the user through his/her options, or act as a maze that forces the user to endlessly search through a labyrinth of functions. It should allow an average computer user (with two to four hours of training) to create a replayable, simple fly or walk through. The controls should be straightforward, and should not require any special motor skills to use. In addition, the interface should have good on-line help documentation as well as a screen with the capability to be customized to suit the user’s preferences.

**System Integration**

System integration is the ability to integrate the software with existing NNS software that will be the data source for visualization. How well will the visualization software fit into ‘total visualization solution’? How well will the software fit into future NNS tools? How can the software be utilized to integrate the many different product-modeling applications used in the shipbuilding enterprise?

**Data Integration**

Data for projects may be supplied from multiple sources. The number of sources will likely increase as we move to a more collaborative environment. Data integration is the ability of the software to import data from these sources and integrate them into one cohesive model. Data Integration scales the entire visualization pyramid. It must be able to interface with multiple CAD systems, multiple PDM systems, and multiple graphics/product model standards such as VRML, STL, SLA, Inventor, DXF, IGES and STEP.

**Cost**

The cost of the software should be structured such that it supports the many different budgets and requirements of the organizations in the enterprise.

**Customer Support**

How well does the software manufacturer support the end user after software purchase? This is based on discussions with references supplied by the vendor or obtained through others means.

**Business Viability**

How well is the company currently positioned in the market? There are a variety of good sources that continually evaluate software companies relative to their technology, market share and balance sheet.

**Benchmark Data**

For each of the requirements, a benchmark was established as a method for comparing the different applications. This data represents empirical data gathered during the evaluation process. The scores are normalized to 10. The geometry/product model data should be based on an actual test scenario and model developed by the shipyard and supplied to each of the vendors being evaluated. Vendor supplied demos are wonderful for seeing the functionality contained in an application but it is our experience that the application must be stressed with real world data specific to your organization and industries processes. By developing your own bid package you can stress the application and
determine not only its strengths but also its weaknesses. Once the applications are validated and put into the weighted evaluation, a composite score can be derived for each application.

Conclusion

The objective of this paper was to provide the user with a high-level example of how a shipyard can go about the process of implementing large-scale visualization. Emphasis has been placed on the front-end evaluation work that is required in order to bring together the best solution for your company. Shipbuilding presents a very big challenge for visualization technology. Ships are the ultimate products when discussing large-scale visualization. No other industry really compares. However, because shipbuilding provides such an enormous challenge, the existing hardware and software solutions on the market today do not satisfy all of the requirements needed in our industry. Nonetheless, visualization technology is only beginning to emerge and NNS believes that this technology along with other distributed network technologies will shape the processes of ship design in the future.
GEOMETRIC MODELLING FOR
SIMULATION BASED SHIP DESIGN

John L Martin C Eng, MRINA, Marconi Naval Systems, UK

Introduction

Marconi Marine (VSEL) have implemented a Concurrent Engineering process with supporting information systems that are being applied to all current surface ship and submarine design build contracts. The process is also being used for concept design studies where Simulation Based Design technologies are being developed and evaluated.

A tree based product structure model is currently applied to ship detail design activities. This process has the functionality and data management capabilities for effective use on concept design studies where 3-D geometric models with associated behaviour attributes can be used to visualise the operational aspects of design options at very early stages of the design process.

3-D CAD modelling technologies and methodologies are well established throughout the shipbuilding industry as an aid to detail design, draughting, and manufacturing. Simulation tools and techniques are evolving for use in engineering and marine design to verify that design solutions meet operational or functional requirements. Combining the results of analytical simulation with 3-D geometric representation gives a high quality visual display for assessment of the design and its effectiveness in a range of operational scenarios. Using results of the simulation algorithms, the 3-D CAD model is animated in real or slow time to enhance visual presentation.

To achieve this the 3-D model must be constructed such that geometric objects may be associated with attributes, constraints and behaviour determined from physical limitations or operational specification. Simulation results can then be associated with the object geometry to visually illustrate the design in an operating condition.

Marconi Marine’s VSEL site have created 3-D models for use in a simulation based design process applied to studies to assess the operational effectiveness of aircraft stowage, preparation and sorting management on future Aircraft Carrier. The modelling techniques are also used for creation of general arrangements in 3-D to show vehicle stowage loading and disembarking options on Logistics Landing Ship concept designs.

Marconi Marine (VSEL) use the product structure trees and concurrent engineering methodologies for the creation of 3-D ship models to create an environment which is suitable for implementing simulation based design applications on future ship concept design studies.

Simulation

An English dictionary definition of Simulate is to “make a pretence of, copy, reproduce or replicate the conditions of a particular situation”. Under this definition, the application of simulation based design tools to the ship design process spans a wide variety of situations.
Simulation Overview

The most basic and simplest form of simulation is the creation of a 3-D model of the geometry of an object or set of objects that may be reviewed visually against some criteria such as spatial acceptability, shape, ergonomics, and/or operational feasibility. 3-D digital models have been created in engineering environments for many years as the alternative to a costly and time consuming physical model prototype. Model technologies used have evolved from full-scale wood mockups, through accurate scaled plastic models (1/5\textsuperscript{th} and 1/20\textsuperscript{th} scales being common) to digital CAD and Virtual Reality models.

Enhancement of the 3-D models to enable animation of objects and their component parts increases the value of the visual simulation and provides improved verification of the design as selected objects may then be assessed within their envelope of operation. The animation movements range from the simple action of opening a door for maintenance access and reserving ‘soft’ space for the extraction of internal parts, to complex actions such as proving a route for the withdrawal of an item of equipment from a compartment for replacement.

Behavioral characteristics which are calculated using accurate physics based algorithms further enhance the animation when detailed analysis is required on a chosen design option against original requirement and/or performance constraints. A simulation of the operational aspects of the design under a range of scenarios will give a representation of the physical behaviour of components and verify their suitability.

Real-time simulation in which the object behaviour includes time constraints and characteristics will give a time-based analysis that reflects a true to life operation of the object. When several interdependent objects are operating as a logical group, real-time simulation will show the sequence of events of each object and their effects on other objects in the group. The inclusion of the time behaviour allows interaction between the simulated objects and human operators which may be used in human factors studies.

Discrete event simulation also uses the behaviour of the objects for accurate calculation of operational scenarios; however in this case the interest is in the logical sequence of events and interrelationships between objects and their status rather than the effects of time. Discrete events are analyzed using event rules and operational processes which verify that the design satisfies all of the constraints and conditions which must apply before any action can be carried out on the selected object.

The analysis of human factor aspects of the design may be achieved by introducing a manikin model with behaviour algorithms that closely match human movements. For human operations to be assessed for operational effectiveness using simulation, the geometric models involved must have monitoring (diuls, gauges, displays etc.) and control (hand-wheels, levers, switches etc.) components detailed and positioned accurately. Figure 1 illustrates an operating position in which the operator needs access to turn levers from a standing position on a platform.

Simulation tools have been used and proven over many years in engineering design and shipbuilding for study of specific aspects of the design. Merging the simulation results from these tools is a desirable objective, and with use of modern geometric display technologies can provide a visualisation of the operation of the objects being simulated and their interrelationships.

The Challenge

The challenge is to use an Engineering Data Management/Computer Aided Design (EDM/CAD) system, normally used for static detail design modelling to create geometry data suitable for use in dynamic simulation modelling systems.
To effectively visualise design simulation studies, a process is required in which the geometric models are constructed to enable attributes and behaviors to be allocated to component parts for use when animating the model during display. This can be achieved if the geometry is decomposed as hierarchical assembly components that are controlled and managed through a product structure tree.

**Behaviour Modelling**

Ships and submarines involve many component parts that may have a simple or complex behaviour that can be verified by use of simulation tools. Figure 2 illustrates some critical items involved in the operation of an aircraft carrier flight deck which may be considered as items in which simulation can show their effectiveness during operation. Table 1 lists the model considerations required for each item.

**Table 1** Models and behaviour considerations

<table>
<thead>
<tr>
<th>Geometry</th>
<th>Modelling Requirement for Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft folding wing</td>
<td>Aircraft Model split into body plus wingtips</td>
</tr>
<tr>
<td>Aircraft lift</td>
<td>Lift modelled as platform with corresponding opening in deck geometry.</td>
</tr>
<tr>
<td>Wheeled vehicles such as aircraft tow-truck</td>
<td>Truck and tow-bar modeled separately, wheels modeled as component parts</td>
</tr>
<tr>
<td>Fork lift truck</td>
<td>Steering wheels, forks modelled as component parts of truck</td>
</tr>
<tr>
<td>Weapon stowage box containing missile.</td>
<td>Box model with separate lid and internal weapon</td>
</tr>
<tr>
<td>Aircraft Arrestor wires</td>
<td>Pulleys. Modelled with separate geometry for wire</td>
</tr>
<tr>
<td>Helicopter rotor blades and tail – folded and operation position</td>
<td>Fuselage, rotors blades and folding tail section modeled as component parts of assembly</td>
</tr>
</tbody>
</table>
Geometric Modelling

To create a 3-D definition of the layout of a ship design, a product structure tree may be built with selected nodes allocated a geometric reference that points to the appropriate geometry libraries or database to extract the geometry. The product structure also defines the location and orientation of each item of geometry used to give a true representation of the physical design. The geometry consists of the ship structure and steelwork, equipment and components, connecting services and operational items.
3-D models that are to be used for displaying simulation results must be constructed such that all component part geometry affected by the simulation is independently modelled and allocated a unique identifier. This will ensure that the parts may be selected and manipulated as necessary. The form and content of the component part geometry is dependent on the level of simulation to be undertaken. For simple general arrangement layout it is sufficient to create one-piece geometric components and locate them as required within some boundary as illustrated in the simple layout in figure 3.

As a design evolves, several options are considered when trying to achieve stated requirements, the optimum solution will be selected after full analysis of the options against the requirement criteria. Using the product structure tree, each option may be modelled and the appropriate supporting data attributes allocated to tree nodes. The optional solutions can then be reviewed by selecting one or more subsets of the product structure tree and analyzing the results.

**Animation**

For animated components, the geometry must be further decomposed to model the parts to be animated in the simulation, for example an aircraft lift will be modeled as the platform and the critical component parts of the lift assembly. During simulation the mechanisms will be animated in accordance with the behaviour calculated from the kinematics see figure 4.

![Figure 4 Aircraft Lift for Simulation](image)

**Design options**

When design options are to be considered, a product structure tree is used to define a common origin for the options as a tree node and then define the options as sub-nodes. Each option can then be selected from the list of sub-nodes for display. The design option technique may be used to verify the effects of each of the possible options during design review with simulated assessment of the
operational effect of each option analyzed and either the appropriate option selected or several options proven. See figure 5 and figure 6

**Figure 5**  Small superstructure option

**Figure 6**  Large superstructure option

*Simulation Considerations - Physical Characteristics*

Physical items that form component parts of the ship have associated restraints or constraints, which affect their behaviour in operation. The constraints may be directly associated with partial geometry such as a hinge along one edge of a door or a rotating component on a shaft. They may be associated with movement of an object having one or more degrees of freedom, or may involve a change of form of the object itself.

To simulate the physical behaviour of component parts of an object, the object must be represented by a model constructed such that all geometry affected by the simulation is individually
identified and controlled. A product structure tree has the functionality to enable this to be achieved as well as allowing behaviour attributes to be allocated to all parts.

For complex simulations in real time, the component geometry being simulated will have location and orientation provided at specific time intervals from the simulation calculation to convey the impression of movement within the model. To maximize performance, the component geometry in this case will be optimized in level of detail (for example, the simulation of an aircraft on an Aircraft Carrier will use an aircraft geometry which minimizes level of detail whilst providing a reasonable representation of the aircraft see figure 7).

Figure 7 Aircraft on a Aircraft Carrier

To manage the geometry models for components and their individual parts requires a structured database with individual controls for each part and a hierarchical relationship between the parts. This can be achieved using a product structure model as provided in modern EDM/CAD systems. Furthermore, if the simulation involves mixed discipline geometry of large complex data sets such as ships and submarines then well defined database technologies and concurrent engineering philosophies are desirable to guarantee the data capture and access required.

The Product Model

Concurrent Engineering

The implementation of concurrent engineering is well documented as being a complex task involving a combination of people, processes and technology issues. The concept and ideas that enable concurrent multi-discipline working demand the flexibility and functionality to decompose component parts of a design into a logical hierarchy for controlled data management, access and sharing. This gives the option of attaching attribute data and conditional status to individual, logically grouped or physically grouped object representations.

The ability to allocate non-geometric attribute data to the nodes of a product structure tree and control the access and extraction of selective information from the hierarchical structure during design development allows object behaviours to be recorded for each component part of the design.

In the concurrent engineering environment customer and suppliers contribute to the operational aspects of the design requirements for the vessel and its component parts using their knowledge and
experience to specify the expected behaviours of systems, individual equipment and the ship itself under stated operational conditions. This information can be managed through a product Model.

**Single Source Data**

The requirement for single source data is critical to successful implementation of concurrent engineering. All data users to ensure consistency, accuracy and effectiveness share the single source. The data capture, management and control must be arranged such that individual or selective groups of data are owned by a pre-determined creators whom have full write and edit access and accountability for the quality and content. The data will then be extracted and used by project team members with read only access authority.

In practice the management and control of a single source of product data may be achieved using the product structure concept. The product structure also allows geometric data and associated behaviour attributes to be stored and managed in a logical manner making it a good tool for dissecting geometry into component parts, ideal for simulation.

**Product Structure Tree**

The product structure tree manages the configuration of the model, its operational characteristics and design options and specifies the relationship, location, orientation and function of all items on the ship. Optional design details may then selected for review, comparison and manipulation or an advanced visualisation (VR) display technologies.

Data review, approval and release of all information that defines a product can be achieved through the product structure tree with the status of individual or grouped subsets of the data recorded as attributes on the tree nodes. Using a product structure tree, the design configuration can be maintained from concept design, through production and in-service life of the product see figure 8

![Figure 8 Extract from Product Structure Tree](image)

**Attribute Data**

The definition of a ship involves a very high volume of data in a wide range of formats and media types, all of which contribute to the final configuration of the product design. The data will include requirements, specifications, design calculation, systems definitions, equipment lists, cost estimates, applied standards, build criteria and the geometric layout of the vessel. In concept design, feasibility studies and design class evolution, the definition may have several optional design solutions from which the optimum is selected for a particular product. The product structure tree can manage the configuration and all of the attribute data associated with the design giving a single well-controlled specification of the product, its component parts and their status at any time during the design cycle.

A product model based on a hierarchical tree structure offers the functionality to capture, and manage design attributes and options as component parts, as required for input to simulation tools.
Project VITESSE – A Practical Example

Overview
The Ministry of Defence (UK) department of Naval Architecture and Future Projects have a vision to adopt Simulation Based Design in the procurement process for future warships. The process will merge the simulation results for specific detail of concept designs to verify the operational effectiveness of the options studied. To achieve the vision, the project, named VITESSE chose the CVF UK future aircraft carrier concept design options to illustrate the application, accuracy, and potential use of simulation and interface technologies to prove the operational effectiveness of the ship.

The objective of the project was to create an environment using 3-D geometric modelling with the capability to display concurrently, and/or in real time, the results of several simulation studies involving aircraft sortie rates. Simulators used for flying the aircraft, the behaviour of the ship in various sea states, and human factors in preparing aircraft for flight were remotely located and merged over a distributed network using DIS technology with visual display of the simulations at each site.

Project Architecture
To achieve the objective of the project, several industries were invited to contribute their skills and expertise to the project. The principle tasks were to develop the ship hull form and major subdivisions, create a 3-D CAD model of the ship. To calculate the physical behaviors of the ship and its components, link the ship model and its simulated behaviors to aircraft flight simulators and present a combined display showing the aircraft landing on the ship, refueling and weapons loading and take off operations.

Contributions from industry involved companies working closely to develop data models then convert them and use them in a range of simulator tools, the resultant architecture as shown in fig 9.

Figure 9   Project Architecture

```
Hull form & Major Subdivisions

Visualisation Display of 3-D Model

3-D Geometric Model

Simulation Models

DIS Network

Aircraft Flight Simulator

Other Simulators For example Sea-states, Weapons

Aircraft Flight Simulator
```
Data Conversion

Geometric representations of ship designs are created using computer aided design (CAD) systems. Modern CAD systems capable of handling large complex data volumes use hierarchical structures to manage the design geometry. Assembly Modelling and/or Product Structure Trees are used to manage the relationship between the component parts of the design.

When converting the design from the source CAD system to a visualisation/simulation environment, the assembly/product structure relationships are critical in defining the design and must be converted accurately along with the geometry files for the components. The hierarchical structure of the assembly or product structure may contain the location and orientation data essential to create the assembled model, failing to convert this data results in all component parts of the geometry being located at the model origin.

A conversion process from an assembly based CAD system involves two steps, first the assembly model or product structure tree is converted to the target system, then all of the component parts used in the assembly are identified, extracted from the CAD library and converted.

In the VITESSE project, three visualisation technologies were evaluated and the master CAD data converted for input to each. One converter, provided by a system vendor and used by Marconi Marine for design review visualisation, considered the product model structure and delivered a usable converted model for visualisation/simulation with minimum effort and time. Two other converters supplied from specialist third party companies did not consider the product structure with the result that all part geometry was converted to the target system format and located at the model origin. Manual correction was required to redefine the location and orientation of each geometry item on the ship in accordance with the product structure data.

Scope of Modelling Tasks

Marconi Marine (VSEL) was allocated the task of generating the 3-D geometry, which would be used to evaluate several simulation and visualisation systems during the VITESSE project. The scope of the work was to deliver models for three concept ship designs that could be used to study operational aspects of the aircraft and their sortie rates. To do this, the models addressed exclusively the flight deck, aircraft hangar and magazines. The task involved input of a hull form and creating product structures and geometry models for areas to be studied.

The main input to the process was an IGES format wire frame model of hull forms. 3-D solids/surface models were then created using Marconi Marine (VSEL) Concurrent Engineering processes based on ‘Parametric Technology’ Optegra/CADDS5 systems with visualisation using ‘Division Ltd.’ dVise VR products. Output to third party companies for simulation studies involved data formats including CADDS5 (with product Structure tree), dVise, Open Flight and IGES files.

The Flight Deck details with major markings, superstructure, operational items for aircraft launch and recovery, aircraft lifts, maneuvering and other service vehicles, and other operational equipment were modelled in the CAD system with a component breakdown to suit the simulation behaviour criteria.

The aircraft hangar and magazines were detailed in the CAD system. Magazine weapons stowage was represented by simple block models of groups of weapons cases with one case of each type modelled as box, lid and weapon to enable weapons handling simulation and animation.

In order to effectively visualise the model using standard display technologies, the geometry representing moving components such as doors, blast screens, and lifts were modelled in open and closed condition, selected via the product structure tree, with one or the other switched on for viewing at any time. This enabled the CAD data to be displayed in optional positions/conditions without the need for simulation software or advanced visual tools.
Spatial Layout

Spatial layouts of ship arrangements are have been created on CAD systems for many years and the techniques and processes to achieve an acceptable arrangement are well proven. However, at the concept design stage several options need to be considered and their effects taken into account on the overall requirements for the ship. To address the issue of a mix of aircraft types on the ship, and the requirement that each will need to be parked, moved, stored, maintained and serviced, the model needs the flexibility to show each of the potential aircraft at all possible operating spots.

This is achieved in the EDM/CAD system by using the product tree to identify the location of the aircraft with a reference node and define all possible aircraft as tree nodes that are created as ‘children’ of the reference node. Any one of the aircraft options can then be switched on for display, and the spatial requirement for all possible aircraft can be considered. Using this principle figure 10 and figure 11 show the effect of selective aircraft display showing two of the many options.

Figure 10  Selective Aircraft Display I

![Figure 10](image1.png)

Figure 11  Selective Aircraft Display II

![Figure 11](image2.png)
**Kinematic Simulation**

The operational process for aircraft movements on the flight deck of an aircraft carrier requires complex kinematic calculations involving a four wheel steer truck as shown in figure 12, a linkage bar with hinged connections at both ends, an aircraft turning wheel, and the turning behaviour of the aircraft itself.

![Figure 12](image1.png)

**Figure 12** Tow-truck Photograph (showing linkages)

To visualise the results of a simulation of the above process, behaviour attributes must be associated with the geometric model components affected by the calculation. For example, the tow-truck wheels, the tow-bar linkages at each end, the aircraft wheel, and the truck and aircraft bodies will each have turning and maneuvering characteristics. The 3-D model of the truck is thus created to ensure that moving components can be manipulated independently of the main body geometry. See figure 13

![Figure 13](image2.png)

**Figure 13** CAD Tow-truck with linkages
Project Results

The VITESSE project completed successfully in March 1999 with a demonstration to senior MOD staff of aircraft carrier models for two concept ships and the in-service ship HMS Invincible. The demonstration showed the ships in various sea states, the maneuvering of aircraft to service points, aircraft fueling and weapons preparation. Qualified pilots in a remote flight simulator carried out landing and take-off operations on the carriers. The ship geometry being displayed in the flight simulator and the position of the aircraft displayed in the ship simulator, controlled by data messages sent along a network using DIS (distributed interactive simulation) technology.

The objectives of building CAD models as ‘master data’ for a simulation exercise were achieved, with the functionality of the product structure tree playing a major role in the flexibility of the geometric data. This gave the ability to model and manages several options for selectively review and consideration from very early stages of ship design. The CAD data was converted to a variety of visualisation/simulation technologies with variable levels of success.

Simulation of the kinematics of the tow-trucks and aircraft was also achieved, the simulation engineer suggesting that models be decomposed into appropriate component parts prior to simulating. The real time exercises involving flight simulators were achieved with some difficulties in response caused by the complexity and size of the CAD models. Although the models were created with considerations for minimizing levels of detail, they were compromised to give reasonable visual representations.

Figure 14 Aircraft Approaching for Vertical Landing

The Future

The success of the VITESSE project in developing models, simulation applications and a distributed simulation network demonstrated the technology as feasible. VITESSE will be followed by an in depth study over several years on application and use of the technology as an aid to warship procurement.
Conclusions

General

Creation of the CAD 3-D models for the VITESSE project involved some consideration of the requirements of simulators to define component parts of assemblies and allocate behaviour to them as appropriate. When creating EDM/CAD models for the layout of ship compartments the principle consideration is spatial arrangement of equipment and services, which requires representative geometry rather than high definition manufacturing detail. Use of a product structure tree representation of a ship is also acceptable to shipbuilders who often work with structured work breakdown definitions such as weight groups.

The only aspect of concern when preparing CAD geometry for simulation is to identify the component parts to be considered during the simulation and break down the product structure to a level to suit this requirement. The process is thus a minor deviation from normal shipbuilding practice, the costs in resource being to add additional nodes on the product structure tree and decompose the CAD model into component parts (we used to do this often with CAD layering technology!).

In terms of real-time simulation, such as the flight simulators feeding positional information through DIS, the major consideration is for the animated geometry (in this case the aircraft) to be modelled with the lowest level of detail that gives acceptable visualisation. This will maximize visualisation performance.

Specific

The use of simulation based design tools and technologies is evolving as a stable environment which may be applied with minimum effort over and above the current work of preparing and executing independent simulation studies on critical aspects of the design.

The development of EDM/CAD systems with structured data management and attributes associated with geometry offers the controlling mechanism for defining databases which can be used to detail component parts of assemblies with relative ease.

The high performance visualisation systems offered by CAD and Virtual Reality vendors can display large data models and have the capability and performance for real-time visual display.

Combining the above statements suggests that simulation based design is achievable, the technologies involved have been proven and are further developing (DIS network is being replaced with more powerful HLA technology for example). In addition the process required to prepare the data does not deviate from normal ship design practice other than to add more detail to selected components.

Concept Design

The VITESSE project vision is to develop a simulation based design environment suitable for use in ship procurement. This implies that the simulation based design activities will start at the very early concept stages of the design process. The technology is available to achieve this, as illustrated by the VITESSE project. Naval Architects and Ship Designers, however, need to adopt the use of product modelling and CAD 3-D arrangements to replace traditional 2-D drawings in order to gain advantage from the process and technologies.

Developing concept designs using 3-D arrangements in a modern product structured EDM/CAD environment has the advantage that geometry created will be stored in the database and may be selected for use in future designs. Adding appropriate generic attributes to this geometry will develop a library of re-usable ‘blocks’ which may be selected and used in future design considerations.
and adopted in a block based design philosophy to create quick, accurate and flexible design options at concept stage. Verifying this with simulation results gives a high confidence of the expected performance of the design options studied and a means of selecting the optimum design. If the 'block' attributes include cost models and build considerations, selecting an optimum design will not only consider design expected performance against requirements, it will also consider costs against performance.

**Summary**

Simulation Based Design is an evolving technology that may be used from the early stages of concept design to select from several potential options.

The underlying EDM/CAD processes do not deviate from normal shipbuilding practice and may be brought forward to the concept design stage. If the selected concept design becomes a contract, then the product structure tree can be extended through the design build cycle.

The VITESSE project has shown the technology to be feasible.
USING PROCESS MODELS AND INTELLIGENT AGENTS TO SUPPORT COLLABORATIVE ENGINEERING IN SHIPBUILDING

Masahiro Sonda, Ship & Ocean Foundation, Tokyo, Japan
Yutaka Takemoto, Ship & Ocean Foundation, Tokyo, Japan
Hiroyuki Okabe, Ship & Ocean Foundation, Tokyo, Japan
Yan Jin, University of Southern California, Los Angeles, USA

Abstract

In this paper we discuss collaboration support for engineering activities in shipbuilding. Ship design is a complex process and involves multiple engineers to work concurrently on different parts of an overall design. The demand of short lead time has created a highly concurrent engineering practice in the real world.

We take a process-based approach to achieve coordinate engineering activities. In this approach, engineering activities are prescribed by a set of pre-definable and dynamically updateable active process models. An agent-based approach has been taken to implement the active process models. Each agent is associated with an engineer. It helps the engineer to get information from process servers, and report status of, and the exceptions generated from, the engineer to the process servers.

We will present active process models and the framework of process-based collaboration support. After that, we will show a collaboration support case example that uses the proposed framework for collaborative engineering support to prove effectiveness of our approach.

This research is one of the main works in on-going ACIM(Advanced CIM) project, which is funded by Ship and Ocean Foundation with participating 7 major ship-building companies in Japan[1, 2].

Introduction

Large scale engineering problems, such as ship design, require multiple engineers to work on different parts of an overall problem. At any time, different engineers may work on interrelated tasks—i.e., the results or decisions of one designer may have significant impacts on those of other designers. To prevent inconsistency in results and reduce redundant activities, engineers must collaborate effectively so that information flows correctly and timely and engineering activities are well coordinated.

Sharing engineering information through a common product model has been proven an effective way for engineering team members to identify inconsistencies and generate timely information flows. By accessing a logically centralized product model database, an engineer can retrieve the latest engineering information generated by other engineers and pass his or her task results to others through the database. While shared product model can be used to facilitate information flows among engineers, it contains only the results generated by engineers. The information of the process through which the results were generated is not part of a product model. Furthermore, the control information that defines how engineering activities should be carried out and how they related to each other is also missing from most product models.
In this paper, we propose a framework based on a process-driven and agent-supported approach to collaborative engineering support called ‘Active Process.’ Active Process has a capability to describe ‘how engineering tasks should be done’ in order of precedence. It is powerful but still not sufficient because engineering task procedures will be often changed according to situation. To meet this problem, agent-supported approach is introduced to Active Process[3, 4]. Each participants in the system has its own agent who is interested in negotiation with others to acquire information. Agent has intelligence to modify pre-defined process based on negotiation and can offer participants useful information to ask them for instruction if the situation is too difficult to judge.

The basic ideas behind this approach are:

1. Engineering activities can be prescribed by a set of pre-definable and dynamically updateable process models;
2. These process models can be used as points of reference as well as points of control for engineers to coordinate their engineering activities; and
3. Intelligent agents can be developed to support the “referencing” and “controlling” based on the predefined process models.

The overall goal of our research is to develop a process-driven and agent-supported framework collaborative engineering support and test framework through prototyping and applying it to example scenarios. The specific goals of the research include:

1. To clarify requirements of process-driven collaborative engineering support.
2. To identify process elements and develop a process model for engineering support
3. To define a process-driven work paradigm based a set of engineering task scenarios.
4. To develop intelligent agents and an agent-network for monitoring and controlling the engineering processes based on the process models.
5. To test the concepts and the framework by developing prototype systems and testing scenarios.

To achieve our research goals described above, we are taking a modeling-prototyping-case study approach.

**Process-Driven Work Paradigm**

To support engineering collaboration, we propose a process-driven work paradigm. There are two basic research issues that must be addressed in order to realize the proposed work paradigm. One is how to model the process information, and the other is how to apply and manipulate the process information for engineering work and collaboration support.

**The Need of Process Model**

While knowledge about engineering products captured by product models is important for engineering support and needs to be shared among distributed engineers and systems, another important aspect of engineering knowledge is about engineering processes. Engineering tasks are not carried out in a random way. Rather, they are well planned and managed based on the planned processes. An engineering process is composed of a set of interrelated activities that collectively realize certain engineering objectives. A process model is a representation of an engineering process in
a form which supports automated manipulation or enactment by a process management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities such as participants associated IT applications and data, etc.

From an engineering point of view, process models are needed to support engineering process design and planning, process knowledge sharing, and engineering collaboration. The goal of engineering process design is to generate a document that prescribe a set of tasks and the ways in which the tasks should be carried out by whom using what tools. A well defined process model will help process designers and managers identify key activities, required participants (resources) and define their relationships. Since process models can be made machine readable, it makes it easy to use process management systems to manage process execution. Furthermore, the process information in a process model can help coordination among engineers. Engineers can share the process model so that they are aware of the whole engineering process, in terms of who is responsible for which activity, who is busy, who is advanced and who is behind. This information about current situation can help engineers determine whom to ask for information, to whom to route a certain task, and who needs help.

**Process-Driven Engineering Task Management**

As described above, a process model is a representation of a business or engineering process in a form that supports automated manipulation, such as modeling, or enactment by some process management systems or programs. Figure 1 illustrates how process information represented as process models can be used to support engineering work and collaboration.

**Process design:** Engineering process design starts from putting together process ideas (e.g., relevant processes, activities) about a specific engineering task. The ideas may come from experience or from a process repository that contains all known process ideas about an engineering domain. As shown in Figure 1, the process ideas are applied to define a specific process model.

**Execution, Work and Collaborate:** Once a process model is determined for a given task, then the model is used to guide the execution of the process by human engineers, project managers as well as computer controlled systems. Based on the task specification described in the process model, each engineer carries out his or her task and coordinates with others when needed. Interdependencies predefined and dynamically generated are explicitly represented in the process model for collaboration support.

**Monitor, Report, Evaluate:** The execution process information can be monitored by certain systems, e.g., intelligent agents, or reported directly by engineers. The execution information can be used to evaluate how well the engineering team is following the process plan. For example, by comparing the progress information with the original process model, it will become clear whether the project team is ahead of the schedule or behind.

**Control and modify:** One output of the evaluation is the instructions or control of the project team to improve their performance. If the evaluation result shows that the process model does not fit the current project team well, then the other output of “evaluation” will be to modify the original process model to make it more achievable by the project team. This loop, middle loop in Figure 1, of process model application is quite similar to a feedback control system in which the process model is used as a reference or input to the system and the monitored information is used as the feedback for generating control instructions.
Improve and Store: Another way to use the monitored process information is for “long term” process knowledge improvement through learning, as shown in Figure 1. The accumulation of the monitored process execution information can provide important insights on how effective and efficient the original process model is. By analyzing the information through benchmarking the executions of different process models, one can acquire new process knowledge and improve understanding of the known process models. The new knowledge can then be stored into the process repository.

Figure 1. Process Model and its Application for Engineering Support

Those 2 feedback loops described above are based on process model, which are defined in advance. But it is very difficult to pre-define all the processes because of the complexity in shipbuilding. So it is preferable to get dependencies in processes dynamically during operation. Using intelligent agents, we introduce this function which is represented as the inmost feedback loop in Figure 1.

Agent-Based Work Support

To make the task management framework a real engineering work support system, the issues of systems design and implementation must be addressed.

Figure 2 illustrates the conceptual system image of Active Process System (APS).

We have identified three important system components that will be explored through this research. The components are: Intelligent Agents, Process Server and Enterprise Resource Server.

Intelligent-Agents: Most important function that intelligent-agents will do is to extract dependencies in processes from monitored results dynamically. It can compensate uncertainty or complexity of shipbuilding engineering processes. Intelligent Agent can facilitate appropriate local operations service for its engineer as well. This may include local application management, operation bookkeeping and communication management. And the agent can function as a systems interface between Process Server and human engineers. This function makes it possible to make the Process
Server more general and leave engineer specifics to Intelligent Agents to manage.

**Process Servers:** The functions of Process Servers include capturing process models and managing process execution based on the process models. Process models must represent "planned (or intended)" processes, "simulated" processes, and "executed (or real)" processes. For process execution management, Process Servers must be able to generate activation signals, control the real process by reasoning based on the given rules and the data of "planned", "simulated" and "real" processes. Process Servers may be organized in a hierarchical way so that the higher level servers can take care of more "global" issues of the process while the lower level ones can manage activities at more engineering or task specific levels.

**Enterprise Resource Server:** To manage engineering processes, resource information is very important. Enterprise resource server centralizes resource information logically. It has also process templates and journal information. Process templates are typical pattern of process combination. Those will be enhanced through the outmost feedback loop in Figure 1. Journal information means the results of engineer's operation or communication. Dynamic dependencies may be extracted from journal information.

![Diagram of Active Process System](image)

**Figure 2.** Active Process System

### Process Modeling

The purpose of process modeling is to develop an ontology, i.e., a set of concepts and their interrelationships, that can be used to capture process knowledge and information and to manipulate (i.e., design, monitor, control) the process information. In this section we show the process description language in Active Process.

Active Process is a dynamic work process model. Unlike existing process or workflow models, Active Process is intended to be able to meet not only pre-definable process, i.e. activities(defined later) and dependencies among them, but also unforeseen, finer work unit
dependencies. Our model has description capability in compliance with granularity of tasks.

One feature of Active Process is that processes will be linked to product model data. Product model represents component data which comprise a product and their inter-dependencies. Execution of process results in the evolution of product model data.

**Active Process Model**

Following components are addressed in the Active Process model.

1. **Process:** A process is composed out of a set of activities.
2. **Activity:** An activity is a unit of behavior of a process and used to describe a process at most abstract level for process design. Detailed dependency relationships between the activities should be investigated for collaboration support.
3. **Resource:** A resource is an object that is used or produced by one or multiple activities. Product is a specific resource.
4. **Dependency:** A dependency relation defines relationship between activities, or an activity and a resource, or resources.

**What is Process?**

The definitions of process can be given according to the following two view points. As a description model view point, a process is defined as a network of interrelated activities. As a control mechanism view point, a process is defined as a set of continuous or discrete behaviors that can be observed and controlled. As general features of a process are as follows. For description and documentation, a process expresses concepts, relations, requirements and properties. For enactment, a process has the information related to the behavior, performance, environment and feedback (learning).

**General Characterization of Process**

The process is generally characterized as follows;

\[ P(t) = \{ A, U, D, R, Q ; t \} \]

Where,

- **A:** activity, \( A = \{ a_1, a_2, \ldots a_I \} \)
- **U:** resource, \( U = \{ u_1, u_2, \ldots u_K \} \)
- **D:** product, \( D = \{ d_1, d_2, \ldots d_T \} \)
- **R:** relation, \( R = \{ R^a, R^u, R^d, R^{ad}, R^{ad} \} \)
- **Q:** requirement, \( Q = \{ q_1, q_2, \ldots q_N \} \)

- **R^a:** activity relation, \( R^a = \{ r_{1}^a, r_{2}^a, \ldots, r_{F}^a \} \)
- **R^u:** resource relation, \( R^u = \{ r_{1}^u, r_{2}^u, \ldots, r_{G}^u \} \)
- **R^d:** product relation, \( R^d = \{ r_{1}^d, r_{2}^d, \ldots, r_{H}^d \} \)
- **R^{ad}:** resource assignment relation
- **R^{ad}:** product assignment relation
- **R^{ad}:** relation between resource and product

**Implications of Process**

We have to notice the implications which process has. \( P(t = t_0) \) is a “complete” description of the process state at \( t = t_0 \) can be described as a static process description. The analytical properties of a
process can be addressed. From a P(t)'s point of view, the unit of analysis is Activity. On the other hand, process is a continuous behavior changing from \( P(t = t_0) \) to \( P(t) \). The above mentioned characters of process, \( A, U, D, R, Q \), all evolve and/or change over time. This means that addressing \( P(t) \) requires behavior models of \( A, U, D, R, Q \). Also, this need to move unit of analysis from Activity to components of Activity.

Another important implication of process concerns about \( R \), relations. The complexity of a process \( P \) is largely determined by \( R = \{ R^e, R^r, R^d, R^w, R^o, R^{od} \} \). Also, the uncertainty of process comes from \( R \), relations. Practically, a complete information of \( R = \{ R^e, R^r, R^d, R^w, R^o, R^{od} \} \) is not available both in \( t = t_0 \) and \( t \). To acquire more information about \( R \) requires interaction and also costs time and money. Furthermore, \( R \) changes over time. This means that better or lower level model of \( R \) is needed.

### Activity

Activity is a unit of a behavior of a process. The definition of an activity is as follows:

\[
an = \{ o, d, u, q; B \}, \quad \text{where} \quad o = \text{action}, B = \text{Behavior}
\]

The function of an activity can be described as follows:

- Function = \{\( o, d \)\}, e.g., \{design, hull\}
- Action \( o \) is resource relevant concept.
- Behavior \( B \) means manipulation and completion of activity components. Behavior \( B \) can be described by following method:
  - complete behavior: differential equation based
  - computational behavior: discrete event simulation
  - Proximate human behavior

Those are the formal definition of Activity. This model is intended to be applied to ship-building tasks. From the viewpoint of application of the model to tasks in ship-building, we assume Activity to be corresponding with the manageable granularity of tasks.

Moreover, we introduce \( \text{WorkElement}, \ Event, \ Exception \) and \( \text{Performance} \) in our modeling concepts.

### Work Element

The concept of Work Element has been introduced to express the running of Activity or the behavior of Activity. An action \( o \) is performed by an actor/designer on a set of components \( d^i, (i = 1, \ldots, S; d^i \subseteq d) \) of product \( d \).

A work element \( w^i = \{ o, d^i \} \) is defined as a component of an activity \( a \), i.e., \( w^i \subseteq a, \quad i = 1, \ldots, S \).

The reason why a work Element \( w^i \) is not an Activity are as follows;
- WE is too simple: All \( w^i = \{ o, d^i \} \) are within individual's reach
- WE is too complex: Because \( R \) cannot be clearly defined so that teamwork is needed. If all \( R \) are clearly pre-defined, each designer can proceed his/her task according to the predefined instruction.

The status of Work Element is basically described as one of two conditions, i.e. pre-start and completed. The requirement or possibility to describe intermediate status of Work Element depends on domain & problem. Work element relations composed of intra-relations in a Activity and inter-relations between Activities. That is, a Work Element may have a relation with other one belonging to the same activity, or other activity. Work Element relations are essential model to capture underlying
Mechanisms of activity relations. The relation is dynamic. It is generated when recognized/identified. And it becomes strong/weak when manifested in a process.

Work element is defined as a unit of work that can be measured or evaluated based on the unit progress of product information ($d^*$ in the definition of $w^*$) being generated from engineering design. Work element is introduced to link product model with process model for work and collaboration support.

The connection of process model and product model is a significant feature of ActiveProcess. Product model has a complete data set for a product while it has not information of process. That is, product model has information on 'what a product is' but has not information on 'how to make it'. Process model offers that information. On the other hand, the progress of process can be represented based on the object state of product model data. Those two models are complementary to each other.

**How Do Relations Evolve?**

The relations are identified by working together at work element level. As the process proceeds, new relations are detected through the fact that two designers have contact each other. If the relation once identified are manifested again, the relation is strengthened. Figure 3 illustrates this situation schematically.

![Figure 3. Relation between Work Elements](image)

**Events**

The concept of event is used for process monitoring. An event $e$ is a notable and noteworthy happening in an activity $a$ and associated with one or more work elements $w^*$. An event compound $e'$ is a recognizable set of events that can approximate an recognizable action $o$.

$$e' = \{e_1, e_2, \ldots, e_k\}; \text{ } o = e'$$

Knowledge is needed to generate $e'$ and $o$ based on observable events.

Events $E$ represent observable behavior of activities. The underlying mechanism of the $E$ is not assumed, i.e. events can be outputs of a simulation program or can be human actions.

Interpretation of events means to infer "what needs to be done" from $e$. 1-to-1 mapping based interpretation is useful for fully. On the other hand, the reasoning based interpretation of events may involve uncertainty as shown in Figure 4.
Exceptions

Exceptions are unexpected situations when executing a plan. In Active Process, exceptions are departures from a planned/ideal project process. In Active Process, agent detects/anticipates the exceptions from exception category. While exceptions can occur for many reasons, it is expected that they have a relatively small number of kinds of categories that include. Example of categories may be as follows:

Inquiry:
- Ask for status of work elements
- Prediction
- Communication failure

Change:
- Design status change, change of design process
- Requirement q change

Wrong doing:
- Break commitment

Conflict:
- Critical and non-critical conflicts

Figure 4. Interpretation of events

Active Process System Architecture

Active Process has two aspects: process modeling capability and management capability for modeled process. As described above, it is very difficult to define all the process precisely in advance. To overcome this difficulty, it is indispensable to collect or modify process dynamically during operation. Therefore we adopted agent-based approach as Active Process system architecture.

To collect dependencies dynamically, our system is modeled as collection of intelligent agents each responsible for one activity or task. Since each agent can work autonomously on behalf of human designer, they are expected to identify both problems and opportunities with the current design process. The initial definition (or design) of the design process will not be followed as a static template. Instead, it will be actively revised and updated by agents who are interested with each other to acquire information and to negotiate revision. We implemented prototype system to examine those ideas.

In this section, we show Active Process system architecture based on agent approach.

Active Process System

Figure 5 illustrates the prototype system architecture. Intelligent agent is assigned to each component. Designer + Intelligent Agent is 1 client unit. Communication is performed through agents. All the information which is necessary for collaboration among engineers (i.e., dependencies with activities other engineer does, progress state of other activities, state transition of product model data, etc.) will be captured by an agent instead of an engineer himself. Those information is stored in
Enterprise Resource Server (explained later) as engineering task journal. Components of the system are detailed below.

![Diagram of system architecture](image)

**Figure 5.** Prototype system architecture

*Function of Intelligent Agent*

Intelligent Agent is a state-of-the-art IT[5, 6, 7]. Intelligent Agent is regarded as a core technology to realize engineering collaboration support system using process model. One of the feature of Intelligent Agent is to support system operation and participants in the system intelligently and autonomously. Agent's intelligence does not mean all the knowledge that engineers will depend on when they meet engineering tasks. In this architecture, Agent is a kind of facilitator who may provide engineers useful information, i.e., state of other engineer, past examples, alternative options. Those information will help them to make decision. Agent has learning capability as well. It has knowledge that what information should be offered to its engineer depending on cases, and this knowledge will be modified and enhanced through the interaction with its engineer or other participants. As for process model, this learning capability enables to obtain dependencies among WorkElements dynamically.

In Active Process system, Intelligent Agent has four functions as follows:
1. Communication function with other agents.
2. Event Monitoring function.
3. Exception Detection function from events information.
4. Task Execution function in compliance with events/exceptions.
Intelligent Agent Architecture

Figure 6 shows architecture of Intelligent Agent implemented in the prototype system. It consists of 3 modules: communication function module, message interpretation function module and local service module. Intelligent Agent is implemented with Java.

KQML (Knowledge Query and Manipulation Language) is used as communication protocol. Contents of message are expressed in the form of KIF (Knowledge Interchange Format). Terminology in messages conforms to domain specific ontology. The part which is domain specific is just only ontology. This means that Intelligent Agent architecture is applicable to any domain just only to change ontology.

Process Server and Enterprise Resource Server

Process Server has three functions. It defines a process (activities and inter-relations among them), supports resource assignment and monitors progress state.

(a) Process definition: Activity level process is defined through GUI. It may be generated newly or may be extended based on templates which will be stored in Process Repository. Processes defined in Process Server are able to be separated and spawned to several client participants recursively. That is, one process may be assigned to an engineer and he/she can separate it into pieces and assign them to other engineers. Information defined here are as follows: task contents, schedule, dependencies, responsible workers, etc.

(b) Resource assignment: Available engineers lists and machines lists are available as resource information. Those resource information will be referred during process definition. Processes are optimized during operation if idle resources (we call them slacks) are found. We call this approach slack driven process management.

(c) Progress state monitoring: Process server asks all agents who are responsible for each engineer about the progress of their activity and collects the current status. It will be displayed and broadcasted to all agents. Every engineer can know the whole progress status in real-time through his/her agent.
Enterprise Server integrates information needed for collaboration support, that is, process repository, resource information and journals information. The contents of those information are as follow.

(a) Process repository: This contains typical process templates which will be extracted out of journal information in previous projects.

(b) Resource information: This contains organization information (departments of managers and engineers), personnel information (name, dept., agent name, current project, skills, etc.) and available facility resource information (tools, machines, data, operation schedule, constraints, etc.).

(c) Journal information: This contains communication logs between agents and dependency data generated from communication logs. After the project finishes, more detailed process may become evident with analyzing journal information. This helps to improve engineering process. Moreover, engineering knowledge can be shared and organized.

Running Image of this prototype system

Here we show how Agents work in our prototype system.

(a) After a process is defined, Agent of a project manager notifies agents of each engineer that activities are assigned. If an engineer accepts, communication link between a project manager and an engineer is established and dependencies or progress state information will be exchanged through this channel.

(b) After activities are assigned, each activity is carried out under agents supports. Agents monitor its engineer and provide appropriate information when needed, i.e. how to notify other engineers if a design change is occurred.

(c) During operation, agents accumulate its engineer’s work logs as an engineering task journal and agents retrieve dependencies out of the journal. For example, an engineer may have to look for an appropriate engineer to notify his design change for the first time. But after that, his/her agent will notify automatically pertinent agents when a design change on the same part occurs.

(d) On the other hand, project manager can manage the progress status based on the information reported by all the agents. It enables him/her to re-order engineers to catch up with pre-defined schedule.

Case Study

The objective of this case study is to examine the validity of Active Process approach and retrieve extract problems. Building the Blocks game is met here. This scenario is intended to be simple but still cover as many as possible features of process description and enactment that can be seen in typical collaborative engineering problems.

Game Scenario

An 8*8 area (Figure 7) needs to be built with four kinds of blocks which have different colors (Red Block: RB, Blue Block: BB, Green Block: GB, Yellow Block: YB). Three constraints are defined:

1. You can't place a block to a certain area if its under area is vacant.
2. Cost and time consumption are attributed to each block locating
   (RB: $1, 4min., BB: $2, 3min., GB: $3, 2min., YB: $4, 1min.)
Cost must be less than $160 and time consumption must be less than 160 min.

3. All adjoining blocks in horizontal must have different color blocks.

The objective of this game is to fill all the area with the cheapest cost and the least time consumption. The number of participants are not limited but there are two kinds of players: Project manager and workers. Project manager assigns tasks to workers. The lowest cost and time consumption as a whole is desired. In this game, all participants are required to collaborate each other to carry out their assigned tasks under those constraints.

**Analogy With Models**

Numbered shapes in Figure 7 comprised of several small squares represent Activities in Active Process model. Small square represents a Work Element. Tasks are assigned to workers in the unit of Activity. For example, some worker may have to deal with Activity 1, 8 and 11 Planned workflow (right-hand in Figure 7) represents activities dependencies which comes from geometry constraints of replacement. Workers can begin his task in the unit of Work Element, that is, he/she can place a block on a new activity area even if current activity has not finished yet.

Following two points should be noticed:

1. Concurrent Activity Execution: Activity 4 may be processed though Activity 1 has not been finished for example.
2. Necessity of negotiation: Adjoining blocks in Activity 1 and 2 must have different colors so that workers who are responsible for each activity must negotiate each other.

Dependencies of Work Elements will be accumulated during operation by agents. How to support effectively workers and managers based on acquired dependency information is the point of this prototype system.

![Diagram of Activities and Planned Workflow](image)

**Figure 7. Activities and Planned Workflow**

**Verification through Prototype System**

Active Process has two aspects: process modeling capability and process manipulation to support engineering collaboration. Prototype system was useful to:

- Visualize and embody modeling concepts.
- Examine process manipulation mechanism (in particular, agents’ motion).

The game was played by several people and we gained following insights.
First, to know information needed for a task, neighboring status in this case, contributes significantly toward the increase of task efficiency. Though the case here seems to be simple, it is important to be able to obtain necessary information timely by agents, without knowing whom should be appropriate participants with regard to the information. Agents asks Process Server an appropriate participants’ agent and establish communication with the agent to gain desired information. This game proved agents to be useful greatly information collecting.

Secondly, dependency relationships accumulated in real-time enable prompt reaction to changes of circumstances and prevention of inconsistency occurrence because of lack of communication.

Thirdly, the possibility of ‘super concurrency’ was suggested. In this game, players could grasp progress status, in the unit of not Activity but WorkElement. They could start their tasks(WorkElement) though the previous activity had not finished yet. Actually the manager could grasp the whole progress status in detail so that he/she could give instructions to players timely. This means that the game management becomes smooth and the total playing time will be shortened.

Lastly, players operation journals were recorded, so that strategy planning for high-scoring became easy.

As shown above, this prototype system proved that process models and agents mechanisms in Active Process was effective in supporting multiple participants’ collaboration and was able to manage the whole process smoothly.

Conclusions

In this paper we discussed collaborative engineering support system using process model and agents approach. To connect process model with product model and enterprise resource data, dependencies of product components are transformed into dependencies of engineering processes and those information are used as workflow data. As for unpredictable dependencies(Work Element dependencies), agents collects dependencies monitoring engineer’s operation. Followings are our conclusions.

• We proposed Active Process model which has capability of learning dependencies of finer granularity task level. It is difficult to do for existing workflow tools.
• Active Process can meet dynamic(unpredefinable) process modeling as well as static(predefinable) process modeling.
• Intelligent-Agents approach are adopted as Active Process mechanism.
• Active Process was verified through prototype implementation using a simple scenario(building the blocks).
• Modeling-prototyping-case study approach was so effective that we could gain many insights.

Future research will be focused on refining Active Process framework in terms of modeling(description capability) and mechanism(agent-approach), through examining scenarios in various design phases in shipbuilding.
Acknowledgements

This research has been carried out under the sponsorship of the Nippon Foundation, and the authors would like to express their cordial thanks to them. They would also like to express their gratitude for the guidance and encouragement received from Prof. Takeo Koyama, University of Tokyo.

References