National Shipbuilding Research Program

Navy Product Data Initiative

Integrated Product Data Environment Specification

Version 1.0 dated 30 June 2008

Navy Program Executive Offices asked NSRP for an industry-wide, open architecture solution to address shipbuilding program cost drivers related to Integrated Product Data Environments.

NSRP responded with an enterprise-wide consensus plan to solve the root issues with as-is IPDEs:

- Current systems are optimized to support lead ship design & construction
- The complexity of executing and managing ship-to-ship configuration variability within the IPDE adds significant costs to technology insertion
- System architectures hinder interoperability for support of multiple hulls and post-delivery support and require costly maintenance to develop interfaces with other shipyard and Navy data systems
- Closed system architectures result in a significant amount of customized proprietary software
- Lack of reconfigurability leads to high costs (time and money) of developing unique Integrated Product Data Environments for each ship class.
- It is often hard to retrieve the data needed after spending lots of money to create and store it

NSRP facilitated experts from shipyards, PEOs, and NAVSEA with the tasking to "specify the requirements for and drive the implementation of product data systems based on an open architecture having suitable functionality and enterprise-wide interoperability to support affordable Navy ships design, construction and service life support."

The result of NPDI consists of two companion deliverables:

1. An **IPDE Specification** that defines a set of high level requirements that will enhance the configuration management and interoperability of ship electronic product data
2. A **Specification Compliance Assessment** document which provides a proposed process for determining specific vendor/tool/system compliance with the specification requirements.

The NSRP and the Navy each made significant investments in the development of product data interoperability standards over the past two decades. The two NPDI deliverables build on these standards by providing a path for future implementation.

The vision of creating, managing and sharing ship product data across the enterprise and throughout a ship's life cycle is now within reach. *The significant cost reduction potential from the implementation of these consensus specifications will, however, require sustained commitment from the Navy, shipbuilders and IPDE vendors.*

FOREWORD

In September 2006, the National Shipbuilding Research Program (NSRP) agreed with the Navy that there was a need to develop an enterprise approach for a more effective Integrated Product Data Environment (IPDE) for ship acquisition and service life programs. The result of this effort, titled the Navy Product Data Initiative, is a proposed Navy IPDE specification defining a set of high level requirements that will enhance both the configuration management and interoperability of ship product model data. The requirements contained in this specification represent the naval shipbuilders' best efforts to build consensus among the various construction shipyards and address a set of common needs that will enhance the creation, management and use of ship product model data throughout a ship's life cycle. It is recognized that achieving the significant benefits that are expected from the implementation of these requirements will require additional Navy, shipbuilder and IPDE vendor analysis and commitment. The work of the NPDI Team and the resulting set of requirements, as documented in the proposed specification, should become the starting point for a continuing effort to improve the utilization of ship product model data across the entire Navy Enterprise.

EXECUTIVE OVERVIEW

The Navy Product Data Initiative (NPDI) was established in response to a NAVSEA request to reduce ship acquisition and maintenance costs associated with computer systems used in the design, manufacture, and life cycle support of shipbuilding programs. The effort was sponsored by the National Shipbuilding Research Program (NSRP) and funded by the Program Executive Offices (PEO) Submarines, Carriers and Ships. NAVSEA tasked NSRP to specify the requirements for and drive the implementation of product data systems based on an open architecture having suitable functionality and enterprise-wide interoperability to support affordable Navy vessel design, construction and service life support. The NPDI developed a proposed Integrated Product Data Environment (IPDE) specification to be invoked on future shipbuilding/service life contracts, and a proposed assessment mechanism to measure compliance, effectiveness and provide guidance for a shipbuilder's IPDE implementation.

An integral component of all shipbuilding programs is the shipbuilder's IPDE. This collection of business processes, computer systems, and associated services, which house the product model data, enable people to work in concert towards common business goals throughout the life cycle of a product. The establishment of new shipbuilding programs has frequently required shipbuilder IPDEs to be extensively upgraded, or at times replaced, to leverage technology advances enabling improved shipbuilding processes. A common drawback of these environments is their proprietary (closed) architecture inhibiting the flexibility to reconfigure and adapt to changing program requirements without significant, costly customization. Despite the investment by both the Navy and the shipyards in building and maintaining these systems, there are still many shortfalls in the ability of existing IPDEs to efficiently manage changes and share product information with Navy activities or other shipyards.

Representatives from Northrop Grumman Shipbuilding, Electric Boat, NASSCO, Bath Iron Works and NAVSEA drafted this proposed specification based on lessons learned from previous submarine, carrier, and ships programs. It represents hundreds of man-years of experience in the design, development and deployment of IPDEs. This specification is intended to help implementers: identify common IPDE capabilities based on best industry practices, improve configuration management processes, build similar IPDEs to minimize software design, acquisition, and maintenance, enable IPDE interoperability between ship HM&E collaborators (designers, builders, equipment suppliers, Navy laboratories) and mission/weapon systems suppliers, and facilitate digital delivery of ship engineering and logistics data to customer's IPDEs for eventual life cycle support. It will provide the shipyards the criteria which are needed to efficiently manage product data throughout a ship life cycle and a common statement of need for commercial software vendors to enhance their products. The ultimate goal being to enable shipbuilders and operators to minimize total ownership costs. We anticipate the Navy and shipbuilders will determine how to implement this information in future ship acquisition projects (RFP language, specifications, contract language, instructions, guidance documents, etc) to achieve the ultimate goal of reduced total ownership costs for all stakeholders.

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

The design, construction and life cycle support of Navy ships is an extremely challenging, technically complex endeavor which can span 30 to 40 years. The design and construction of a class of ships can span a decade all by itself. The information required to support this effort is voluminous and constantly changing. History indicates that the advent of computer technology to the shipbuilding process has been a double-edged sword bringing advances in process automation as well as reductions in design cycle time, design errors found during construction, the number of procured parts, and the number of Navy drawing approvals. It has eliminated the need for physical mock-ups and many other labor-intensive design and development practices albeit at the price of increased complexity in information management. The ability for disparate computer systems to effectively communicate and interoperate with one another has also proven to be a complicated and costly undertaking. Managing change and obtaining efficient information interoperability have risen to become major cost drivers in the shipbuilding industry.

An Integrated Product Data Environment (IPDE) is a collection of business processes, computer systems, and associated services, which house the product model data, and enable people to work in concert towards common business goals throughout the life cycle of a product. Designing and maintaining a ship using an IPDE can add significant efficiencies to the process, but if not done wisely, it can also add unacceptable cost and risk to a shipbuilding Program.

During the course of large shipbuilding programs, shipyards are required to collaborate with partnering yards, the customer, suppliers, the fleet, and sub-contractors. Disparate IPDEs from each organizational entity need to interoperate at the system and data level, enabling effective and efficient collaboration.

Using lessons learned from several major US shipbuilders, this specification describes the capabilities, functions, and architecture that an IPDE must support in order to enable affordable Navy ship design, construction and service life support.

1.1 Purpose

The purpose of the Navy Product Data Initiative is to define a common set of capabilities which will help the shipbuilder and the Navy reduces ship acquisition and maintenance costs through their life cycle. The purpose of the specification is to address shortfalls in current product development environments, especially in the areas of configuration and change management, and the interoperability of systems within an IPDE and between disparate IPDEs; the goal being to drive down the cost of change and increase the reuse of ship product data over the entire ship's life cycle. The specification defines a set of requirements for an effective IPDE, focusing primarily on the traditional product data management capabilities and interfaces with CAD/CAM/CAE, ERP and catalog systems used for the design, construction and in-service support of Navy ships and submarines. The scope of the specification covers aspects of the product development environment used to create, manage and disclose ship product information throughout the life of the ship. The requirements contained in the specification, in all cases, are intended to enhance the current capabilities to manage product configuration and enable efficient product change.

The specification will address the need to enable system interoperability within a company's IPDE or across disparate IPDEs in a collaborative nature. There are two aspects to interoperability; Open Systems and Data Exchange. Open Systems addresses the need for an open IPDE architecture, enabling robust system integration, and the ability to perform IPDE system component upgrades without major cost and disruption. Data Exchange deals first with the scope and content of the data to be transferred and second, the methods of transfer. It is intended that invocation of this specification will, by extension, invoke the scope and content of the data to be available for integration/interfaces defined in the Ship Common Information Model (SCIM) Ref. (2).

Current IPDE tools do not support all of the requirements defined in this specification and will need to be enhanced to meet the objective of efficient change management. For example, the authoring tools, e.g. CAD, and other applications must be capable of using data, (create, export, retrieve), at the part level to obtain maximum benefit from the IPDE.

1.2 Scope of the Specification

1.2.1 IPDE Specification Boundaries

This specification focuses on requirements for the core capabilities of an Integrated Product Data Environment along with requirements for Interoperability and Data Exchange between the major applications that make up an IPDE. Figure 1-1 illustrates the boundary of the IPDE specification vs. an overall IPDE. This notional model shows the applications supporting the major shipbuilding and support functions as vertical columns. These vertical applications are different for each implementation of an IPDE; some IPDE's will not have all the applications or will include additional applications. The IPDE core capabilities support the vertical applications and provide management and control access to the product model data.

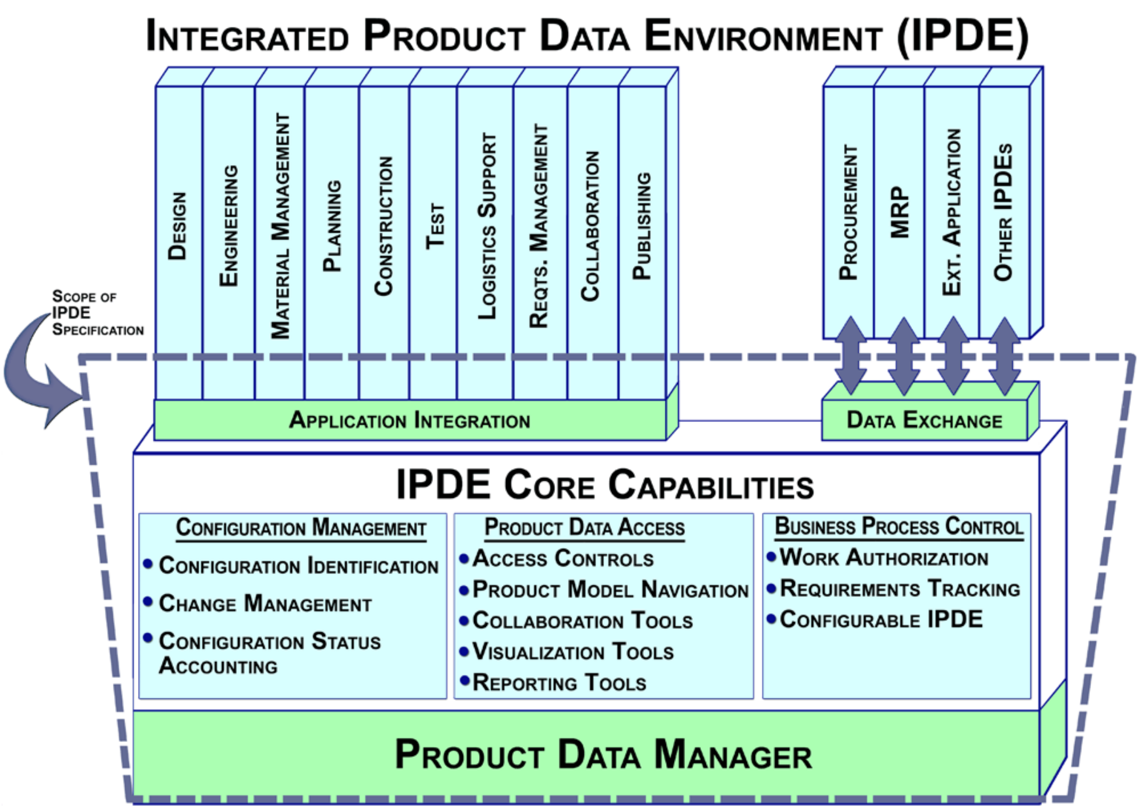


Figure 1-1 *IPDE Specification Boundaries*

The IPDE specification defines the requirements for the core capabilities that support the interaction between the product data manager, the vertical application domains and the transfer of information to external systems. Although the IPDE specification includes toolset requirements for cross cutting applications like collaboration, visualization and reporting it does NOT specify the requirements for the functions in the vertical applications. The specification also does NOT address several other IPDE requirements, including:

- Use of a standard parts catalog - only the ability to interface with a Common Parts Catalog is required. It is up to the program to determine the catalog to be used.
- Network security, single sign-on or the ability to handle various DOD security levels
- Program Management metrics and reporting.

1.2.2 Shipbuilding Needs and the IPDE

As mentioned above, the life cycle of a class of ships can be 30 to 40 years in which case the IPDE must be designed to support the product model throughout the life cycle. Section 2 of the IPDE specification documents specific shipbuilding and support business process needs as they relate to management and access control to product data. These needs are required for all shipbuilders, but will be fulfilled differently according to business reasons, legacy tools, existing processes, etc. These needs will also be the basis for test scripts to test specific technical requirements. These needs provide the context for the IPDE requirements. These needs are organized according to the different shipbuilding activities:

- Design/Engineering
- Material Management
- Planning
- Construction and Test
- Integrated Logistics Support.

In addition, Section 2 describes cross cutting needs that exist:

- Requirements Management
- Collaboration
- Reporting and Publishing.

Figure 1-2 illustrates the IPDE's role in supporting the product data over the life cycle of a class of ships and the transfer of that data from one IPDE to another. The figure shows the progression of life cycle phases, from Concept Definition through Construction and Test, In-Service Support (including Shipalts), and finally Decommissioning. During the design and construction phases, there is typically a set of PDM and application tools that capture and maintain the product data in a native format. However at any time during those phases, product data may need to be transferred to external systems or archived. As shown in the figure, this transfer would comply with the Ship Common Information Model (SCIM). Once a ship is in service, product data may be transferred to other contractor's IPDEs or other Navy Systems.

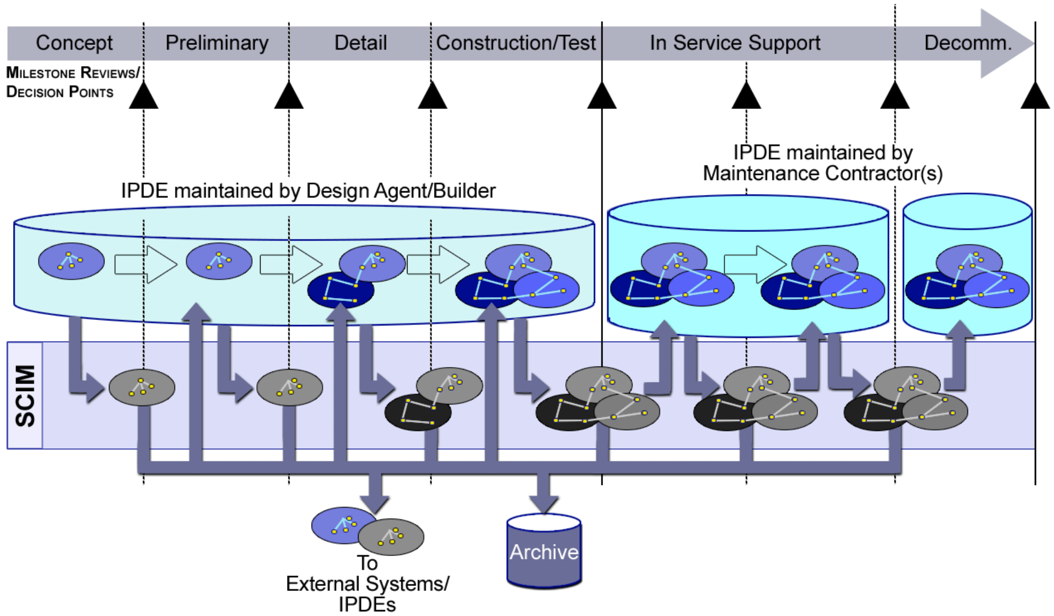


Figure 1-2 *IPDE Product Data needs to transition throughout the life cycle*

1.2.3 Core IPDE Capabilities

The core IPDE capabilities shown in Figure 1-3 represent a set of functionality that need to be addressed by an IPDE implementation to effectively manage ship product model data over the life cycle of a Navy vessel. The figure shows a notional relationship between these core capabilities. Some of this functionality may already be incorporated into existing commercial Product Data Managers, but it is the integration of these capabilities that creates the most cost-effective functionality for the IPDE. Specific requirements are described in Section 3.

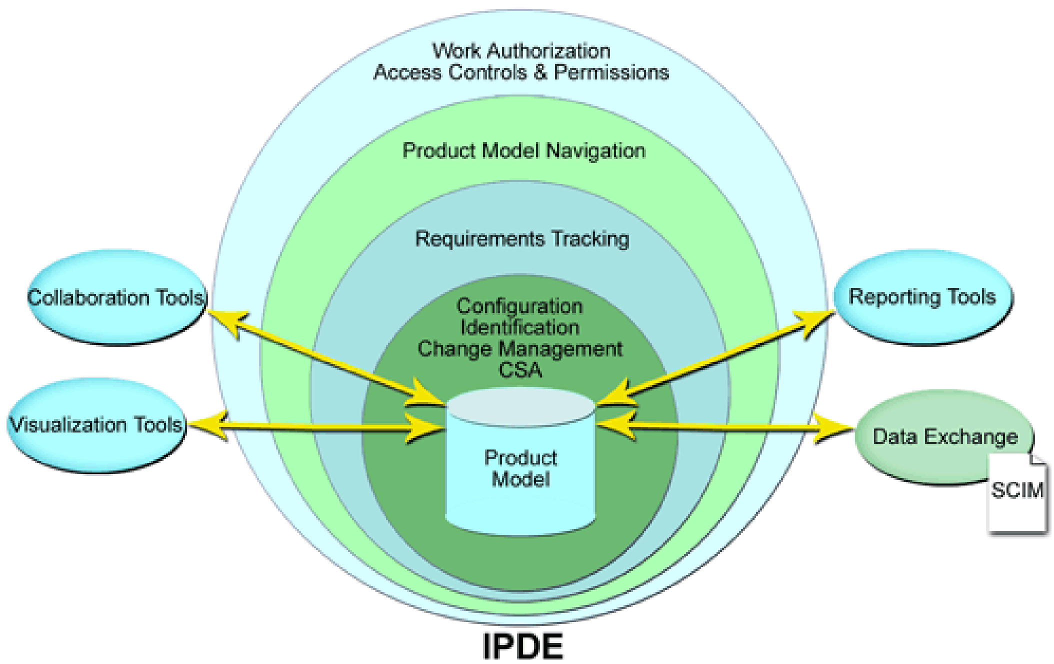


Figure 1-3 *IPDE Core and Cross Cutting Capabilities*

Surrounding the Product Data is a set of capabilities that maintain, control, and protect the data from corruption or misuse. At the heart of the system is Configuration Management which is designed to manage all the shipbuilding product data elements. This capability includes **Configuration Identification**, **Change Management** and **Configuration Status Accounting**. **Requirements Tracking** maintains linkages to program and product requirements as changes are identified and implemented. **Product Model Navigation** provides the tools to find and select the parts of the Product Model relevant to a particular user's needs. **Access Control and Permissions** protect the IPDE from unauthorized user activity and **Work Authorization** integrates with the Shipyard business processes to ensure the proper accounting for changes to the Product Model.

Once all these functions are in place, the Product Data can be used by a variety of tools that support **Collaboration**, **Visualization**, **Reporting** and **Data Exchange**. Behind all these core capabilities is a **Configurable IPDE** designed to be easily reconfigurable to accommodate specific program requirements.

1.2.4 Interoperability and Open Systems Compliance

Section 4 of this document addresses interoperability by defining requirements related to data exchange and the use of the **Ship Common Information Model** (**SCIM**). A significant component of interoperability of a system is its ability to provide its data in a standardized format using a well-defined interface. The SCIM has been developed in parallel with this specification and is included via reference throughout this document. It defines the minimum data elements necessary to support interoperability and interfaces between various IPDE systems and components of a given IPDE. It also serves as a common Data Dictionary. Section 4 will describe the role of the SCIM and the constraints of its use within an IPDE implementation.

An IPDE implementation will most likely consist of a collection of integrated COTS software applications along with purpose-built, custom functionality to suit the specific needs imposed by this document along with its intended environment. Because of the wide range of existing systems that can be employed and the disparate software configurations that define them it would be impractical for this specification to mandate or assume any specific architecture. However, ensuring that any given IPDE meets the goals of the NPDI program (specifically in regards to Interoperability and Open Architecture) requires that each implementation conforms to the principles defined by Open Architecture guidelines to the fullest extent possible.

In the context of shipbuilding, an IPDE implementation will be compliant to the tenets of an open architecture and provide for interoperability if it:

- Is constructed from an extensible, scalable architecture using open standards where applicable
- Provides accessibility to its content
- Enables the efficient exchange of product model information using the SCIM
- Is described with sufficient, comprehensive, coherent, and consistent interface specifications

Using the Navy *Open Architecture Computing Environment – Design Guide* (OACE-DG) as a reference, section 4.3 identifies each of the design elements for a software system that are relevant to shipbuilding IPDEs. Note that the information provided in this document with respect to software architectures is meant to be interpreted as guidance only. Specific architectural requirements will be left to the requirements defined for each individual IPDE implementation.

1.3 Compliance Assessment of IPDE Implementations

It is recommended that NPDI compliance assessment be conducted by an independent organization. Sections 3 and 4 of this specification delineate the IPDE requirements that are the basis for this assessment. However, it is recognized that requirements will be phased in as technology evolves to enable the capabilities in individual shipyard implementations.

1.4 Organization of the Specification

Section 1 – INTRODUCTION

This section provides background on the use of IPDEs in shipbuilding, the problems that exist with current implementations, and defines the scope of the specification.

Section 2 – SHIPBUILDING NEEDS

This section describes the shipbuilding functions which require core IPDE capabilities in order to execute effectively. The intent of this section is to provide the context for the requirements in Sections 3 & 4.

Section 3 – CORE IPDE CAPABILITY REQUIREMENTS

This section contains the enforceable core IPDE requirements. The core IPDE requirements describe what the IPDE has to do to meet the shipbuilding needs described in section 2. IPDE requirements are identified by an "R" number and are in ***bold italics***.

Section 4 - INTEROPERABILITY REQUIREMENTS

This section contains the enforceable requirements for data exchange and the use of the SCIM along with architectural guidelines to support interoperability within an IPDE as they relate to open system principles. IPDE requirements are identified by an "R" number and are in ***bold italics***.

Appendix A - CONFIGURATION MANAGEMENT CASE STUDY

This appendix is a case study that compares two configuration management strategies. It shows the advantages of using a part-based strategy verses a document-based strategy for reducing data duplication and data management complexity.

1.5 References

1. NPDI Compliance Assessment Plan (Version 1.0, 30 June 2008)
2. Ship Common Information Model (SCIM)
3. Open Architecture (OA) Computing Environment Design Guidance Version 1.0 23 August 2004
4. Open Architecture (OA) Computing Environment Technologies and Standards Version 1.0 23 August 2004
5. ANSI/EIA-649-A, National Consensus Standard for Configuration Management, 28 October 2004
6. NAVSEA Technical Specification 9090-700, Ship Configuration and Logistics Support Information System (SCLSIS)
7. MIL-HDBK-61A, Configuration Management Guidance, 7 February 2001

1.6 Definitions

Applicability – the authorization of a part to a class of ships.

Configuration Item – a configuration item, or CI, is a unit of configuration that can be individually managed and versioned. Configuration items form the basis of configuration management. Specific types of configuration items are defined in Section 3.

Hull Effectivity – the assignment of an occurrence of a configuration item to a specific hull or series of hulls.

Information Interoperability – the ability to effectively share and use information across system, application and organization boundaries.

Integrated Product Data Environment (IPDE) – a collection of business processes, information systems, and associated services, which house the product model data, and enable people to work in concert towards common business goals throughout the life cycle of a product.

Integrated Shipbuilding Environment (ISE) – an NSRP-funded program, managed by a consortium of US shipbuilders and technologists, whose goal is to define and document the information requirements of US Naval shipbuilding for the purpose of information interoperability among and within shipbuilding Integrated Product Data Environments.

Navy Product Data Initiative (NPDI) – a NAVSEA funded effort to specify the requirements for and drive the implementation of product development systems based on an open architecture having suitable functionality and enterprise-wide interoperability to support affordable Navy ships design, construction and service life support.

Open Architecture – an architecture that employs open standards for key interfaces within a system.

Product Data Management (PDM) – a category of computer software used to control data related to products. PDM creates and manages relations between sets of data that define a product, and store those relationships in a database. It is an important tool in product life cycle management.

Product Life cycle Management (PLM) – the processes and associated software applications for managing the entire life cycle of a product from its conception, through design and manufacture, to service and disposal.

Product Model – the entire product information database(s) which describes the product completely and unambiguously.

Product Data – information about the constituent components of a product, such as a ship, including its properties, representation, relationships, with other product data, support documents and information (e.g. logistics and support data, analyses, reports), relationships to output products (e.g. drawings, technical manuals, EBOM).

Product Structure – a relationship of configuration items which often shows the hierarchical decomposition of a ship**.** There will be multiple product structures within the ship e.g. logical design, physical design, manufacturing assembly, and logistics structure.

Ship Common Information Model (SCIM) – a specification describing a neutral data model used for the capture and exchange of ship product model information to enable effective interoperability among US Navy sponsored shipbuilding IPDEs.

Vertical Applications – software applications that utilize the IPDE product data; examples of these would be CAD, CAM, Analysis applications, Requirements Management tools and Planning and Scheduling Tools

Work Breakdown Structure (WBS) – a product-oriented ‘family tree’ of elements that organizes and defines the hardware, software, services and related work tasks required to develop a ship.

1.7 Acronyms

BOM - Bill of Material

CI - Configuration Item

CM - Configuration Management

CPC - Common Parts Catalog

DfP – Design for Production

EBOM - Engineering Bill of Material

EFD - Equipment Functional Descriptions

EPL – Engineering Parts List

ERP - Enterprise Resource Planning

FSI – Functionally Significant Item

HSC - Hierarchal Structural Codes

IPDE - Integrated Product Data Environment

IPT - Integrated Product Team

LCB - Logistics Configuration Baseline

MBOM - Manufacturing Bill Of Material

NSRP - National Shipbuilding Research Program

OSJTF - Open Systems Joint Task Force

RMA - Reliability Maintainability Availability

SCIM - Ship Common Information Model

SME - Subject Matter Expert

VFI - Vendor Furnished Information

WBS - Work Breakdown Structure

WIP - Work In Process

2.0 SHIPBUILDING NEEDS

This section describes the functions within the domains that the IPDE needs to support. The intent of this section is to provide the context for the requirements in Section 3 and 4. Each of these sub-sections describes the needs that the IPDE is expected to support. These needs are required for all shipbuilders, but will be fulfilled differently according to business reasons, legacy tools, existing processes, etc. Since these domains represent project-specific/shipyard-specific implementations, they are not part of the enforceable specification.

Each of the following sections contains a table describing specific IPDE needs for that section's functional area. The table maps the functional need to the Section 3 Core Capability requirements and Section 4 Interoperability requirements. A subsection number in a column indicates that requirements in this area are applicable to the described need.

2.1 Design / Engineering Activities

Conceptual Phase

The conceptual phase is the time when design and engineering concepts are evaluated, compared to the requirements, revised, reevaluated, and so on until convergence to one or more satisfactory concepts is achieved. These concepts may develop from new ideas or extractions of previous designs. The models and analytical results created during this time may be linked to ship specific configurations or may stand on their own to be used in future configurations.

Output products from this phase include system descriptions, models, analysis results, arrangements, tradeoffs, notional build strategies and shop identification.

To support the concept phase multiple class and ship specific configurations are modeled and analyzed. The models, studies, and analytical results which are created during this time are stored so that they may be reused in the same or a disparate IPDE.

Design models are shared with engineering for analysis and ergonomic studies. This allows for integrated domain (e.g. electrical, piping, HVAC,) design, analysis and simulation capabilities.

Models and analysis results developed during concept design are visualized at the desktop or in team visualization rooms.

Preliminary Phase

The preliminary phase begins by translating the requirements specifications as derived from the concept phase into the system design. During preliminary design various design concepts are subjected to more rigorous analysis and evaluation, including Design for Production (DfP), in order to define and validate the design that best meets the requirements. This phase provides a detailed plan of system functions and descriptions, diagrams, preliminary arrangements, a revised build sequence and strategies.

During the preliminary phase system descriptions are developed and reconciled against the ship specification. The requirements defined in the ship specification are linked to specific configuration items. Diagram components are developed and applied to specific hulls and 2D diagrams and 3D arrangements are reconciled to ensure all parts have been consumed.

Throughout this time models are transferred with the appropriate level of detail (system, space allocations, long lead time material etc.) between design and engineering analysis environments. In addition, integrated multi-discipline analysis and simulations are conducted with their outputs being related back to the product model.

Virtual ship models are developed and ergonomic and anthropomorphic studies are conducted.

Detail Phase

The detail phase takes the preliminary arrangement agreed upon and develops it with more detail. This phase involves working out all major design components. It covers all areas of the systems, how they will be built and what is needed to build them. Output products from this phase include detailed arrangements, 2D & 3D models, drawings, build sequence, manufacturing data, analysis results and test procedures.

During the detail phase multiple models and analysis results are created and stored. 3D physical design models are integrated with 2D diagram logic to allow the 3D model to consume system components and supplemental information. The detailed models are updated when the arrangement model has been revised. Analysis and simulations are conducted and related back to the design data.

Models, analytical results, ship specific configurations, technical data (notes, EPL, JSI, etc.) and technical instructions (e.g. work packages) are applied to a specific hull.

Virtual walkthroughs of the ship are performed in team design review settings. As the ship's design develops product model related comments from shipbuilding partners and customers are documented, tracked, and reconciled.

Construction/Test Phase

During the ship's construction, as systems are installed and tested, any problems discovered with the design are relayed back to the design agent for resolution. The resolution process may include remodeling, analyzing, testing, and other activities that took place during the previous design/engineering phases.

Problem identification notices are tracked and resolved. Stop work orders and design improvement notices are issued. Testing is performed on the systems and in the case of a design-related failure, a failure notification form is sent back to design and engineering.

In Service Support Phase

After the ship has been commissioned, maintenance or alterations may occur on the vessel. When a change is called for, a liaison action request or a condition report is issued. New models are developed and analysis and simulations are performed to support the alterations. The alterations must be able to be compared to the original design, the as-built ship or to the as-maintained ship configuration.

2.1 Design/Engineering Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must provide the ability to support multiple programs and contracts concurrently.	3.2.1		3.4.3	
2) The IPDE must provide the ability to manage multiple ship alterations and compare ship alterations to the original design.	3.2.1 3.2.2	3.3.2 3.3.5		

2.1 Design/Engineering Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
3) The IPDE must provide the ability to manage and track changes from the as-designed to the as-built to the as-maintained ship configurations.	3.2.1 3.2.2 3.2.3	3.3.2 3.3.4 3.3.5		
4) The IPDE must provide the capability to manage system diagrams, detailed drawings, and their associated components and component attributes. The IPDE must produce reports listing all the parts for a particular drawing with associated part information.	3.2.1	3.3.2 3.3.4 3.3.5		
5) The IPDE must provide the ability to provide bi-directional links between technical data, such as notes, EPLs, work packages and other product model data.	3.2.1	3.3.2 3.3.5		
6) The IPDE must provide the ability to create bi-directional links between specification requirements and configuration items.	3.2.1	3.3.2 3.3.4	3.4.2	
7) The IPDE must provide the capability to associate technical data to component parts or systems. Examples of technical data include design documents, interface documents, work packages, and test procedures and test results.	3.2.1 3.2.2	3.3.4 3.3.2 3.3.5		
8) The IPDE must provide notification to update derived design and manufacturing data when the 3D arrangement is revised	3.2.2			
9) The IPDE must provide the capability to manage engineering concept studies, engineering models, and engineering analysis results.	3.2.1	3.3.2		
10) The IPDE must provide the capability to track problem identification notices, stop work orders and design improvement notices. These should be linked to all product model data involved in their resolution, including but not limited to models, analysis results and the BOM.	3.2.2 3.2.3	3.3.5		
11) The IPDE must provide revision comparison tools to support change management.	3.2.1 3.2.2 3.2.3	3.3.2 3.3.5		
12) The IPDE must provide the capability to manage the Ship System Description and associate it with system requirements and design. This allows changes in the requirements, system description, or design to be reconciled with each other in order to maintain consistency among the system technical data.	3.2.1 3.2.2	3.3.2 3.3.5	3.4.2	

2.1 Design/Engineering Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
13) The IPDE must provide the ability to allow integrated multi-discipline analysis and simulations, manage the outputs of each and relate them back to the product model without having to duplicate design information or re-create design models.	3.2.1	3.3.3 3.3.2 3.3.5		4.2.1
14) The IPDE must be able to integrate 3D physical design models with 2D diagram logic allowing the 3D models to consume system components and supplemental information. It must have the ability to verify that all components on the diagrams are accounted for in the 3D model.	3.2.1	3.3.4		
15) The IPDE must provide the ability to allow data owners to easily make changes to data which has no requirement for re-approval.	3.2.1	3.3.1	3.4.1	
16) The IPDE must provide the capability to perform ergonomic and human factors analysis using the ship 3D product model.		3.3.4		
17) The IPDE must provide the capability to exchange design models and engineering analyses between systems to support activities such as concurrent design and independent analysis.	3.2.1 3.2.3	3.3.3 3.3.5		4.2.1 4.2.2 4.2.5
18) The IPDE must provide the capability to manage and reconcile product model related comments from shipbuilding partners and customers.	3.2.2	3.3.3		
19) The IPDE must provide the capability to extract data such as weight estimates area/volume, power consumption, heat rejection, etc. from the product model.	3.2.1	3.3.4 3.3.5		

Table 2-1 Shipbuilding IPDE needs for Design/Engineering

2.2 *Material Management Activities*

Conceptual Phase

The material management process begins by defining the material requirements from the ship specification.

After the baseline is defined applicable mil-spec, commercial specification, and contractual specification requirements are identified with their appropriate revisions. With these specifications defined a standard parts library is created. The standard parts library consists of legacy parts and parts to be designed. Using this library as a guide, design and engineering personnel begin to identify the requirements for long lead time material, thus creating the initial Bill of Material (BOM).

At this time vendor and mil-spec graphic and catalog data begins to be imported, this information will later be linked to part information.

Preliminary Phase

As the BOM is developed changes to it in the form of trade-offs or what-if part information must be recorded and tracked in order for the required part to be procured. Part information may be exchanged with outside sources such as other shipyards and suppliers. This part information is released on a part by part basis as each part is elevated to a particular status.

Detail Phase

As the BOM becomes further detailed and finalized the quality attributes (Subsafe, hull integrity, etc.) and the quantity of each of these attributes, for the occurrence in which it is used, are added.

Part occurrence data is also added to the BOM and is linked to elements in the 3D model. This occurrence information may be used for reporting and visualization purposes along with graphical information in the product model.

When a design has reached a certain state of approval material is released for procurement. A reconciliation process manages items forecasted to be bought or purchased versus the actual material that was released upon the drawings approval.

Construction / Test Phase

Throughout the material management process manufacturing and design views of the BOM, along with as-planned and as-built revisions, are maintained.

In Service Support Phase

After a ship has been commissioned warranty and guarantee information against material and any repairs related to them must be tracked. On-board and in-shipyard repair parts must also be tracked along with the delivery availability of these items.

The as-built, as-altered or modernized BOM views may be compared at any point during the ships' life. This data must be able to be easily exchanged with other shipyards or government agencies if alterations are needed to the vessel.

2.2 Material Management Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must provide the ability to support multiple material flows, allowing a shipyard to be involved with multiple programs and multiple customers concurrently.	3.2.1	3.3.1 3.3.3	3.4.3	
2) The IPDE must be able to manage multiple bills of material.	3.2.1	3.3.4		
3) The IPDE must provide the ability to store occurrence data for each part.	3.2.1	3.3.4 3.3.5		
4) The IPDE must provide the ability to link testing data, test requirements, test schedules and test results to the BOM.	3.2.1	3.3.3 3.3.4 3.3.5		

2.2 Material Management Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
5) The IPDE must be capable of integrating construction and installation schedules into the BOM.	3.2.1	3.3.4 3.3.5		
6) The IPDE must provide the ability to record and track warranty and guarantee information against materials and any repairs related to them.	3.2.1	3.3.4 3.3.5	3.4.2	
7) The IPDE must provide the ability to track both on-board and in-shipyard repair parts, along with the delivery availability of these items.		3.3.4		
8) The IPDE must provide the capability to manage part information and material data including part numbers, part attributes, specifications, certifications, and test requirements.	3.2.1	3.3.2	3.4.2	
9) The IPDE must provide a parts catalog system used across all disciplines in order to maintain standardization.	3.2.1			
10) The IPDE must provide the capability to identify, order, and track long lead material.	3.2.1			
11) The IPDE must provide the ability to release material for procurement when a design has reached a certain state of approval.	3.2.2			
12) The IPDE must provide the capability to identify and release procurement requirements at the part level.	3.2.1 3.2.2			
13) The IPDE must provide a reconciliation process to manage items forecasted to be purchased versus the actual material that was released upon design approval.	3.2.1 32.2			
14) The IPDE must have the ability to manage standard part data and be able to electronically create an engineering parts list (EPL).		3.3.4 3.3.5		
15) The IPDE must provide the ability to identify functionally significant items and maintenance-worthy parts and be able to report where they are used at any given time during the life of a ship.		3.3.4		4.2.1
16) The IPDE must provide the ability to import and configuration manage vendor and mil-spec graphic and catalog data. It must be able to provide bi-directional links from that information to the part.	3.2.1		3.4.2	4.2.1
17) The IPDE must provide the ability to retrieve and visualize part occurrence data along with graphical information in the product model.	3.2.1	3.3.4 3.3.5		

2.2 Material Management Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
18) The IPDE must provide the capability to associate the shipyard part system with the Navy stock system.				4.2.1
19) The IPDE must provide the capability to exchange part and material information with outside systems.		3.3.1 3.3.3 3.3.5		4.2.1 4.2.2 4.2.5

Table 2-2 Shipbuilding IPDE needs for Material Management

2.3 *Planning Activities*

Conceptual Phase

The conceptual planning phase is the time when design/engineering concepts are evaluated relative to build strategy and construction facilities constraints, compared to alternatives until convergence to one or more satisfactory concepts is achieved. These concepts may develop from new ideas or extractions of previous designs. Output products from this planning phase include build strategy alternatives, analysis results, arrangements and tradeoffs.

To support the planning concept phase the 3D models, studies, and facilities analysis which result during this time must be stored in a manner that allows them to be easily extracted for reuse in the same or disparate IPDEs.

Revision comparison tools must be able to support change management and enable integrated multi-discipline design, analysis, planning and simulation capabilities. Build strategy models must be able to be easily shared with design/engineering and construction for review. Product structure alternatives and analytical results may be linked to ship specific configurations or may stand on their own to be used in future configurations.

Preliminary Phase

The preliminary planning phase translates the design/build alternatives as derived from the concept phase into the single selected build strategy. During preliminary design, concepts from the conceptual design phase are subjected to more rigorous analysis and evaluation in order to define and validate the design product structure, at a part level, meets manufacturing requirements for applicable construction facilities. This phase provides further detail to a work package plan of preliminary design/build arrangements.

Detail Phase

The detail planning phase takes the work break down structure agreed upon in the preliminary phase and develops it with more detail. This phase reviews of all major design components; it covers all areas of the systems, how they will be built and what is needed to build them and decisions on make-buy material sourcing.

2.3 Planning Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must provide the capability to manage multiple ship configurations to support the preparation of build strategies based on available facilities.	3.2.1	3.3.2 3.3.5	3.4.1	
2) The IPDE must provide the capability to maintain a Bill of Material (BOM).	3.2.1	3.3.4		
3) The IPDE must provide the capability for multiple views of the BOM such as the Engineering Bill of Material (EBOM) and Manufacturing Bill of Material (MBOM).	3.2.1	3.3.2 3.34	3.4.1	
4) The IPDE must be able to maintain as-planned versus as-built revisions of the BOM	3.2.1	3.3.2 3.3.4 3.3.5	3.4.1	
5) The IPDE must have the ability to reconcile the EBOM and the MBOM views to verify all parts have been consumed.	3.2.1 3.2.3	3.3.2		
6) The IPDE must provide the capability to indicate engineering data above manufacturing product structure, such as zones, systems, etc. The capability should include the ability to simulate a large product structure using attributes loaded at the part level.	3.2.1	3.3.2 3.3.4		
7) The IPDE must be able to link the fabrication and assembly process to the product model.	3.2.1	3.3.5		
8) The IPDE must have the ability to support work sequence modifications. These include adjusting manufacturing schedule, updating routings, modifying the BOM and updating required shop floor applications.	3.2.1 3.2.3	3.3.4	3.4.1	
9) The IPDE must provide the capability to associate manufacturing work packages with engineering data.	3.2.1	3.3.4		
10) The IPDE must provide the ability to assign Assembly and Fabrication Work Centers to construction products.	3.2.2	3.3.3	3.4.1	
11) The IPDE must provide the capability to create and manage manufacturing aids and provide notification of changes to these products to ensure that changes are incorporated into construction work packages.	3.2.1 3.2.2 3.2.3			
12) The IPDE must provide the ability to create and manage Fabrication data, including estimated hours, based on assigned Work Center (documents/machine files).	3.2.1 3.2.3	3.3.2 3.3.4 3.2.5		

2.3 Planning Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
13) The IPDE must be able to link technical data (notes, EPL, JSI, paint requirements, lubrication, etc.) and technical instructions (e.g. work packages) to the part. Results of changes to planning decisions/work packages and accuracy control requirements must be able to be applied automatically to published/released documents.	3.2.1 3.2.2	3.3.2 3.3.4 3.3.5	3.4.2	
14) The IPDE must have the ability to rapidly change the design and product structure to support a reschedule of a single or multiple ship work sequences.	3.2.1			
15) The IPDE must provide the capability to perform planning analysis of alternatives using the design model, or information derived directly from the design model, without having to duplicate design information or re-create design models. This analysis supports the preparation of storyboards or narratives, textual or graphic descriptions, to go along with the build strategy.	3.2.1	3.3.2 3.3.5		
16) The IPDE must have the ability to rapidly model and analyze different ship configurations to prepare a build strategy based on available facilities, produce weight estimates, and evaluate from a capacity management view of facilities.	3.2.1	3.3.2 3.3.5	3.4.1	
17) The IPDE must have the ability to link planning decisions and models to specific ship configurations.	3.2.1	3.3.2		
18) The IPDE must allow for product structure and work breakdown alternatives and analysis results fashioned during this phase to be visualized alone or as part of one or more ship arrangements.	3.2.1	3.3.2 3.3.3 3.3.5	3.4.1	
19) The IPDE must have the ability to electronically exchange work package EBOM definition with multiple shipyards.		3.3.3		4.2.1 4.2.2 4.2.5
20) The IPDE must provide design schedule impact analysis for changes to product structure, construction sequence or schedule		3.3.4 3.3.3	3.4.1	

Table 2-3 Shipbuilding IPDE needs for Planning

2.4 *Construction/Test Activities*

The construction/test process includes the manufacture, assembly, installation, inspection and testing of parts, components and other material to form the final product. Using manufacturing data defined in earlier phases (drawings, work packages, and material information) the construction and test processes involve a shipyard's labor and facility resources to execute a planned systematic approach to assembly of these elements into the final end product that is delivered to the customer.

2.4 Construction/Test Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must provide the ability to store test attributes (e.g. test pressure) and link them to the appropriate items in the BOM.	3.2.1 3.2.2 3.2.3	3.3.2 3.3.4		
2) The IPDE must provide the ability to capture preliminary test structures and testable sections.	3.2.1 3.2.2	3.3.2 3.3.4		
3) The IPDE must provide the capability to maintain the as-built baseline and provide changes to other functional areas such as Logistics and Engineering for incorporation into their respective products.	3.2.1 3.2.3	3.3.2 3.3.4 3.3.5		
4) The IPDE must provide the capability to maintain and display certification status and completion for the system/component/ compartment as it is completed, tested, and installed.	3.2.1 3.2.2 3.2.3	3.3.4	3.4.1	
5) The IPDE must provide the ability to provide bi-directional links between models, the BOM, notes, routing, welding requirements etc. to create manufacturing aids and construction work packages.	3.2.1 3.2.3	3.3.2 3.3.5	3.4.1	
6) The IPDE must provide the ability to manage the closeout of test forms including tracking and reporting of metrics.	3.2.1 3.3.3 3.3.3	3.3.5		
7) The IPDE must provide the capability to integrate the installation, test, and construction schedules with the product model in order to view a scheduled build sequence by assembly and zone/compartment progress/completion.	3.2.1 3.2.3	3.3.4 3.3.5		
8) The IPDE must provide the ability to deliver parts-based technical data and instructions on demand to the shop floor.	3.2.1	3.3.2 3.3.3 3.3.4		
9) The IPDE must provide the ability to display zones and compartments with schedule impact.	3.2.1	3.3.2 3.3.4	3.4.1	

2.4 Construction/Test Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
10) The IPDE must provide the ability to electronically exchange work package and manufacturing information with multiple shipyards.	3.2.3	3.3.3 3.3.5		4.2.1 4.2.2 4.2.5

Table 2-4 Shipbuilding IPDE needs for Construction/Test

2.5 Integrated Logistics Support Activities

Common Logistics Activities

Integrated Logistics Support manages the as-designed, as-built, and as-maintained configuration of the ship's logistic products. It ensures that the equipment on each ship can be adequately maintained and repaired throughout the life of the vessel.

Logistics output products include Logistics Configuration Baselines, training materials, maintenance material, Reliability, Maintainability, Availability (RMA) analysis, Configuration Status Accounting and Supply Support analysis.

Common logistics activities combine the functional and business processes required to ensure the correct logistic support of the ship's components. These common services include collaboration capabilities, the use of a common product model, managing changes, visualization, reporting, etc. Logistics activities such as configuration changes, provisioning, maintenance planning, training and technical data are managed during the design, construction and in-service phases of the ship's life cycle.

During all life cycle activities information is electronically transferred to and from the design/planning yard and in-service agents. This includes product model data such as product structures and key data such as Equipment Functional Descriptions (EFD) and Hierarchical Structural Codes (HSC). It also includes ship maintenance and completion record and Repairable Identification Codes (RIC).

Design / Construction Logistics Activities

Logistic activities during the design process require establishing a baseline configuration of a hull based upon the as-designed ships models. When the as-designed baseline changes, notification is made to logistics personnel, as this may result in updates to their products.

Throughout the design process training manuals, interactive documents and an illustration repository are developed.

During the ships construction the as-built configuration is established. Component changes are evaluated for logistics impact and recorded in the as-built configuration. This is also the time when the Coordinated Shipboard Allowance List (COSAL) is developed. This COSAL is maintained for the life of the ship.

Information which supports logistics product development is tracked throughout the lifecycle of a part. Vendor furnished and supply support information along with the ability to cross reference supply information from navy stock systems or alternate shipyard part systems and track stowages is required. Attributes such as reliability, maintainability, and availability of parts as well as the location, status, etc. of hazardous materials are maintained.

In Service Logistics Support

After a vessel has been commissioned In-Service Logistics Support begins. The ship's as-built baseline transforms into the as-maintained configuration. The ship's as-maintained configuration captures the maintenance which results from a change to a component whether it is a modification, deletion or addition. It also captures changes that result from modernization of the ship because of technology advances in its components.

Logistics modernization support provides the information required to modernize equipment and the feedback that work has been accomplished. The feedback loop allows the correct logistics products to be identified and supported for the ship.

Logistics maintenance support provides historical data to support component selection, modernization, and component removal for the ship. The data is analyzed to support the quantity of repair items on the ship and the maintenance periodicity.

2.5 Integrated Logistics Support Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must have the ability to manage design, engineering and construction change that occur and affect logistics products including tracking changes to hierarchical structural codes.	3.2.1 3.2.2			
2) The IPDE must have the ability to configuration manage vendor furnished information (VFI) and engineering data (e.g. diagrams, drawings, drawing parts lists, 3D models, etc.) in support of Logistics.	3.2.1 3.2.3 3.2.5			
3) The IPDE must have the ability to manage Logistics work product information (e.g. RMA data, maintenance data, technical manuals, training materials, supply support information, etc.) and link this information to the logistics configuration items.	3.2.1 3.2.3 3.2.5			
4) The IPDE must have the ability to manage changes to the as-maintained baseline and associate these changes with the appropriate ship configuration item(s). Examples of changes include ship alterations, ship maintenance completion records, and warranty and guarantee material fixes.	3.2.1 3.2.2	3.3.4 3.3.5		
5) The IPDE must have the ability at the part level, to manage data in support of logistics product development throughout the lifecycle of a part.	3.2.1 3.2.2	3.3.2 3.3.4		
6) The IPDE must have the ability to maintain a ship's COSAL (Coordinated Shipboard Allowance List).	3.2.1	3.3.5		
7) The IPDE must have the ability to maintain ship alteration work packages.	3.2.1			

2.5 Integrated Logistics Support Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
8) The IPDE must have the ability to record and track warranty and guarantee information against materials and any repairs related to them.	3.2.1		3.4.2	
9) The IPDE must have the ability to manage Liaison Action Requests and Conditional Reports and associated design changes.	3.2.1 3.2.2 3.3.3			
10) The IPDE must have the ability to manage change flows which affect logistics products including managing changes to logistics product model baselines.	3.2.1 3.2.2			
11) The IPDE must provide a common illustration repository and be able to deploy interactive documents.	3.2.1 3.2.2	3.3.1 3.3.2 3.3.3		
12) The IPDE must have the ability to notify Logistics of design, engineering and construction changes. This will facilitate the process of updating relevant Logistics products based on approved changes and ensure the most current information is being used.	3.2.1 3.2.2			
13) The IPDE must support the development of logistics products including Logistics Configuration baselines, technical data, training materials, maintenance material, RMA (Reliability, Maintainability, Accountability) analysis, Configuration Status Accounting and Supply Support analysis.	3.2.1 3.2.3	3.3.5		
14) The IPDE must have the ability to cross reference supply information from navy stock systems or alternate shipyard part systems and track stowages.				4.2.1 4.2.2 4.2.5
15) The IPDE must have the ability to electronically exchange product model data between multiple shipyards during life cycle activities.		3.3.3		4.2.1 4.2.2 4.2.5
16) The IPDE must have the ability to transfer logistics and configuration information and products to and from the Navy Program's of Record at the transition from design/construction to life cycle support and throughout the in-service phase.		3.3.3		4.2.1 4.2.2 4.2.5
17) The IPDE must have the ability to transfer ship maintenance completion records and other applicable data to and from the design/planning yard and in-service agents		3.3.3		4.2.1 4.2.2 4.2.5

2.5 Integrated Logistics Support Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
18) The IPDE must have the ability to exchange ILS product model data among design yard, planning yard, government or in-service agents.		3.3.3		4.2.1 4.2.2 4.2.5
19) The IPDE must have the ability to exchange ship alteration work packages with the Planning Yard, other shipyards, and the Navy.		3.3.3		4.2.1 4.2.2 4.2.5
20) The IPDE must provide the ability to maintain the reliability block diagram content associated with the ships FSIs (Functionally Significant Items).	3.2.1	3.3.5		

Table 2-5 Shipbuilding IPDE needs for Integrated Logistics Support

2.6 *Requirements Management Activities*

Requirements management is the practice of gathering and managing business, technical, functional, and process requirements within a product development environment.

A shipyard stores multiple baselines of contractual requirements. These requirements are tracked from the ships conceptual stages through its lifecycle. Attributes such as where the requirement came from, how it was satisfied, tested, and changes that were made to it are all accounted for.

2.6 Requirements Management Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must have the ability to store multiple baselines of contractual requirements.	3.2.1		3.4.2	
2) The IPDE must have the ability to track a requirement through all phases of its development.	3.2.1 3.3.3 3.3.3		3.4.2	
3) The IPDE have the ability to manage bi-directional links between specification requirements and configuration items.	3.2.1		3.4.2	
4) The IPDE must support traceability in the forms of change management and revision control as the requirements change.	3.2.1 3.2.2 3.2.3			

Table 2-6 Shipbuilding IPDE needs for Requirements Management

2.7 *Collaboration Activities*

Collaboration is the foundation for bringing together the knowledge, experience and skills of multiple team members to contribute to the development of a product. It may take place during all phases of shipbuilding.

Data, including but not limited to CAD data, documents, the BOM and design/construction schedules are electronically exchanged between multiple Integrated Product Teams (IPTs). These IPTs may be in the same shipyard or span multiple shipyards, vendors, government agencies, etc.

Concurrent access capabilities to 3D product model data are also needed for IPT reviews. During the review process customer comments are captured and tracked.

Workflow functionality through the IPDE is needed to support design/engineering collaboration for work authorization, customer and vendor product design reviews, interfaces with ERP applications for scheduling, EVM progressing for payments of work performed, and fleet support lifecycle maintenance changes and approvals for technical insertion.

Users involved in a collaborative effort have access only to the data needed to perform the task at hand. The approval process is based on the responsibilities of each user. When changes are required only the data affected is allowed to be unlocked and changed.

2.7 Collaboration Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must have the ability to provide concurrent access capabilities to the 3D product model for IPT reviews. It must have the ability to capture and track customer comments during program reviews.		3.3.1 3.3.3		
2) The IPDE must have the ability to manage data in a manner which easily allows multiple Integrated Product Teams (IPTs) to work concurrently.	3.2.1 3.2.2	3.3.1 3.3.3	3.4.1	
3) The IPDE must have the ability to provide a consistent change management and notification process to alert all affected parties, including vendors and subcontractors of changes as they occur.	3.2.2 3.2.3	3.3.3 3.3.5	3.4.2	
4) The IPDE must have the ability to identify and locate specific parts of the ship and be able to visualize them in a team design review setting.	3.2.1	3.3.4		
5) The IPDE must provide access to data based on user authorization.		3.3.1 3.3.3		
6) The IPDE must have the ability to provide only the data needed to perform the work.		3.3.1		
7) The IPDE must enable a user to make change by unlocking only affected data.		3.3.1		
8) The IPDE must provide the ability to add or associate data to data which has already been released by another group without affecting released data or requiring its reapproval.	3.2.2			

2.7 Collaboration Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
9) The IPDE must have a workflow capability that allows the routing of information for review, comment and approval			3.4.1	
10) The IPDE must have the ability to provide an electronic approval process based on user responsibilities.	3.2.2	3.3.1 3.3.3	3.4.1	
11) The IPDE must provide the capability to electronically transmit data, including but not limited to Engineering documents, CAD data, documents, the BOM and design/construction schedules.				4.2.1 4.2.2 4.2.5
12) The IPDE must have the ability to exchange data within the same shipyard or to multiple shipyards, vendors, government agencies, etc.				4.2.1 4.2.2 4.2.5

Table 2-7 Shipbuilding IPDE needs for Collaboration

2.8 *Reporting & Publishing Activities*

Reporting and publishing is the process of assembling information and disseminating it to the proper people, at the proper time, and in the proper format. Reporting and publishing activities are conducted through all phases of the shipbuilding process.

Product data is retrieved in many different ways; by discipline, product type, space, etc. and is published as prearranged deliverables (e.g. government shipyard paper vs. private yard paper) as well as ad hoc reports. Information is published to multiple end-user devices (e.g. PC, wireless tablet, etc) and in multiple formats (Text, MS Excel, HTML, etc) providing the end user with only the data which is needed at any given time.

Drawings are published by assembling elements from the product model. A drawing may include 2D graphics, joint surface index, engineering parts lists bills of material, general notes, etc.

2.8 Reporting and Publishing Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
1) The IPDE must be able to report and publish through all phases of the shipbuilding process. It must be able to retrieve data in many different ways; by hull, discipline, product type, space, etc	3.2.1 3.2.3	3.3.2 3.3.3 3.3.4 3.3.5		
2) The IPDE must have the ability to publish data as prearranged deliverables, as well as creating ad hoc reports.	3.2.1	3.3.5		
3) The IPDE must have the ability to automatically assemble elements from the product model and publish a drawing.	3.2.1 3.2.3	3.3.2 3.3.4 3.3.5		

2.8 Reporting and Publishing Needs	3.2 Configuration Management	3.3 Product Data Access	3.4 Business Process Control	4.2 IPDE Interoper-ability
4) The IPDE must have the ability to publish deliverables to multiple end-user devices.	3.2.1	3.3.1 3.3.3 3.3.4 3.3.5		4.2.2
5) The IPDE must provide the ability to publish in multiple formats.	3.2.1 3.2.3	3.3.4 3.3.5		4.2.2
6) The IPDE must have the ability to search parameters within the product model and extract data accordingly.		3.3.2		

Table 2-8 Shipbuilding IPDE needs for Reporting and Publishing

3.0 CORE IPDE CAPABILITY REQUIREMENTS

3.1 Overview

This section, along with Section 4, contains the enforceable IPDE requirements. These requirements are derived from the Shipbuilding Needs described in Section 2 and address the *core* Product Data capabilities of an IPDE. These core capabilities, depicted in Figure 3-1, are organized into three main areas: configuration management, product data access, and business process control.

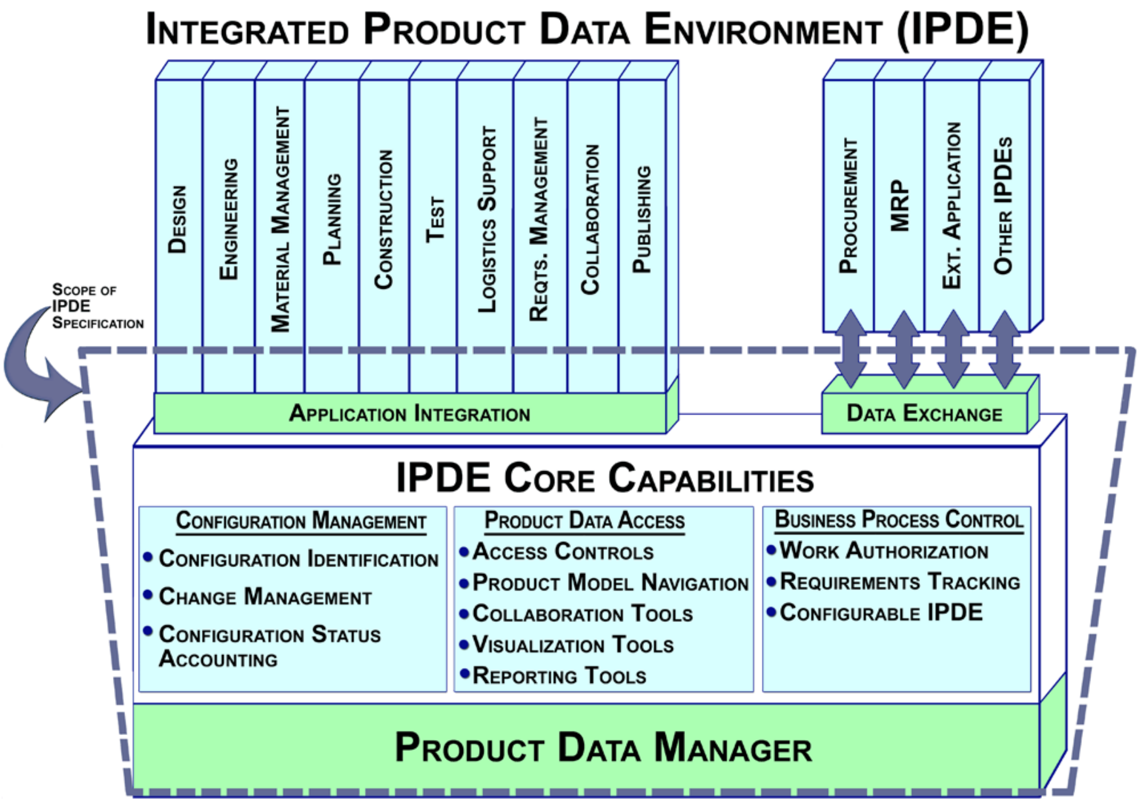


Figure 3-1 *IPDE Core Capabilities*

It is not expected that these requirements will be satisfied by one software application, rather it is anticipated that there may be several applications, whether they are commercial, government, or custom shipyard-specific applications, that are integrated or interfaced to provide a shipyard-specific implementation of these requirements. The IPDE core capabilities may be satisfied by a combination of COTS and Custom-built applications. The specific allocation of software requirements to applications will be shipyard-specific. For example, suppose that Application A is the PDM application used in a shipyard. The entire application is within the shipyard IPDE, but it only implements a portion of the core IPDE capability defined in this specification. Application B may provide the additional PDM capabilities required by this specification. These applications may also provide additional PDM capabilities that may be used as part of the overall shipyard IPDE implementation. This additional functionality may be utilized within the shipyard, but it is not part of the core IPDE capabilities specified in this document. Another point to note is that there may be overlap between functionality provided by one application and functionality provided by another application. Again, how this functionality is integrated or utilized within a specific occurrence of an IPDE depends on the needs of the shipyard and/or program.

The requirements in this section are organized around the three main core capability areas: managing product data, accessing product data, and business process control. Section 3.2 contains the configuration management requirements, including Configuration Identification, Configuration Change Control, and Configuration Status Accounting. Section 3.3 contains the requirements for accessing product data, including Access Controls and Permissions, Product Model Navigation, Collaboration Tools, Visualization Tools, and Reporting and Publishing Tools. Section 3.4 contains the business process control requirements, including Work Authorization, Requirements Tracking, and Configurable IPDE.

3.2 Configuration Management

Configuration Management (CM) is the process that establishes and maintains consistency of a product's attributes and product configuration information with its requirements throughout the product's life cycle.

This section is organized by the key configuration management pillars (functions) defined in ANSI/EIA-649-A, National Consensus Standard for Configuration Management, 28 October 2004. The key pillars, depicted in Figure 3-2, include Configuration Identification, Configuration Change Management, and Configuration Status Accounting. The CM functions of Planning & Management and Verification & Audit encompass programmatic and process-related aspects of configuration management and are not covered in this specification.

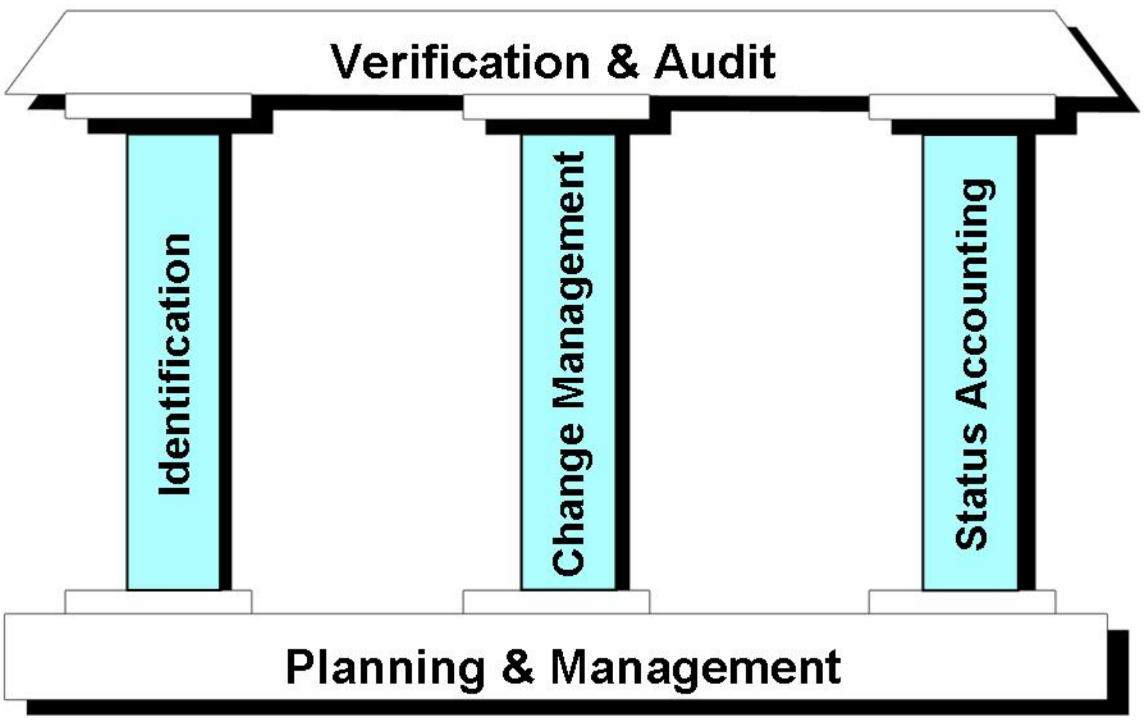


Figure 3-2 *Functions of Configuration Management*

3.2.1 Configuration Identification

Configuration identification is the basis from which the configuration of products is defined and approved; products and documents are labeled; changes are managed; and accountability is maintained throughout the product life cycle. Configuration identification establishes:

- A method for organizing the composition of the product elements and associated information
- Unique identification of products and product configuration information
- Consistency between a product and the information about the product
- Product attributes that are defined, documented and baselined

This section defines the minimum set of configuration items an IPDE is required to manage (section 3.2.1.1) along with the requirements needed to manage these configuration items (sections 3.2.1.2 through 3.2.1.13).

3.2.1.1 Configuration Items (CIs)

A *Configuration Item (CI)* is an item that is configuration managed within the IPDE. Examples of configuration items include the product being built and its components (e.g. hull, systems, assemblies, parts, etc.) as well as supporting products used to design, manufacture, operate, or maintain the product (e.g. requirement documents, system diagrams, design drawings, 3D models, manufacturing aids, assembly drawings, technical manuals, training materials, operating procedures, etc.).

R-1: The IPDE shall provide the capability to manage the following set of core configuration items:

- ***Ship Class***
- ***Flight***
- ***Hull***
- ***Unit/Block***
- ***Compartment***
- ***System***
- ***Assembly***
- ***Part-Occurrence***
- ***Connections, Joints, and Penetrations***
- ***Attribute Group***
- ***Logistics***
- ***Document***
- ***Output Product***

These configuration items are defined in the following sub-sections.

R-2: The IPDE shall provide the capability to define and manage program-specific or shipyard-specific configuration items.

R-3: The IPDE shall provide the capability to manage a core set of data entities, attributes, and relationships associated with a configuration item. Configuration item entities, attributes, and relationships are defined in the Ship Common Information Model (SCIM).

Figure 3-3 is a view of a compartment which will be used to illustrate the various configuration items throughout this section of the specification.

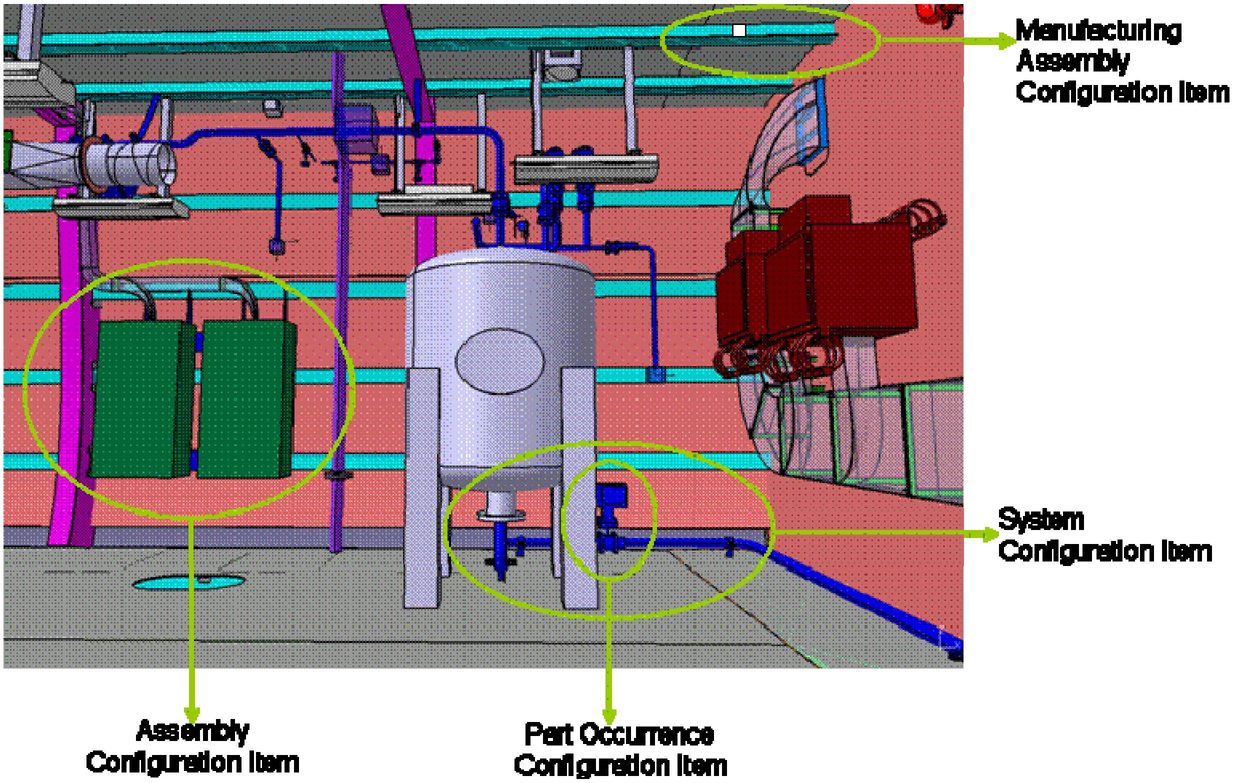


Figure 3-3 ***Various Configuration Items used as examples throughout this section***

Ship Class Configuration Items

A *Ship Class* is a group of ships which share the same basic hull form and characteristics. A ship class may consist of flights or may simply be a collection of hulls.

Flight Configuration Items

A *Flight* is a series of hulls within a ship class defined by a contract which share the same basic tactical capabilities.

Hull Configuration Items

A *Hull* is a specific ship configuration (one of a kind) within a class of ships.

Unit/Block Configuration Items

A *Unit/Block* is a construction entity developed during the planning phase to facilitate the construction of the ship. Construction planners generally start with an erection diagram which breaks down the ship into erection units / blocks. To facilitate the fabrication of steel, the erection units are designed to be as close to identical as possible in size (or weight). The size (or weight) of the erection units selected are usually limited by the amount of crane capacity available.

Compartment Configuration Items

A *Compartment* is a partitioned section of a ship that is designated for a specific duty or use.

System Configuration Items

A *System* is an object intended to meet a specific requirement. In the early stages of a ship's life cycle, a system may be represented by a requirement describing the desired characteristics that the physical system must meet. A system can also be represented by two or more systems (also called sub-systems) and/or parts that are designed to meet the specific requirement.

Figure 3-4 illustrates a system configuration item. In this case the system configuration item consists of several part occurrences, such as a motor operated valve, hangers and pipe segments.

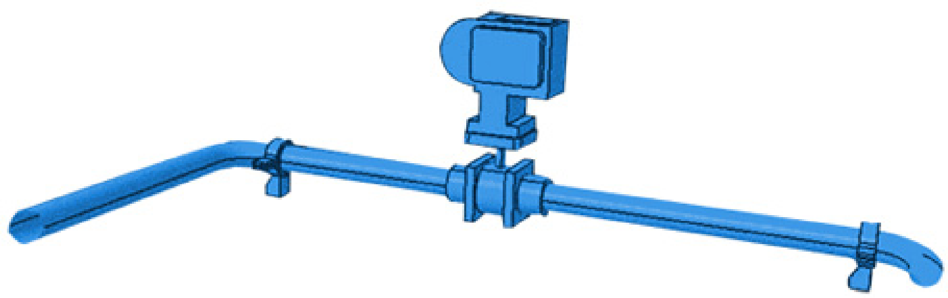


Figure 3-4 *Example System Configuration Item*

Assembly Configuration Items

An *Assembly* consists of one or more assemblies (also called sub-assemblies) and/or parts, which may or may not be a member of the same system. From the design point of view, assemblies can be used to represent parts of the design (e.g. a break-out of a system into lower level groups of parts). From the manufacturing point of view, assemblies are used to define the build sequence for a hull. The highest level assembly for a ship, the hull assembly, consists of combinations of lower level assemblies.

Figure 3-5 illustrates an assembly configuration item. In this case the assembly configuration item consists of part occurrences, such as electrical power panels, foundation structure and chocks. The foundation structure supporting the power panels could represent a sub-assembly.

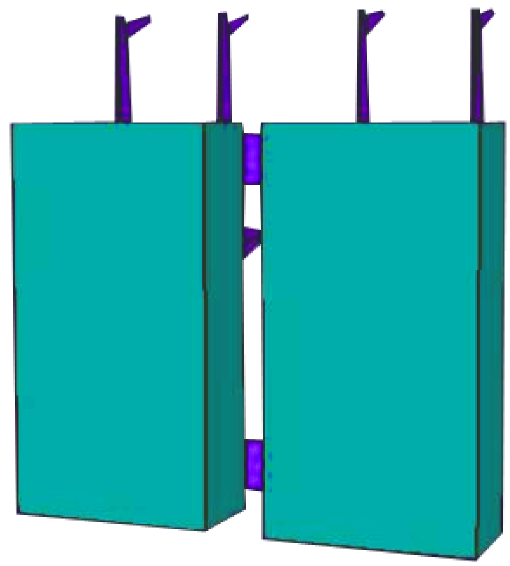


Figure 3-5 *Example Assembly Configuration Item*

Figure 3-6 illustrates the use of manufacturing assembly configuration items to define the assemblies and sub-assemblies relative to objects in the work breakdown structure. In this case, assembly “MO-9” consists of two part occurrences of the “I” Beam (Part D) and one part occurrence of a plate (Part C). Assembly “MO-20” consists of the “MO-9” sub-assembly along with the part occurrence “O-11” of the plate with a hole (Part B). The “MO-20” assembly configuration item is associated with the M2 node of the work breakdown structure. The part occurrences are not configured to the manufacturing assembly; they are configured to the ship in the design product structure. The manufacturing assemblies are configured to the ship in the manufacturing product structure. However, the manufacturing assemblies can reference the part occurrences (an example of the relationships between the different product structures).

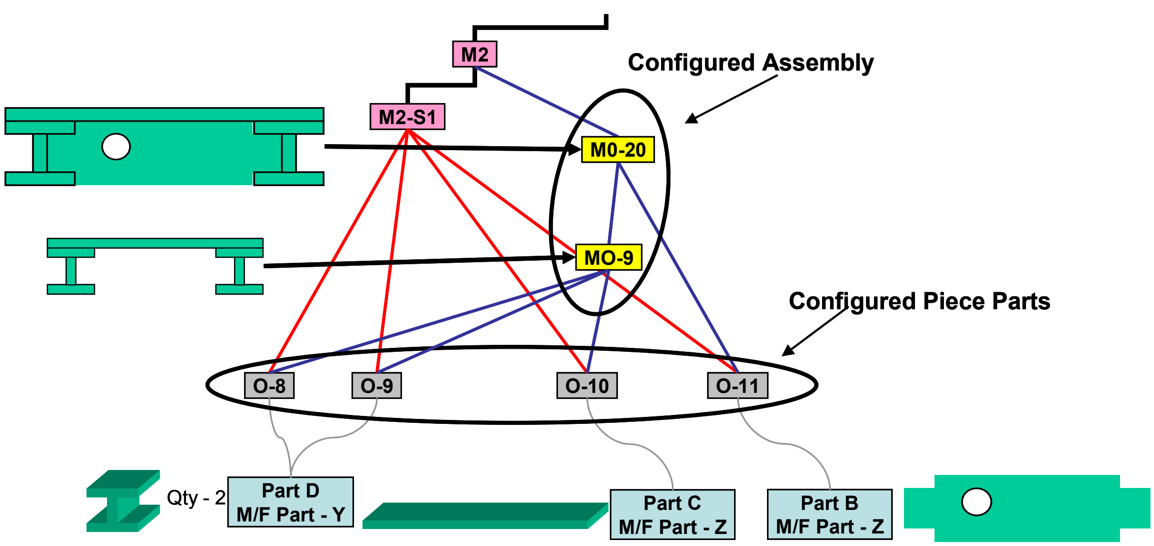


Figure 3-6 *Example Manufacturing Assembly Configuration Items*

Part-Occurrence Configuration Items

A *Part-Occurrence* is the instantiation of a part for a specific usage in the ship. A Part-Occurrence Configuration Item is a constituent piece in the ship's product structure included at the time of design and/or manufacture. It identifies specific usage information that is configuration managed, such as where an individual part is used or its find number within a hull or set of hulls.

A *Part* may be a single part or may be a group of piece parts. There are many types of parts. A *Make Part* is generally a part that is designed in context to the ship. A *Catalog Part* is generally a part that can be ordered or stocked in a warehouse and is managed in a part library or catalog. This is also known as a standard part. Make Parts, Catalog or Standard Parts, NSN Parts, OEM Parts are all components which may be instantiated within a product model as a Part-Occurrence. A Part-Occurrence configuration item has a relationship to the part which is maintained throughout the life of the ship. This association allows access to the part definition data, part specifications, part graphics and so forth.

Figure 3-7 provides an example of a Part-Occurrence configuration item. In this case, the Part-Occurrence configuration item is an occurrence of the motor operated valve.

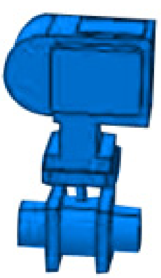


Figure 3-7 *Example Part-Occurrence Configuration Item*

Connections, Joints, and Penetrations Configuration Items

Based on lessons learned from previous and current programs, it is important to manage connections, joints (both mechanical and welded), and penetrations in a similar manner as Part-Occurrences. Configuration management of connections, joints and penetrations as configuration items is required to allow the identification, location, and access to specific system parameters of each of these items. This is particularly important in today's shipbuilding environment given the emphasis placed on penetration and joint control and the amount of work share activities being performed on current contracts. The IPDE needs to manage logical or functional connections as well as physical connections to support connectivity, availability, and modeling analysis (such as pipe stress or cascading failure analysis).

Attribute Group Configuration Items

Configuration management of attribute groups is required in order to allow attributes to be updated independently by different organizations or personnel and to support status and workflow processes. Attribute Group Configuration Items are required to provide the ability to define an attribute group or groups associated to any other type of Configuration Item that can be created, updated and released independently by different users.

An example of this is illustrated in Figure 3-8 in which the part graphic design is updated by the Design representative and the part definition attributes are updated by the appropriate representatives (Material, Weights, and Environmental).

R-4: When part definition attribute(s) are revised, the IPDE shall require a revision of the part to which the attribute(s) belongs.

The Part-Occurrence attributes are specifically related to the part used in the products that are updated by the Design and Logistics users in Attribute Group Configuration Items. These Attribute Group Configuration Items are related to other types of configuration items, in this case with a part-occurrence configuration item, and are configuration managed with the rest of the product structure. An example of this is illustrated in Figure 3-8 in which the Attribute Group configuration items are updated by the appropriate Design and Logistics representatives.

R-5: The IPDE shall provide the capability to distinguish between attributes that "define" a part (part definition attributes) versus attribute group configuration items that provide additional characteristics of the object in a particular usage (part-occurrence).

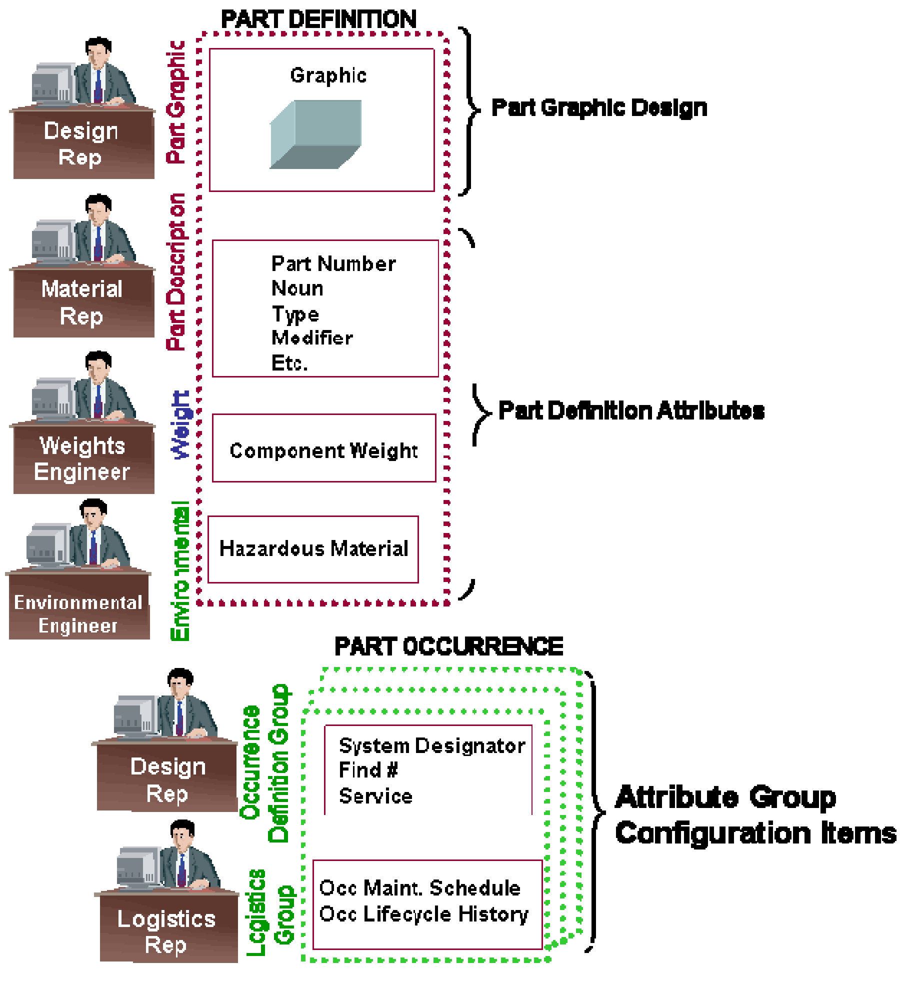


Figure 3-8 *Example Attribute Group Configuration Item*

Logistics Configuration Items

A Logistics *Functionally Significant Item (FSI)* is an item whose failure would have a significant impact on the availability of a required function or life cycle costs. An FSI can be a system, subsystem, equipment, component, or a summary level of two or more FSIs. For life cycle management purposes, an FSI is any item that requires any element of logistics support requiring configuration information to operate or maintain the ship.

Document Configuration Items

Document configuration items include both the metadata for an object and the physical file (or set of files) for that object. Examples of document configuration items include plans, schedules, design documents, technical manuals, change documents, work packages, white papers, analyses, reports, test plans and procedures, certifications, review comments, etc.

Output Product Configuration Items

Output product configuration items represent the various products created and managed throughout the design, construction and lifecycle support phases of a contract. These output products are mostly documents and files associated with the configuration items defined above which are created at intermediate steps in the design and construction process to support design reviews, analysis, etc. Examples of output products include diagrams, drawings, drawing part lists, drawing sheets, 3D models, analysis results, and bills of material.

3.2.1.2 Configuration Items by Phase

In order to effectively and efficiently manage product data over the life cycle of the ship, it is extremely important to understand which configuration items or aggregate of configuration items need to be configuration managed at each phase of the ship's life cycle, along with the processes by which the configuration items are managed within the IPDE. If lower level configuration items (such as parts) enter the change process too soon, the cost of maintaining changes to these items will drive up the cost of the ship development. On the other hand, if enough detail is not configuration managed at a certain phase, or the detail is not managed properly with the IPDE, this could also increase the cost of change and cause excessive rework in down stream processes such as logistics and manufacturing. This is illustrated in the Configuration Management scenario provided in Appendix A. The conclusion drawn from the scenario is that it is desirable to have the capability to perform part-based configuration management at specific phases of the ship's life cycle. Specifying the management of a particular configuration item at one level during a phase does not preclude the capability to manage other configuration items at a lower level if required by a specific program.

R-6: The IPDE shall provide the capability to manage configuration items at different levels during different phases of the ship's life cycle.

3.2.1.3 Unique Identification

Unique identification of configuration items is required so that one product can be distinguished from other products; one configuration of a product can be distinguished from another; units (occurrences) of the product can be distinguished from other units (occurrences) of the product, and the source of a product can be determined.

R-7: The IPDE shall provide the capability to uniquely identify configuration items.

R-8: The IPDE shall provide the capability to maintain the configuration item identifier throughout the ship's life cycle.

3.2.1.4 Configuration States

R-9: The IPDE shall provide the capability to apply promotion states to a configuration item as it moves through the life cycle. At a minimum, these states shall include: Working, Released, and Archived (as defined in ANSI/EIA-649-A).

- *Working.* Represents work in progress, which is highly subject to change without notice. Sub-states can be defined for working information to distinguish information that is draft state, review state, etc.
- *Released*. Released information has been reviewed and authorized for use or, if required, has been authorized for submittal to or access by, a customer. Released information is under configuration change management. A new version cannot replace a released version until the new version has also been reviewed and authorized. The configuration of an item is fixed once it is in the released state and is only changed by release of a superseding revision.
- *Archived.* Archived information is released information (reviewed and approved) that is no longer authorized for continued use. It is kept as a historical record for reference purposes (e.g., redesign, failure analysis, regulatory requirements, etc.).

R-10: In addition to the states defined in R-9, the IPDE shall provide the capability to define program-specific or shipyard-specific states.

R-11: The IPDE shall provide the capability to record each unique occurrence of a configuration item as a version.

R-12: The IPDE shall provide the capability to control configuration item versions using a set of rules implemented for the different configuration items.

3.2.1.5 Configurations

R-13: The IPDE shall provide the capability to identify and manage groups of configuration items, also known as a configuration. A configuration is a collection of configuration items that make up a specific configuration of the ship.

R-14: The IPDE shall provide the capability to manage the As-Designed, As-Built, and As-Maintained configurations of the ship.

3.2.1.6 Configuration Item Relationships

R-15: The IPDE shall provide the capability to associate configuration items with other configuration items such as hulls, subsystems, compartments, documents, or other output products.

R-16: The IPDE shall provide the capability to associate various representations to configuration items in order to allow access to the information that describes the configuration item.

Representations of configuration items include, but are not limited to, 2D logical representations, 3D physical representations, removal envelopes, shock envelopes, etc..

R-17: The IPDE shall provide the capability to link document configuration items to configuration items at all levels of the product structure.

R-18: ***The IPDE shall provide the capability to link related engineering data, vendor furnished information (VFI), government furnished information (GFI), and logistics information (e.g. Reliability, Maintainability and Availability (RMA) analyses, Maintenance Engineering Analysis (MEA), Technical Manuals, Training Materials, Provisioning data, etc.) to the logistics configuration item.***

R-19: ***The IPDE shall provide the capability to maintain configuration item associations throughout the ship's life cycle.***

Figure 3-9 provides an example of linking the logistics configuration item with associated engineering and logistics information. The figure only illustrates key attributes of the Logistics configuration item (e.g. drawing number, catalog number, repairable identification code, hierarchical structure code, and item control number) used to link to other information, not the full set of attributes. Example engineering and logistics products linked to the logistics configuration item include 3D models, technical data, supply data, vendor data, drawings, RMA data, catalog data, and configuration status accounting data. This linking allows access to all relevant logistics and engineering data from a central index, the Hierarchical Structure Code (HSC), which is based on the ESWBS.

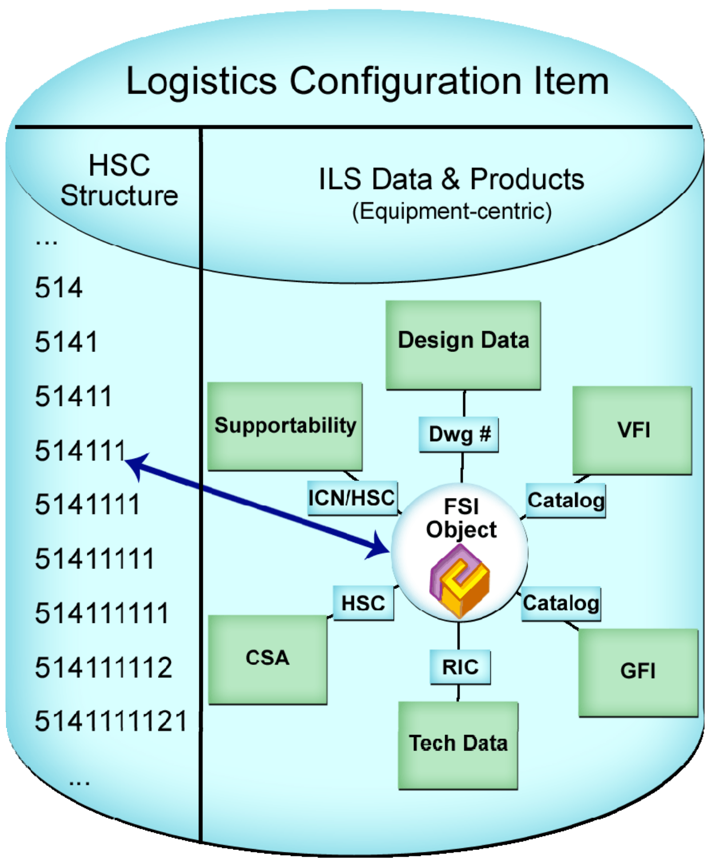


Figure 3-9 *Example Configuration Item Relationships*

3.2.1.7 Applicability

Applicability is the authorization (appropriate, suitable) of a part to be used (a Part-Occurrence instantiated) in a higher level configuration item such as a class of ship (e.g., a part or specification/requirement is applicable to the Virginia and Los Angeles classes and can be used in their design). This is sometimes referred to as Program Applicability.

R-20: The IPDE shall provide the capability to specify which parts may be used as Part-Occurrence configuration items under other higher level configuration items.

3.2.1.8 Hull Effectivity

Hull Effectivity is the specific assignment of a configuration item to a specific hull or series of hulls (e.g., a given part or assembly is instantiated on hull 3 of a specific ship class at location x, y, z). Hull effectivity defines the configuration or range of configurations (i.e. hull or range of hulls) to which the configuration item belongs.

R-21: The IPDE shall provide the capability to assign and update the hull effectivity of a configuration item.

The following are some examples of these relationships:

- hull effectivity of system and assembly configuration items
- hull effectivity of Part-Occurrence configuration items
- hull effectivity of the attribute group configuration items
- hull effectivity of connection, joint, weld or penetration configuration items
- hull effectivity of document configuration items

Figure 3-10 illustrates the hull effectivity of Part-Occurrence configuration items to the ship class configuration item. The term AFS refers to “All Follow Ships” and means that this effectivity applies to all the successive ships in the class.

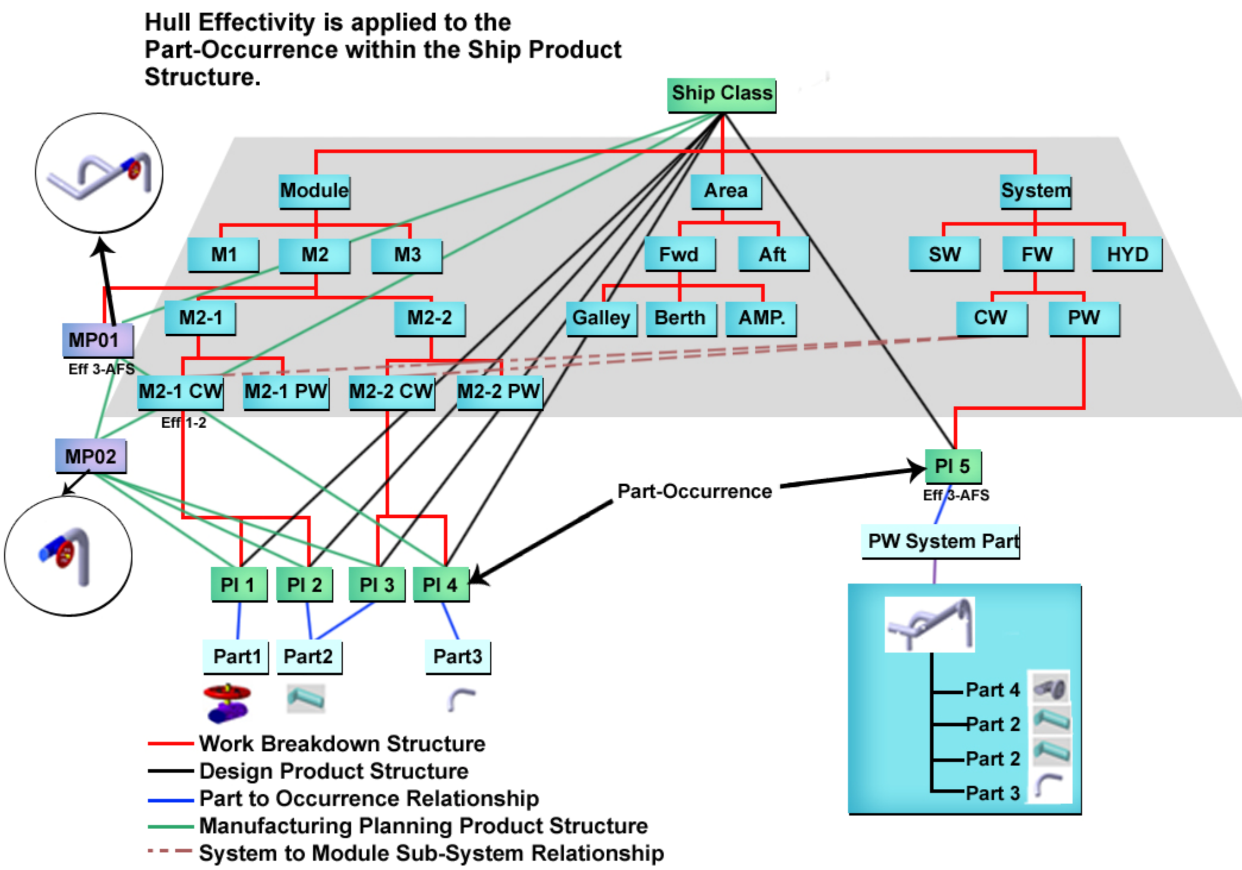


Figure 3-10 *Hull Effectivity*

3.2.1.9 Baselines

This section describes the requirements to identify configuration items at their specific versions to support baselines. A baseline is a group of released configuration items developed during a specific phase of the development process that has been formally accepted. Once the baseline is established, changes to the items can only be made through a formal change process. Baselines incorporate all completed changes. Open or pending changes remain as liabilities.

R-22: The IPDE shall provide the capability to associate specific versions of configuration items to one or more baselines.

R-23: Baselines shall have a unique name and a human readable textual description of the purpose.

R-24: The IPDE shall enforce configuration management of the baselines and the changes impacting these baselines.

Figure 3-11 depicts the evolution of a baseline. The first diagram shows a group of configuration items which is part of the original design. The design is issued for hulls 1-AFS while construction is issued for hulls 1 and 2. In the second diagram, two changes are applied which are issued for design on hulls 3-AFS. Change 1 is construction issued on hull 3 only, while change 2 is not yet construction issued. The third diagram indicates that change 1 has been incorporated into the baseline. The new baseline consumes change 1 and is design issued for hulls 3-AFS, and construction issued for hull 3 only. Change 2 has not been consumed as part of the baseline therefore remains as an open liability.

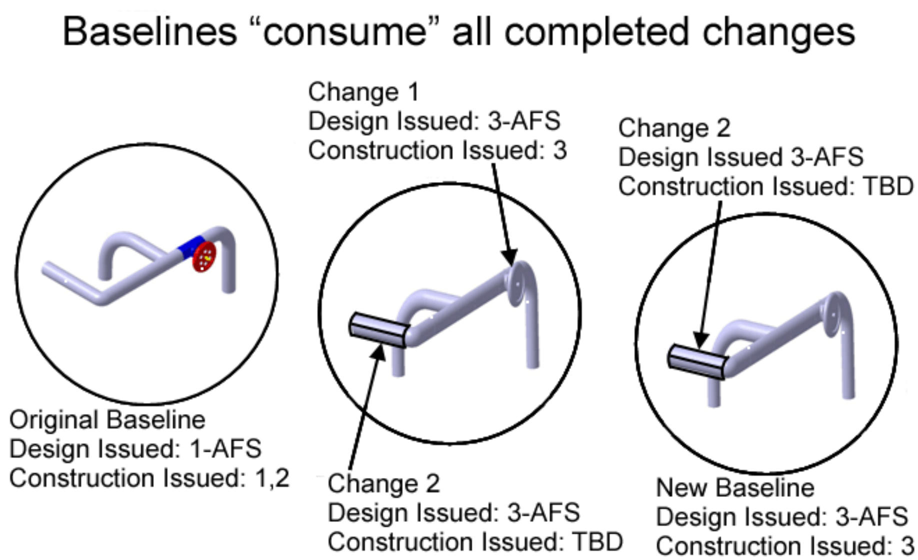


Figure 3-11 *Baselines*

R-25: In addition to formal baselines, the IPDE shall provide the capability to manage intermediate baselines or “snap-shots”.

Intermediate baselines are persistent groupings of configuration items, some of which may not be released, which are created at various stages in the design and construction process to support design reviews, analysis, etc. Intermediate baselines capture in-work effort at a specific point in time. By applying configuration to these “snap-shots” in time, it provides a method to revert back to an intermediate set of products should the need arise.

3.2.1.10 Variants

A variant is a group or collection of configuration items applicable to a class, flight, or hull, which is not part of the formal ship baseline or product structure. Variants are created to support a specific intent (such as analysis of design alternatives or ship alteration) and may share common elements of the product structure. Variants can be incorporated into the product structure as needed to support design decisions or modernization activities.

R-26: The IPDE shall provide the capability to manage multiple variants applicable to the same class, flight or hull.

R-27: Variants shall have a unique name and a human readable textual description of the purpose.

R-28: The IPDE shall provide the capability to compare variants in order to support alternate designs, change impact, and other analyses.

R-29: The IPDE shall provide the capability to incorporate variants into the ship product structure as the design matures and decisions are made to utilize a particular variant in the baseline design.

Figure 3-12 depicts an example of variants. A particular Ship Class has a configuration with different variations of the baseline. For example, diagram A represents the configuration which uses two occurrences of part 1, one occurrence each of parts 2 and 3 and is effective for hulls 1 and 2. Diagram B represents a variant of the diagram A configuration but removes part 3 and adds part 4. Diagram C represents the configuration for hulls 3 and AFS which replaces part 2 from diagram A with part 5. Diagram D represents a variant of the configuration from diagram C but removes part 3, adds part 4.

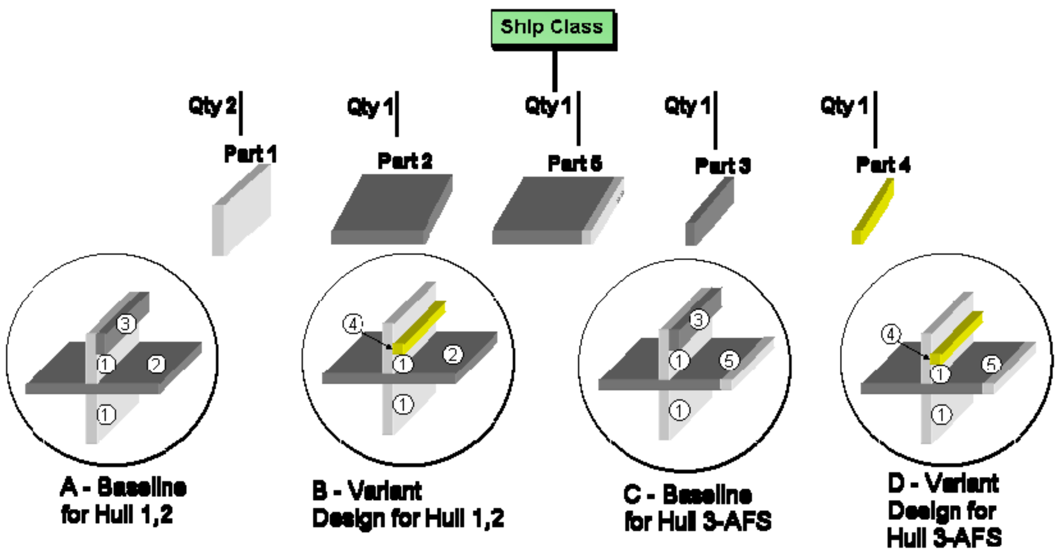


Figure 3-12 *Variants*

3.2.1.11 Product Structures

A *Product Structure* is a hierarchical decomposition of a ship defining configuration managed parent/child relationships between configuration items. There will be multiple product structures within the ship. Some examples include: logical design, physical design, manufacturing assembly, and logistics structure.

R-30: The IPDE shall provide the capability to organize ship configuration items into multiple product structures to support different functional areas and life cycle phases.

R-31: At a minimum, the IPDE shall provide the capability to support the following product structures:

- ***Design Product Structure***
- ***Manufacturing Product Structure***
- ***Logistics Product Structure***

The Design Product Structure defines the configuration items of all the components needed in the ship arrangement.

The Manufacturing Product Structure defines the configuration items of all the manufacturing parts, assemblies, and sequences needed to build the ship.

The Logistics Product Structure is the ship's structure and indexing system for logistics data. It defines the functional descriptions for the ship's FSIs. The Logistics Product Structure is very dynamic in the early stages of logistics analysis as the design is being refined and equipment is being decomposed into maintenance-worthy items. This necessitates a flexible product structure that can accommodate changes without causing major impact to previously completed work.

The Hierarchical Structure Code (HSC) is the index into the Logistics Product Structure. The HSC is composed of the Expanded Ship Work Breakdown Structure (ESWBS), 5 characters, plus 7 additional characters to fully identify top-down breakdowns of systems to equipment at the lowest functionally significant member. The 5 ESWBS characters are defined by the Navy to identify like systems and equipment.

R-32: The IPDE shall provide the capability to manage changes to product structures throughout design/construction and life cycle support, including the maintenance of links to related data.

3.2.1.12 Work Breakdown Structures

A *Work Breakdown Structure (WBS)* is a hierarchical decomposition which organizes and defines the hardware, software, services and related work tasks required to develop a ship. A WBS is often used as a framework for organizing and viewing the configuration items that make up the product structure in different ways. Examples of various work breakdown structures include:

- Ship Work Breakdown Structure (SWBS) - a three-digit hierarchical index used to organize the systems and sub-systems of the ship into standard functional groups that are used throughout the Navy
- Expanded Ship Work Breakdown Structure (ESWBS) - expands the SWBS to 5 digits allowing for further categorization and breakdown of equipment to lower levels of detail
- Zone (Area) - decomposition of the ship by major areas usually used in the design phase to organize design work packages
- Compartment (Module) - decomposition of the ship by physical area on the ship
- System - functional decomposition of the ship by system and sub-system
- Manufacturing Unit/Block - decomposition of the ship into major manufacturing areas

A Work Breakdown Structure is not necessarily configuration managed whereas the ship Product Structures are configuration managed. The difference is that WBS does not contain the product configuration items but is rather a product oriented 'family-tree' division that relates to the configuration items. The configuration items are not configured in relation to the WBS, they are configured in relation to the Product (i.e. Ship). The WBS structure is "derived" from the underlying product structure and data.

Error! Reference source not found. depicts the product structure and its relationship to work breakdown structures based on ship module, SWBS, and ESWBS. Each WBS provides a logical way of dividing the ship in order to emphasize different aspects of the design or manufacture of the ship. Nodes within each WBS can be related to nodes in another WBS and also related to the configuration items within the product structure.

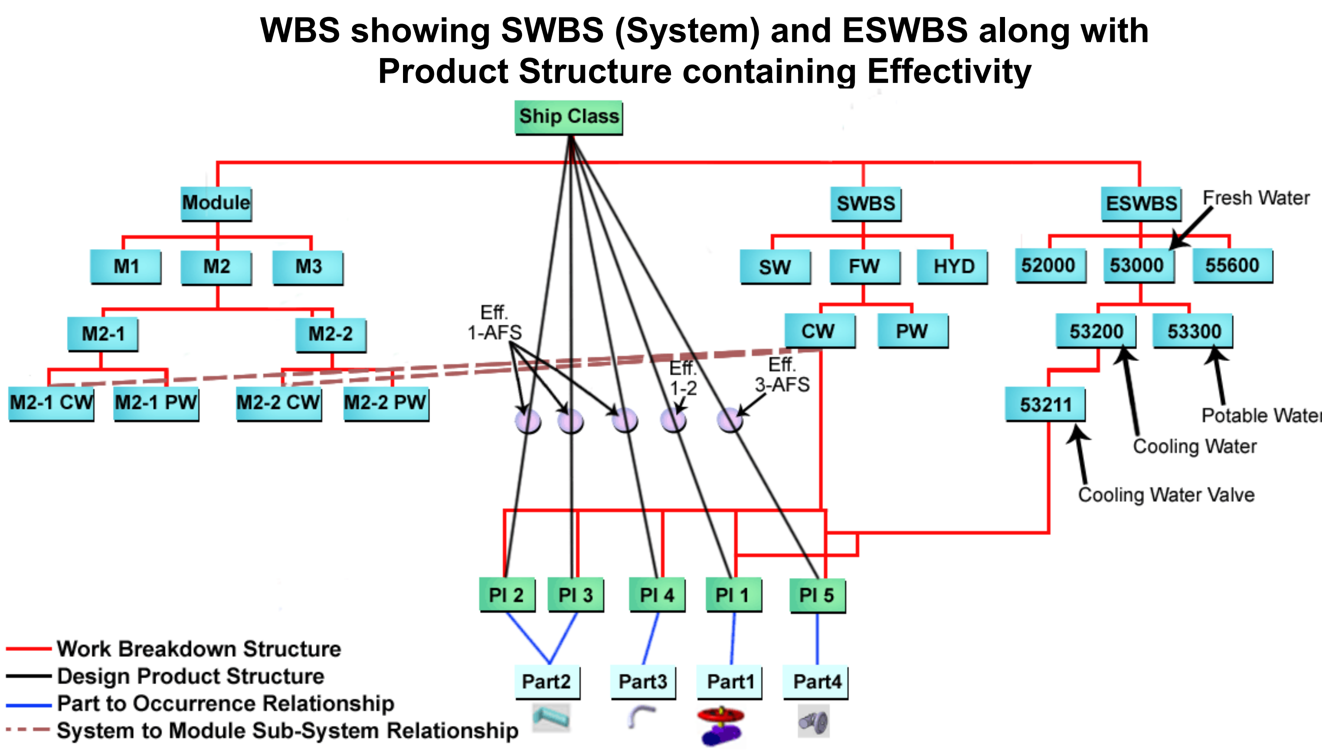


Figure 3-13 *WBS and Product Structure*

Each organization utilizing a given IPDE may have its own WBS. For example, the Logistics group can use the HSC and logistics breakdown structure, design can use a WBS by area or system, and manufacturing can have a WBS by module or ship assembly. These breakdown structures can be used to provide logical views of the data based on area, system, work authorization, task, module or most any other way required. Also, depending on implementation, some of the examples of WBS types may be incorporated as product structure. For example, one shipyard may define Compartment (Module) as part of the product structure hierarchy while another shipyard may define Zone (Areas) as part of the product structure hierarchy.

R-33: ***The IPDE shall provide the capability to define multiple work breakdown structures for navigating, viewing and accessing product structure data.***

3.2.1.13 Bills of Material (BOMs)

Bills of Material define a part list of components required to build a ship, with the top level representing the product or end item. BOMs are derived from the product structure and describe the as-designed (EBOM) or as-manufactured product.

R-34: The IPDE shall provide the capability to manage both the Engineering Bill of Material (EBOM) and Manufacturing Bill of Material (MBOM).

R-35: The IPDE shall provide the capability to maintain a link between the two BOMs specified in R-35 to enable changes in one to be reflected in the other.

R-36: The IPDE shall provide the capability to check for consumption of configuration items between two different BOM views.

R-37: The IPDE shall provide the capability to compare one or more BOM views.

3.2.2 Configuration Change Management

Configuration Change Management is the capability of the system to track and control all changes. It includes control of both changes and variances to a product using a systematic, measurable change process regardless of the type of product or phase of its life cycle. The configuration change management process includes:

- Identifying the need for a change
- Defining the change
- Documenting the change impact
- Evaluating and coordinating the proposed change
- Incorporating the approved change in the product and its related product configuration information
- Verifying change incorporation and continued consistency with the product configuration information.
- Identifying, documenting, approving, and implementing variances from baselined product requirements

R-38: The IPDE shall provide the capability to track change activity from initiation of the change through incorporation into the design and installation on the ship.

Figure 3-14 depicts an example of what happens when an authorized change is identified. The authorized change initiates the change to be made in the affected areas. In this case, both design and manufacturing have two areas affected by a particular change. On the design side, structural, piping and HVAC are affected under area 1. Within manufacturing, this change affects a system part in the M2 module in the product structure.

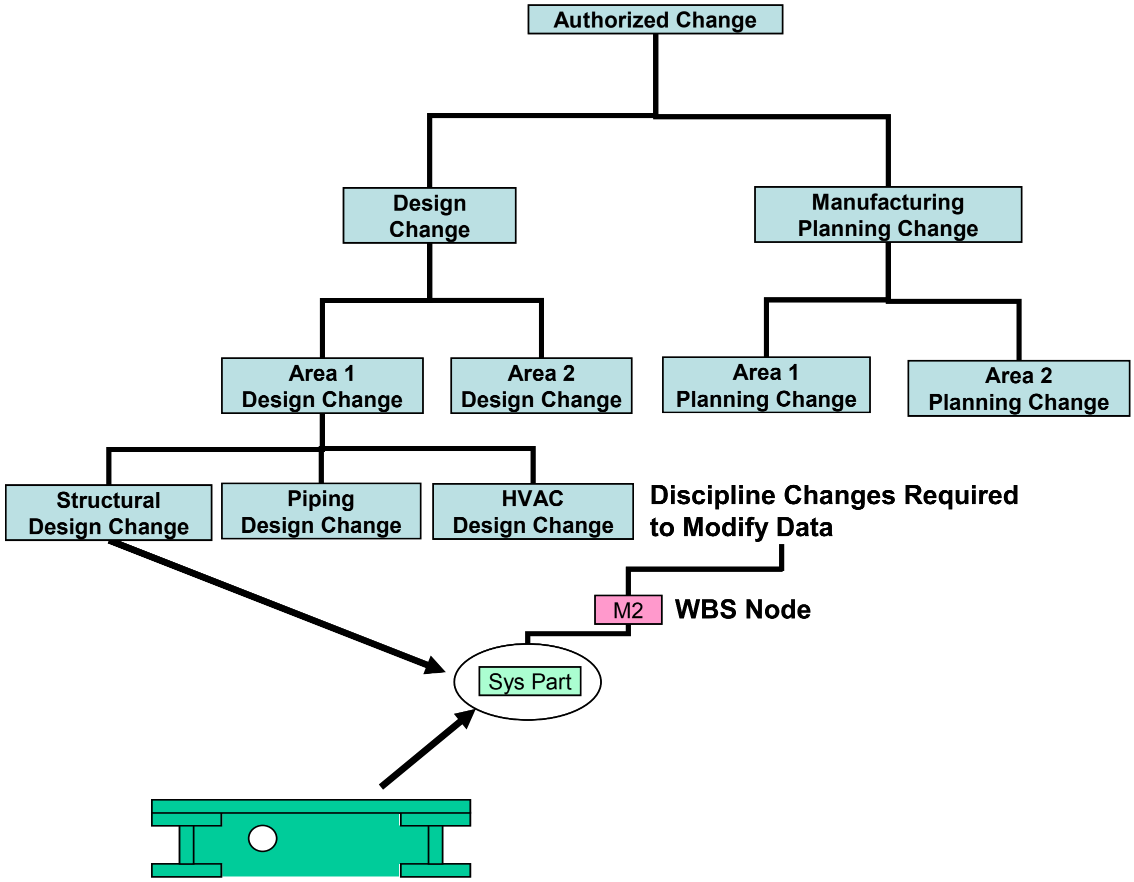


Figure 3-14 ***Change Management***

R-39: The IPDE shall provide the capability to manage the consistency between change effectivity and configuration item effectivity.

A change can only authorize updates to configuration items which are effective for one or more hulls for which the change is authorized to ensure hull effectivity configuration consistency.

R-40: The IPDE shall provide the capability to manage Part-Occurrence configuration item changes in a manner that minimizes the duplication of information.

Figure 3-15 illustrates configuration management of a Part-Occurrence, along with the hull effectivity of the Part-Occurrence configuration item to the hull configuration. It shows how occurrences are created once and applied to multiple hulls eliminating redundant work when configuration managing at the Part-Occurrence configuration item level.

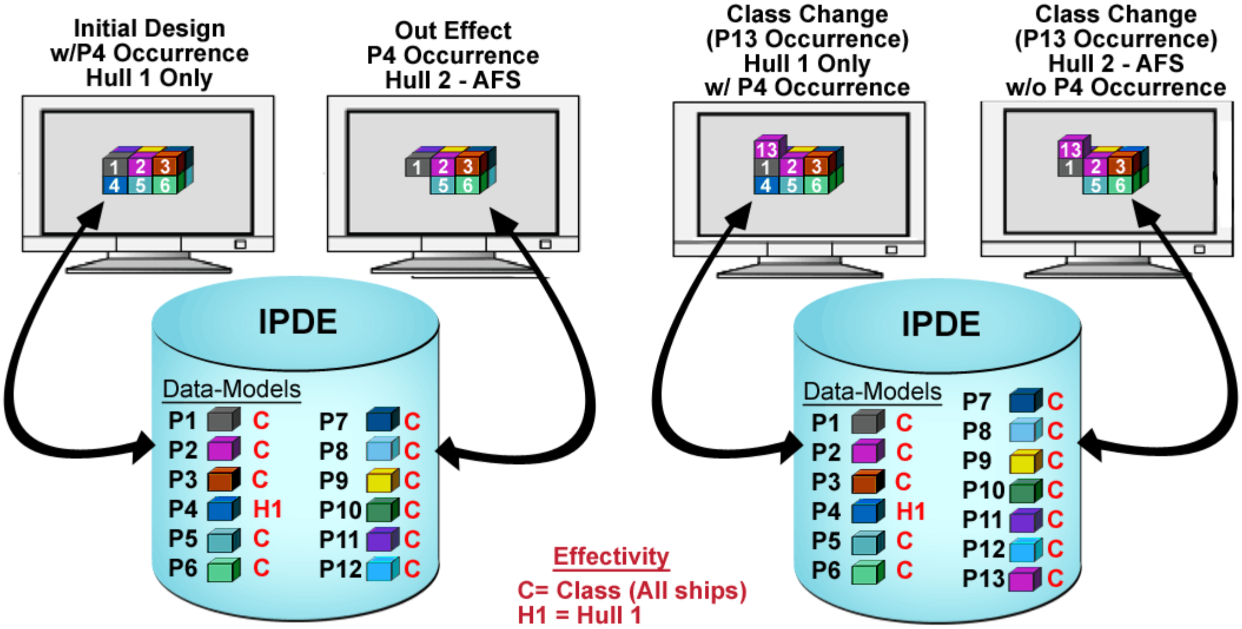


Figure 3-15 *Example of Part-Occurrence Configuration Change Management*

Part-based change management is described in more detail in Appendix A by walking through a configuration management scenario comparing part-based configuration management vs. document-based configuration management. Based on this scenario, it is expected that IPDE implementations will provide a mechanism to manage part configuration items in a manner that reduces the amount of duplicate data and minimizes the time required to make a change.

3.2.2.1 Change Notification

R-41: ***The IPDE shall provide the capability to notify appropriate organizations (e.g. vendors, subcontractors, engineering, logistics, production, etc.) or individuals of changes to configuration items.***

R-42: ***Change notifications shall include information necessary to uniquely identify the change, such as change number, description of the change, and information about the hull effectivity of the change.***

R-43: ***The IPDE shall provide the capability to send change notifications to external electronic mail systems.***

Figure 3-16 depicts an example of change notification. After the change to all of the subtasks beneath area 1 is complete, the affected manufacturing area is notified of this change electronically. Once notification is complete, the CM group can issue the change to manufacturing.

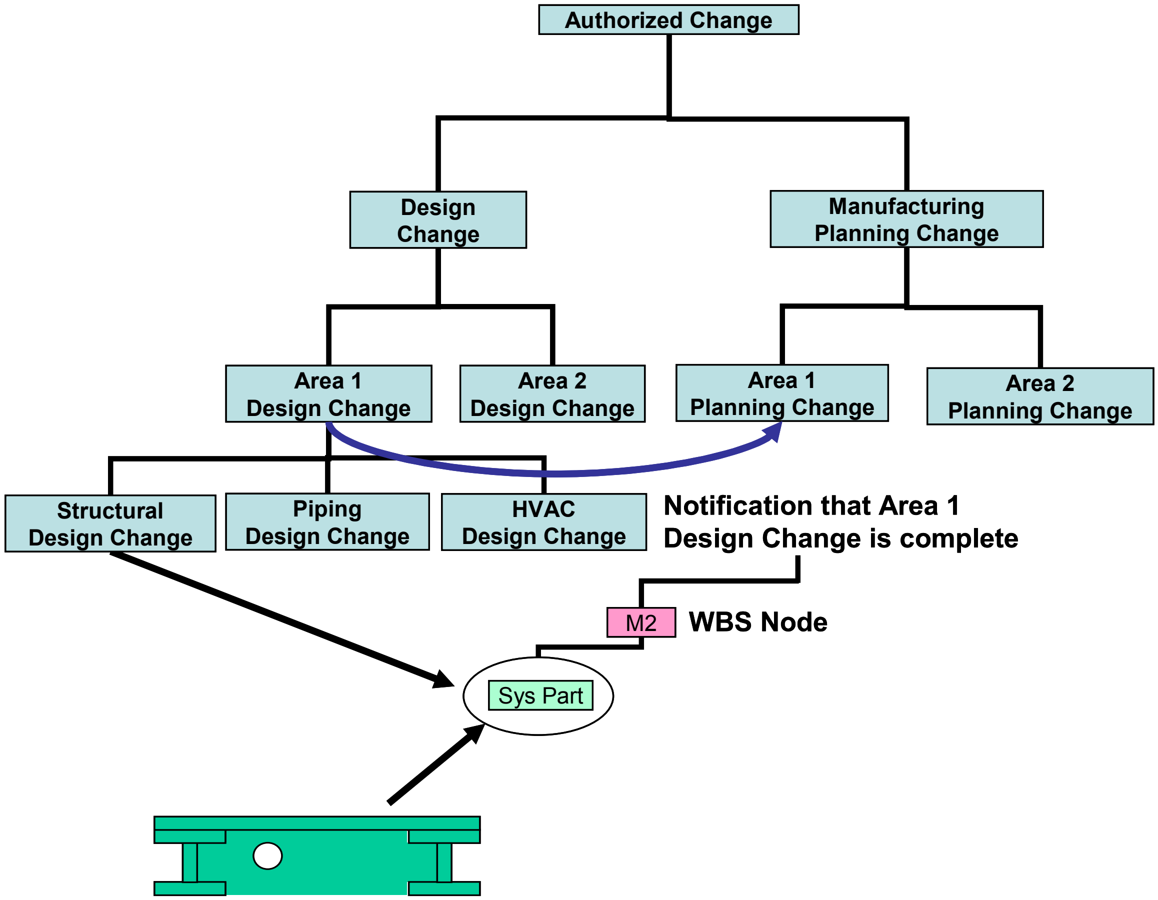


Figure 3-16 *Change Notification*

3.2.2.2 Change Tracking

R-44: ***The IPDE shall provide the capability to track changes throughout acquisition and life cycle.***

R-45: ***The IPDE shall provide the capability to manage change objects (e.g., change documents) for capturing change information. Change object attributes and relationships are defined in the SCIM.***

R-46: ***The IPDE shall provide the capability to associate change objects to various versions of the configuration items.***

R-47: ***At a minimum, change objects shall include the following information:***

- ***Unique change identifier***
- ***Originator organization and responsible individual***
- ***Change classification including priority/urgency***
- ***Product(s) affected by the change (including components and interfacing products)***
- ***Version of product(s) in which the change is implemented***

- ***Change description and scope, including effects on specified performance, operation, maintenance, servicing, operation and maintenance training, repair parts, support and test equipment, etc.***
- ***Product configuration information affected by the change***
- ***Reason/justification for the change***
- ***Proposed hull effectivity of the change***
- ***Estimated cost increase or savings***
- ***Change implementation and delivery dates***
- ***Criteria, if any, for retrofitting products***

R-48: The IPDE shall provide the capability to associate supporting documentation to the change object.

R-49: The IPDE shall provide the capability to classify changes to aid in determining the appropriate levels of reviews and approval.

One of the attributes of change that determines the amount of information and the approval level of the change decision is the change classification. Change classifications differentiate changes with significant impacts, such as impacts to the functional and physical interchangeability and supportability of the product, from changes that have little or no impact on those areas.

R-50: The IPDE shall provide the capability to place downstream dependencies on changes.

When there is a relationship between configuration items, there needs to be an explanation of what a change to one causes to the other. For example, if a Part-Occurrence configuration item is changed an associated document configuration item may also require a change. The system must allow such parent/child relationship dependencies to be defined when a change is involved.

R-51: The IPDE shall have the capability to route changes for review and approval and provide status of changes to the appropriate groups or individuals (e.g. new changes, changes waiting for approval, approved changes, closed changes etc.).

R-52: The IPDE shall provide the capability to manage back fit vs. forward fit of changes. In cases where a back fit alteration is accomplished first on a delivered product, that same alteration will then be applied to new construction.

Figure 3-17 depicts how changes are tracked once initiated. The figure shows a product structure relationship between the authorizing change and the affected areas. The change can be tracked from the top level down to the lowest and from the bottom level back up to the top level. The change is only considered complete once all of the tasks beneath it are complete. Before a subtask can be considered complete, the change object must go through a workflow for review and approval.

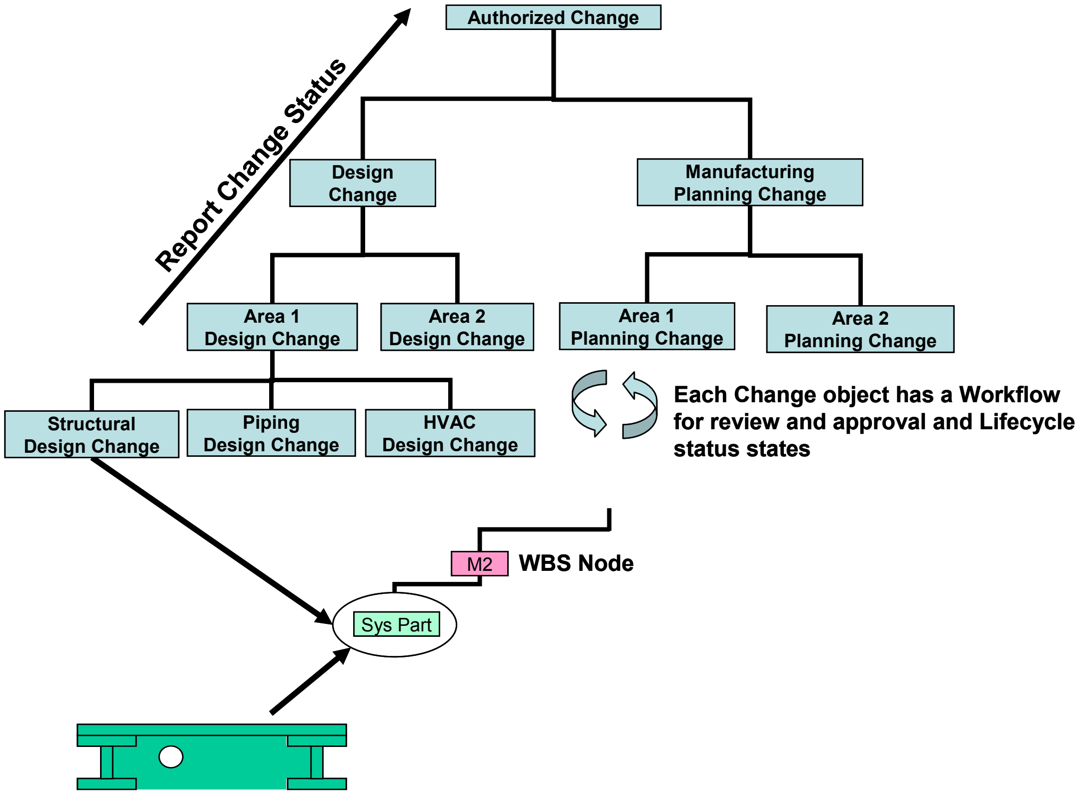


Figure 3-17 *Change Tracking*

3.2.2.3 Configuration States Dependency Management

The system must provide the capability to enforce rules on promotion of configuration items, such as the parent configuration item can not be promoted until all its children have been promoted to the target state (see section 3.2.1.4 for specific Configuration States requirements).

R-53: The IPDE shall provide the capability to enforce rules on promotion of configuration items.

This capability will prevent a configuration item from being prematurely marked as completed without its children being completed first.

R-54: The IPDE shall allow user-defined life cycle statuses for all configuration items.

This capability will most likely be program specific, functional area specific or will be specific to the item being statused. Examples of user-defined life cycle statues are: Working, Released, Archived, Issued, etc.

R-55: The IPDE shall use the rules defined by the access controls and permissions capability to also determine the routing of promotion as well.

This will ensure that the proper functional areas are able to review and status a configuration item according to the rules set up for their area.

3.2.2.4 Change Cause/Effect

The traceability of the causes/effects of changes throughout the product life cycle is extremely important in determining the sources of changes and their actual cost. A change in a system diagram may often trigger a series of changes in multiple detail design areas. These *caused* changes may have their own support organizations and documents that can co-exist outside of the boundaries of one organization. The caused changes are not necessarily discovered at the time of scoping the original change as the need for changes could be discovered during the production or maintenance phases. For example, when a detail design change is created, if it is a result of a system diagram change, it is linked to it. Similarly, the system diagram change object is also updated to indicate that it caused the detail design change.

The change object is a generic term which can be represented in a change record and it's supporting documents in an electronic form.

Change Cause/Effect Data is defined as the capability of a system to:

- Link Customer change orders to its internal change process
- Have an internal change process where the causes and effects of changes are identified and linked throughout the system

R-56: The IPDE shall provide the capability to define dependency relationships between changes and link them to change objects in the system.

R-57: The IPDE shall provide the capability to provide change dependency traceability throughout the product life cycle.

.

3.2.3 Configuration Status Accounting (CSA)

Configuration Status Accounting provides the visibility into status and configuration information concerning the product and its documentation for the purpose of providing status, providing current product configuration information, and measuring performance. All of the other configuration management activities provide information for status accounting as a by-product of transactions that take place as the functions are performed.

CSA provides accurate, timely information about a product and its associated product configuration information throughout the product life cycle. The goal of CSA is to capture and maintain product configuration information to ensure that current and historical configurations of the product and product configuration information can be accurately determined throughout the life cycle. The purpose of CSA is to ensure:

- That data about the product and the product configuration information is captured as the product evolves through its life cycle
- Retrieval of current, accurate information concerning change decisions, design changes, investigations of design problems, warranties, and shelf and operating life calculations
- Access to complete product configuration information
- Historical traceability of product configuration and product configuration information

CSA correlates, stores, maintains, and provides readily available views of an organized, indexed collection of product configuration information. Some of the information stored and maintained includes:

- Metadata related to product definition information (such as document identifiers, part numbers, changes installed in a given unit, and effective dates) with links to product definition information files
- Metadata related to product operational information (such as maintenance manual identifiers and operation check lists) with links to the product operational information files
- Metadata related to change objects, such as current status and approvals with links to appropriate documentation files

Decisions on the CSA data to be captured should be based on such factors as nature of the product, the environment in which the product will be operated, the anticipated volume and complexity of change activity, and the information needs of the customer. [1]

R-58: The IPDE shall provide the capability to capture, store and maintain configuration status information about the product and product configuration information throughout the product life cycle.

R-59: At a minimum, the IPDE shall provide the capability to capture, store and maintain the data defined in the latest revision of NAVSEA Technical Specification 9090-700, Ship Configuration and Logistics Support Information System (SCLSIS).

R-60: The IPDE shall provide the capability to validate and deliver configuration status information at various stages in the ship's life cycle to the Navy's Configuration Data Managers Database – Open Architecture (CDMD-OA) system in accordance with NAVSEA Technical Specification 9090-700.

CSA includes status reporting on configuration items and changes to configuration items. The following requirements identify the minimum recording and reporting requirements for CSA. These requirements are derived from MIL-HDBK-61A, Configuration Management Guidance, 7 February 2001.

R-61: The IPDE shall provide the capability to record and report the current approved configuration documentation and configuration identifiers associated with each configuration item.

R-62: The IPDE shall provide the capability to record and report the status of proposed engineering changes from initiation to final approval and contractual implementation.

R-63: The IPDE shall provide the capability to record and report the status of all critical and major Requests for Deviation that affect a configuration item.

R-64: The IPDE shall provide the capability to record and report the results of configuration audits to include the status and final disposition of identified discrepancies and action items.

R-65: The IPDE shall provide the capability to record and report implementation status of authorized changes.

R-66: The IPDE shall provide traceability of all changes from the original released configuration documentation of each configuration item.

[1] ANSI/EIA-649-A, National Consensus Standard for Configuration Management, 28 October 2004

R-67: The IPDE shall provide the capability to record and report the effectivity and installation status of configuration changes to all configuration items at all locations, including design, production, modification, retrofit and maintenance changes.

R-68: The IPDE shall provide the capability to record digital data file identifiers and document representations of each document and software that has been delivered, or made accessible electronically in support of the contract.

3.3 Product Data Access

This section details the core capabilities required to support accessing product data. This section includes Access Controls and Permissions, Product Model Navigation, Collaboration, Visualization and Reporting and Publishing.

3.3.1 Access Controls and Permissions

R-69: The IPDE shall provide the capability to restrict access to authorized internal users (within the shipyard or company), external users (outside the shipyard or company including the Navy, shipbuilding partners), to appropriate functions and data within the IPDE.

R-70: The IPDE shall provide the capability for only authorized users to access and modify data within the IPDE at the configuration item level, including access to specific attribute groups.

The criteria used to determine authorization will be dependent upon the individual programs. User privileges must be in context of organization, team, individual roles, projects, hulls, and phase of the project.

R-71: The IPDE shall provide the capability to allow multiple users to simultaneously update different areas of the same system.

R-72: The IPDE shall provide access control based on roles, which have associated rules and permissions.

R-73: The capability, specified by R-73, shall not be limited to one set of rules and permissions for each role. Depending upon the functional area and the phase of the product, the associated rules and permissions may be different for each role.

R-74: The IPDE shall provide the capability to enable access to update only the data required to perform the work on hand depending on the task and the role of the user assigned to that task.

This capability, in conjunction with the work authorization capability, will ensure that pertinent data is not erroneously overwritten, the proper user is performing the task, and that the job is being accomplished.

R-75: The IPDE shall provide the capability for the rules and permissions to support the lowest level of configuration items. If the lowest level of configuration item is part of a larger configuration item(s), the role of the user may vary between the two or more levels of configuration.

R-76: The IPDE shall provide the capability to allow user roles to support multiple companies and functional organizations. Each user role may have different grants and permissions within the companies and functional organizations.

R-77: The IPDE shall provide controlled access to proprietary and program sensitive information to prevent unauthorized exposure of this information.

This capability will ensure that only the users with a ‘need to know’ have access to certain data depending upon the role of the user and phase of the end product.

R-78: The IPDE shall support multiple access control administrators from multiple organizations within the security regulations of each organization.

Figure 3-18 shows an example of the permissions available on the part definition attributes. Different users are able to update the attribute information and/or update the status of the attribute group. A user, with appropriate permissions could change the status from In Work (IW) to Checked (CK) or Approved (AP). For example, the weights engineer does not have access to update any of the attributes for which the material rep is responsible.

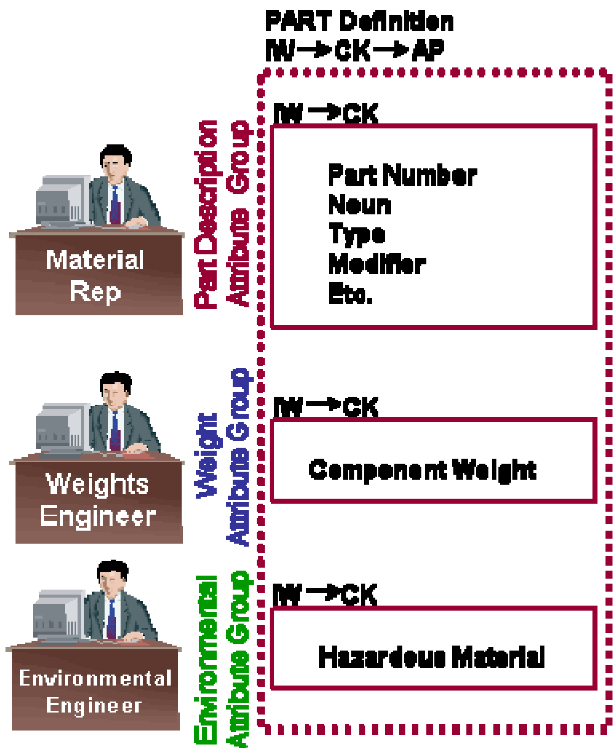


Figure 3-18 *Access Controls and Permissions*

3.3.2 Product Model Navigation

R-79: The IPDE shall provide the capability to dynamically assemble and display different views of the various configuration items. This viewing capability must be able to represent and organize different perspectives of the product model. The perspectives include, but are not limited to, system views, assembly views, and work breakdown views.

R-80: The IPDE shall provide the capability to search, query, and/or view any associated data at various levels of the WBS based upon the user’s criteria.

R-81: The IPDE shall provide the capability to display the configuration items associated with a particular system based upon hull effectivity.

R-82: The IPDE shall support the following views related to hull effectivity:

- ***Single hull view – showing all configuration items associated with the specified hull***

- ***Multiple hull view – showing all configuration items associated with the specified hulls***

R-83: ***The IPDE shall provide the capability to filter configuration item views by date, release status, issue status, and version.***

R-84: ***The IPDE shall provide the capability to view all versions or just the latest versions.***

R-85: ***The IPDE shall enable the user to navigate to associated program artifacts within each level of the WBS for various configurations.***

Examples of program artifacts can be:

- engineering documentation and reports
- BOM views
- cost and schedule control information
- logistics support data

R-86: ***The IPDE shall provide multiple options for navigation including textual, hierarchical, relational and graphical. The navigation options will depend upon the user's query and permission level.***

Figure 3-19 is an example of traversing the Work Breakdown Structure and product structure relationships. The WBS is a product oriented hierarchy that organizes and defines the hardware, software, services and related work tasks required for the products to be delivered (i.e. the work to be done). The top level of the WBS may list products or design phases, specifications, requirements, etc. The intermediate levels provide noun-name based navigable hierarchies of the design, typically organized by area, system and module for shipbuilding. The lower levels of the WBS represent noun-named based descriptions of discreet program level deliverables, activities, or services that are related to the Product Data. The dotted lines of Figure 3-19 depict the required capability to establish and traverse the relationships and dependencies that control product development (i.e. functional requirements, system to system dependencies, design drivers, etc.).

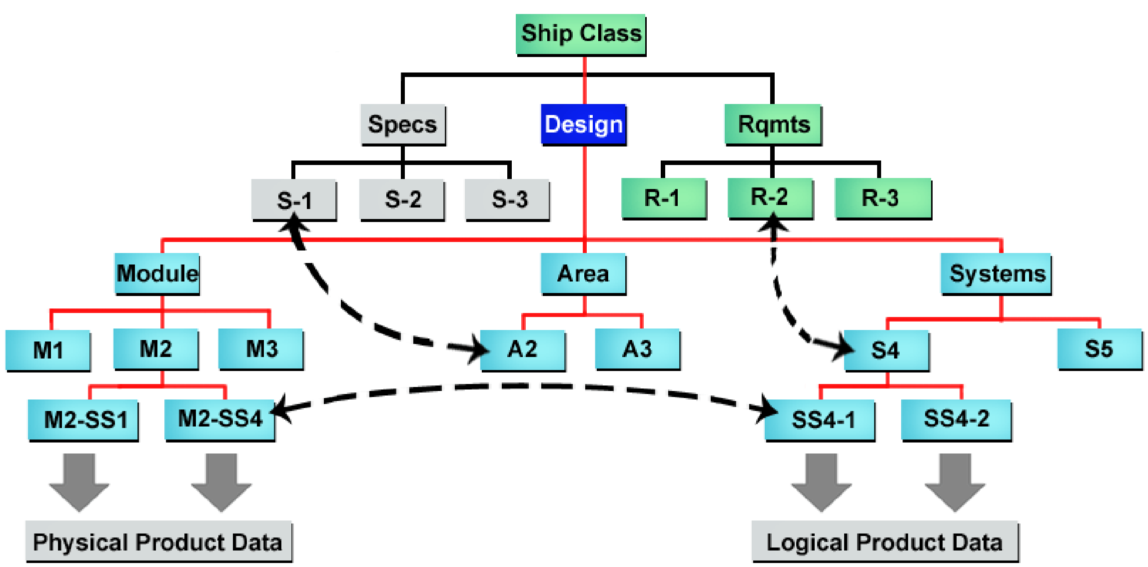


Figure 3-19 *Product Model Navigation*

3.3.3 Collaboration Tools

The IPDE must provide the capability for multiple users to be able to access the system to view and retrieve program data for which they have permissions from numerous remote sites. The system should be utilized for team collaboration to the maximum extent possible through the use of file and desktop sharing. Collaboration can be established in many ways depending on the data and the process by which the data is managed.

R-87: The IPDE shall provide the capability for all authorized users (e.g. stakeholders, teaming partners, customers, suppliers, etc.) to access the system to view, retrieve, and modify relevant program data from distributed sites.

For example, the IPDE can provide the customer access to integrated design/construction schedules including in-process drawings and changes. This will allow everyone to work in a real time “team” environment no matter the location of the key players. This capability will also allow models, drawings, etc. to be approved faster and in a timely manner.

R-88: The IPDE shall provide the capability to allow concurrent access to data.

This capability will allow multiple parties to work in the same area simultaneously and not cause clashes. This capability will also allow a user to resolve issues while all parties are looking at the same data at the same time.

R-89: The IPDE shall provide the capability to comment on configuration items and track those comments to resolution.

This capability will allow ease of comment resolution due to the fact that the necessary players can resolve the issues simultaneously. This capability will also allow the necessary players to see the status of the resolution and to whom the action belongs.

R-90: The IPDE shall provide the capability to share working files, accessible to anyone with the proper account and access privileges, regardless of geographic location.

R-91: The IPDE shall provide the capability to create desktop sharing sessions to facilitate meetings, design reviews, customer reviews, etc.

3.3.4 Visualization Tools

Visualization provides a mechanism to view the ship product model data and structure using 2D and 3D graphical representations integrated with tabular and other representations of the data. This is illustrated in Figure 3-20 which provides examples of various ship components and products being represented by 3D models and other graphical and tabular representations.

R-92: The IPDE shall provide the capability to visualize product model data

Examples of visualizations include:

- interferences, access space, flow paths, etc. using a 3D representation of the ship
- work packages and Bills of Material
- unconsumed parts in MBOM
- modeling and simulation results
- walk-through of virtual ship structure
- construction/test status

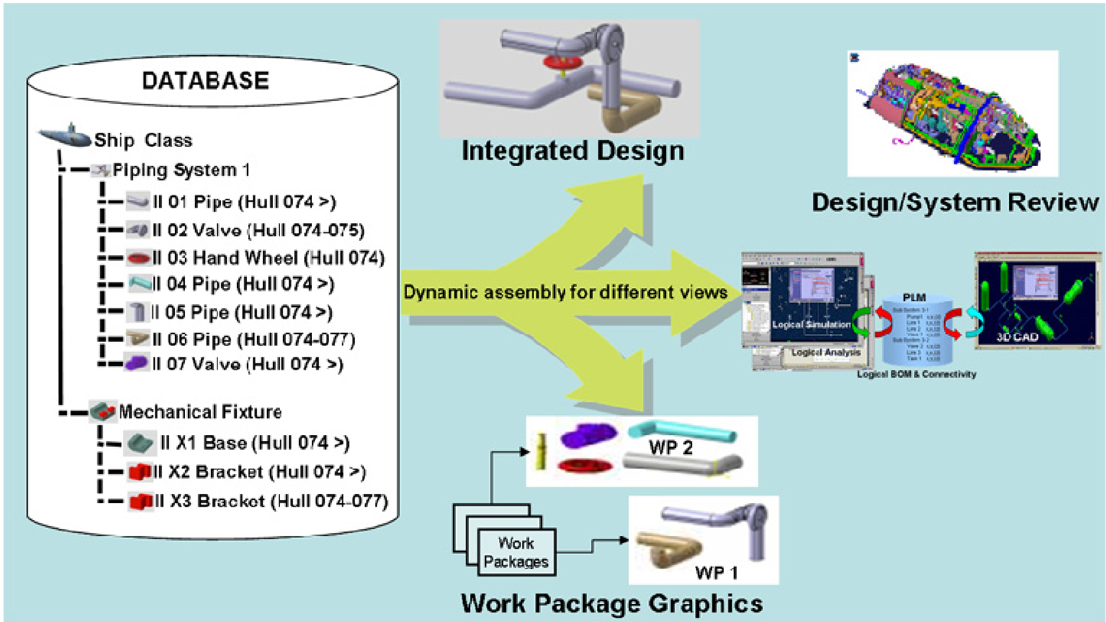


Figure 3-20 *Visualization*

R-93: The IPDE shall provide the capability to visualize the ship and its components at various levels of aggregation (e.g. parts, assemblies, sub-systems, systems, and ship structure).

R-94: The IPDE shall provide the capability to visualize the various levels by hull effectivity.

R-95: No matter at what level a shipyard chooses to configuration manage, the IPDE shall have the capability to view the data at that lowest level and in relation to the higher levels. These views can be custom or part of a standard set.

R-96: The IPDE shall provide the capability to view different configurations (e.g. As-Designed, As-Built, As-Maintained) for each hull.

The capability to view different configurations is necessary depending upon the current phase of evolution of the ship. Users will have the capability to see work package status, maintenance activity status and flow paths. Within the design sequences, users will also be able determine interferences before construction begins so the necessary changes can be made earlier rather than later.

R-97: The IPDE shall provide the capability to interrogate 2D or 3D models of a ship component and display associated attributes.

This will give the user the capability to see how the model fits within the system in which it exists and in relation to the ship (or top level of the WBS). The relation is not limited to the top level; it can be within any level of the WBS.

R-98: The IPDE shall also provide the capability to display design and/or installation status.

This will allow the user to see at what point either the design or construction sequence is. It will provide the user with an overall percent complete as well depending upon the select criteria (percent complete by assembly, system, overall ship etc). Having an electronic view of the installation sequence and status negates the need for a physical mock-up and allows users to do a virtual walk-through of a structure.

3.3.5 Reporting and Publishing Tools

R-99: The IPDE shall provide the capability to create reports on any data within the IPDE.

R-100: The IPDE shall provide the capability to produce reports in various formats (MS Office products, XML, HTML, flat files, etc.).

This reporting data will have to be defined on a case-by-case basis depending on the type of object, the type of data, and the reporting format desired. The reporting functionality needs to be flexible enough to support the need for creating ad-hoc reports for specified data sets.

R-101: The IPDE shall provide the capability to create ad-hoc and scheduled reports on product model data, product structure, WBS, and product data relationships.

R-102: The IPDE shall provide the capability to report on changes to the product model data.

This capability will enable a user to report on data relative to the particular job they are working. The user will also be able to create either hardcopy or electronic documents which depict the data in relation to the ship (top level of the WBS hierarchy) or any other tier within the WBS.

R-103: The IPDE shall provide the capability to publish manufacturing instructions, drawings, and other output products as defined by the program office.

R-104: If a hardcopy or electronic version of the drawing is required to support a team review or when the drawing is ready to issue, the IPDE shall provide the capability to produce the drawing in the required format.

R-105: The IPDE shall provide the capability to produce configuration item reports including the relationship to other configuration items.

For example, a report on all system configuration items effective to specific hull or all Part-Occurrence configuration items located in a specific compartment.

R-106: The IPDE shall provide the capability to record and report change information to the baseline configuration items.

R-107: The IPDE shall provide the capability to store reports in a manner and location easily accessible by the end user.

3.4 Business Process Control

This section details the core capabilities required to support business process control. This section includes Work Authorization, Requirements Tracking, and Configurable IPDE.

3.4.1 Work Authorization

Work authorization is required to ensure that a particular user has the proper authorization, access, and data to execute a unit of work. The user will have access to update only the data for which he/she is responsible. However, the user will have the capability to see how the data fits in relation to any level within the WBS. The work authorization functionality will also enable the review, approval, and closeout of changes and the routing of documents for review, comment, and approval.

R-108: The IPDE shall provide the capability to track work at the user level.

R-109: As new work becomes active or a change comes into effect, the IPDE shall provide the capability to identify the new tasking and assign it to the proper users.

R-110: The IPDE shall provide the capability to assign the proper resources (including the roles) required to perform the task.

This will enable ease of routing of the task and provide notification to the user that their effort is required on a particular task throughout the entire workflow of the task.

R-111: The IPDE shall provide the capability to assign hull effectivity to the work being accomplished. This hull effectivity needs to be within the range of the WBS node to which it is attached; it cannot fall outside of the range.

For example, if the task is to model a part of the chilled water system and the hull effectivity for the chilled water WBS node is for hulls 1-5, the task must have a hull effectivity within the range of hulls 1-5.

R-112: The IPDE shall provide the capability to manage the status of attribute groups. Management of attribute group status includes the capability to apply status (e.g. percent complete) to these groups as well as business rules for maturing status based on related attribute groups or other data.

R-113: The IPDE shall provide the capability to route documents through an approval cycle (including comment initiation and resolution).

This capability works along with the configuration states dependency management and access controls and permissions capabilities. A task will not advance to the next step in the workflow (depending upon the workflow and established dependencies) until the task is at a certain point. For example, a model will not be allowed to be completed until it has been through its review cycle. Also the designer working that model will not be able to review it; the review will have to be done by a user with a reviewer role.

3.4.2 Requirements Tracking

This section will define the process needed to interface with the requirements management tool used to capture and manage the contract requirements.

R-114: The IPDE shall provide the capability to establish a bi-directional link from the applicable configuration items managed in the IPDE back to the requirements.

R-115: The IPDE shall provide notification when a requirement linked to a configuration item has changed. This notification will be sent to all of the parties affected by the change.

This capability will enable the users to incorporate the changes in a timely manner and reduce the amount of rework which often results when the information is not properly dispersed to the appropriate parties. This capability will also allow early visibility to any design and/or production issues.

3.4.3 Configurable IPDE

The IPDE must provide the capability for multi-program support (i.e. multiple ship classes). This means the IPDE must be configurable to support the needs of any new program. This eliminates the need to create a new or separate IPDE for each new ship class program.

R-116: The IPDE shall provide the capability to support multiple programs. Specifically, this includes the following:

- ***The IPDE shall provide the capability to segregate program-specific information.***

- ***The IPDE shall provide the capability to restrict access to data that is proprietary to each program and/or vendor.***
- ***The IPDE shall provide the capability to reuse applicable data between multiple programs.***
- ***The IPDE shall provide the capability to manage a product model schema that is common to multiple programs within the IPDE.***
- ***The IPDE shall provide the capability for user-defined product model schema. This includes user-defined types, attributes, associations (references) among types, and dictionary/definitions.***
- ***The IPDE shall provide the capability to manage rules or configurable features that are common to multiple programs within the IPDE.***

R-117: The IPDE shall be configurable based on program-specific requirements. Specifically, this includes the following:

- ***The IPDE shall provide the capability to tailor and extend the common product model to support program-specific requirements.***
- ***The IPDE shall provide the capability to tailor and extend the common rules or configurable features to support program-specific requirements.***

4.0 INTEROPERABILITY REQUIREMENTS

For the purposes of this document, the definition of interoperability will be consistent with that defined by the Department of Defense:

> *The ability of systems, units, or forces to provide data, information, materiel, and services to and accept the same from other systems, units, or forces, and to use the data, information, materiel, and services so exchanged to enable them to operate effectively together.* (DoDD 5000.1)

There are two main elements of interoperability: technical and operational.[2] The technical aspects of interoperability deal mainly with the definition of data and interface specifications, along with the means with which two or more entities communicate. Operational interoperability is more procedural in nature; its aspects concern the ability of a system to enable effective and efficient collaboration. Although shipbuilding IPDEs need to enable interoperability for both to the greatest extent possible this section will focus on the technical aspects by discussing application integration and data exchange. Sections 4.1 and 4.2 will introduce the SCIM and identify requirements for its use with respect to data exchange. Section 4.3 will identify several aspects of open systems design that are applicable to shipbuilding IPDEs and that should be strongly considered for any IPDE implementation.

4.1 Ship Common Information Model (SCIM)

The ship product model needs to encompass the ship design; the catalogs of parts that support the design; the planning, manufacturing, and testing information to realize the design; the engineering models to analyze the design; and the logistics models to support and maintain the ship itself. It also spans the aforementioned life cycle stages and resides in a number of different (though interconnected) systems, applications, and data repositories. These various systems, while managing distinct aspects of the product, depend upon information in other portions of the IPDE, and the product model itself must behave as if it is managed on one single repository. This is necessary so that the items of the product model are available for re-use, and the IPDE must be able to maintain the proper associations between items and the data objects that define their properties, and between items and the other items they depend upon.

The Ship Common Information Model (SCIM) document defines the minimum information that must be maintained within an IPDE and the format needed to exchange it to enable effective interoperability among shipbuilding IPDEs.

R-118: For each application area specified in the SCIM that is supported by the IPDE, the IPDE shall at a minimum be able to maintain the product model data specified in the SCIM.

The main function of the SCIM is to specify the information that must be captured by the shipbuilding IPDEs. The ability to exchange this product model data efficiently using a neutral file format is a secondary requirement enabled by the SCIM.

It is intended to be used throughout a typical shipbuilding life cycle, and is designed to be invoked and mandated on future contracts. The SCIM is based on several existing STEP Application Protocol (AP) Standards (ISO 10303) and the schemas developed under the NSRP's Integrated Shipbuilding Environment (ISE) Projects for the various shipbuilding domains. SCIM-compliant files will be in XML format. Schema elements from these STEP standards, that

[2] Software Acquisition Gold Practice – Ensure Interoperability, http://www.goldpractices.com/practices/ei/index.php

are relevant to the shipbuilding domain, have been identified and organized as one coherent product model schema. The purpose is to define a single, common data vocabulary to be referenced by all future shipbuilding IPDEs for data exchange and application integration. Figure 4-1 demonstrates that the SCIM addresses application areas defined in several different STEP Application Protocols (APs).

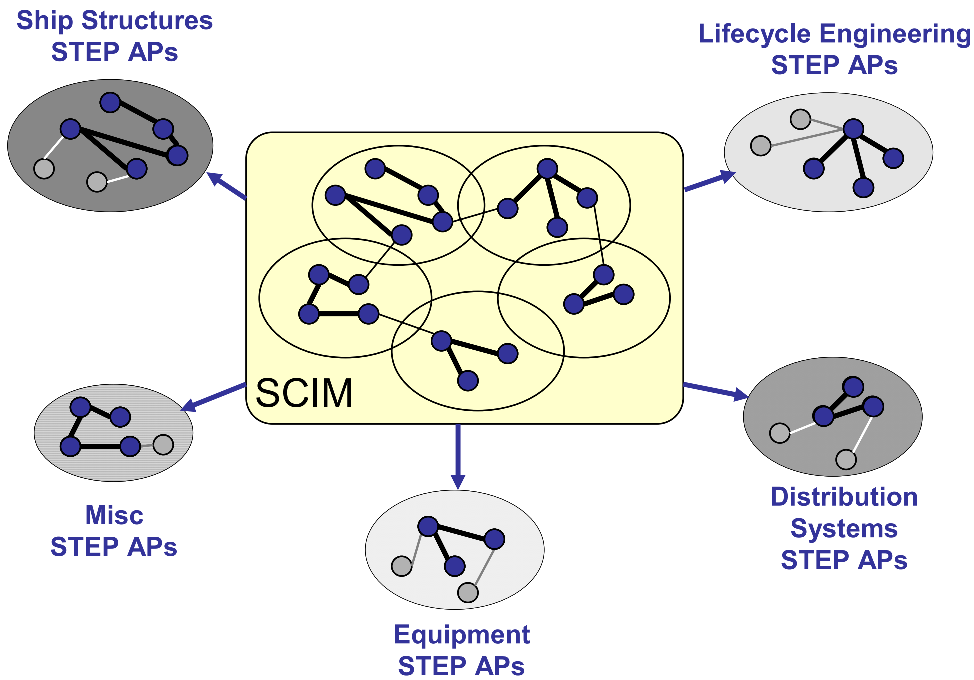


Figure 4-1 *SCIM Maps to Existing STEP Application Protocols*

The SCIM addresses the application areas that are of major importance for a shipbuilding product model. The following is a list of proposed functional areas covered by the SCIM:

- Product Data Management and Change Management Fundamentals
- Structural Moulded Forms (full ship model)
- Structural Arrangements
- Structural Detailed Design
- Piping Functional Design
- Piping Physical Design
- HVAC Functional Design
- HVAC Physical Design
- Electrotechnical Functional Design
- Electrotechnical Physical Design
- Common Parts: Procurement
- Product Life Cycle Support

One should examine the latest version of the SCIM directly in order to determine which functional areas are currently covered.

4.2 IPDE Interoperability Requirements

4.2.1 Product Data Accessibility

R-119: The IPDE shall have the capability to export, import, and query product data as defined by the SCIM.

R-120: The IPDE shall be able to import product data that is defined in the SCIM, and must be able to filter out any additional data within the SCIM that may not be required for a particular application.

R-121: The IPDE shall have standardized interface and query mechanisms, based on the SCIM data schema, to access Product Data. A user should have the ability to find and retrieve this product data efficiently, and multiple users should be able to access the system simultaneously.

R-122: The IPDE shall have version-controlled schema documents to describe any product data being integrated or interfaced within an IPDE or between one IPDE to another [OACE-DG 3.2.1.2.2 k, 3.2.1.5.2 d].

R-123: These schema documents shall be available to IPDE administrators and shall include sufficient references to data standards such as STEP for understandability.

R-124: Integrated systems that produce product data model content of any kind shall provide complete access to the product model data through a comprehensive, well-documented application programming interface (API).

R-125: The format of the supplied product model data shall not require vendor-specific software to interpret its content.

This enables development staff within individual shipyard environments to process product model data at their discretion with the possibility of reusing this processing capability for future programs.

4.2.2 Data Exchange

R-126: The IPDE shall be able to produce a SCIM-compliant (STEP Part 28) XML file to represent the product model data in each application area specified in the SCIM that is supported by the IPDE.

SCIM-compliant XML means that there exists a mapping to and from the XML schema defined by the SCIM such that there is no loss of information in a transformation. This mapping should be able to be defined completely using XPath/XSLT or other compatible technologies.

R-127: The IPDE shall be able to receive product model data in the same SCIM-compliant XML format for each of these application areas.

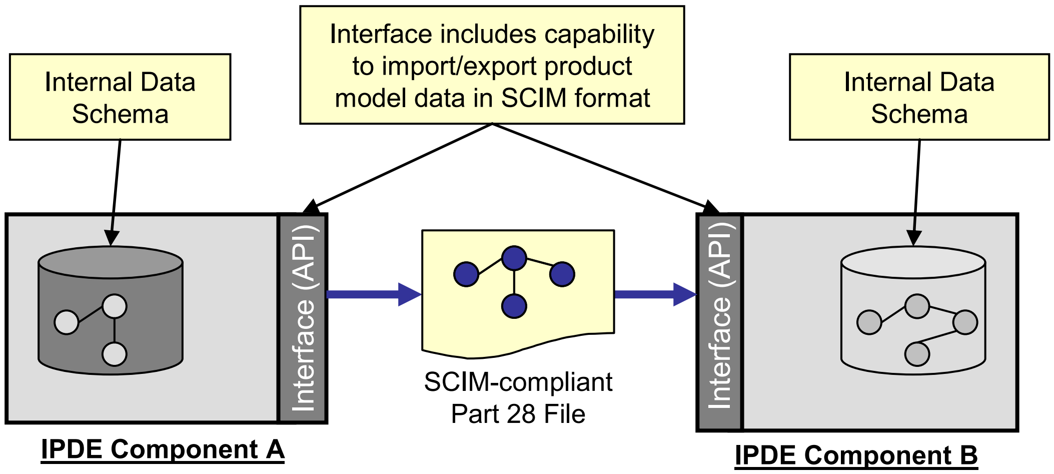


Figure 4-2 *SCIM-compliant STEP Part 28 Data Access/Exchange*

Since the SCIM does not specify a particular format, a SCIM-compliant exchange requires only that the XML file supports each of the data elements in the application area that is specified in the Ship Common Information Model. However, to be compliant with this specification, an IPDE must be able to produce this XML exchange file in a STEP Part 28 compliant format. Figure 4-2 illustrates data exchange between two IPDEs that support SCIM-compliant files in STEP Part 28 format. Although STEP Part 28 is not explicitly required to be used for all transfers, it is the assumed, preferred, neutral file format for these exchanges.

For data exchange, a validation process should verify that information is consistent and complete, conforms to the constraints of a given domain, and can be mapped to existing entities.

R-128: The IPDE shall have the ability to validate that the product data has been successfully transferred. Product data transfer validation shall include: format (schema validation), content (data validation), and an evaluation of error handling by the system.

It should be emphasized that the SCIM's primary function for the IPDE is to specify the information requirements that must be supported. The data exchange requirements discussed here are secondary issues and can be satisfied by various formats and/or mediators.

The ability to efficiently move data across applications in the same domain, applications within the same IPDE, or applications within separate IPDEs may require:

- Integration
- Data Sharing
- Migration
- Design in context
- Consistent semantics and format

To move data across applications in the same domain, direct translators must be developed between software packages. Alternatives include having both software packages access the same database (which is not usually feasible), or if the software packages can produce and read data in the same neutral file format (e.g. STEP), or having both systems support exchanges using SCIM-compliant XML files.

Transfer between domains (e.g. CAD to CAE or CAD to CAM) may require a direct translator or interface program. If neutral file exchange mechanisms (such as STEP or SCIM-compliant XML) are available, the difficulty of supporting many direct translators can be reduced.

For exchange between different IPDEs, direct or point-to-point translators are not generally a satisfactory solution because the transfer format used must be able to communicate with numerous other IPDEs which may be at the same shipyard, different shipyards, customers, team members, or suppliers. These exchanges require a neutral file format, but in addition one must specify the minimum required content of the product model in order to have a successful exchange.

These transfers should be capable of being accomplished via a SCIM exchange, but that is not a required mechanism as a given system may have developed an internal procedure that is more efficient than the neutral file mechanism specified by the SCIM. Note that exchanges using the SCIM are only one option for exchanges within a given IPDE.

Data exchanges can be accomplished by any of the following methods (in order of preference):

- Integrated access within one IPDE (data sharing)
- Exchange using standard neutral file formats (e.g. as defined by STEP)
- Exchange using SCIM-compliant XML files
- Development of direct run-time translators between applications or systems

4.2.3 Mediators

If the XML files supported by the IPDEs involved are not in a standard neutral file format (such as STEP Part 28) then Figure 4-2 does not apply and the SCIM exchange becomes more complicated. The actual representations themselves used by each vendor may in fact be different (different names, different distribution of properties among entities, etc.). A "Mediator" will be required to translate from the XML file generated by the sending IPDE to the XML format that can be read by the receiving IPDE. This is illustrated in Figure 4-3.

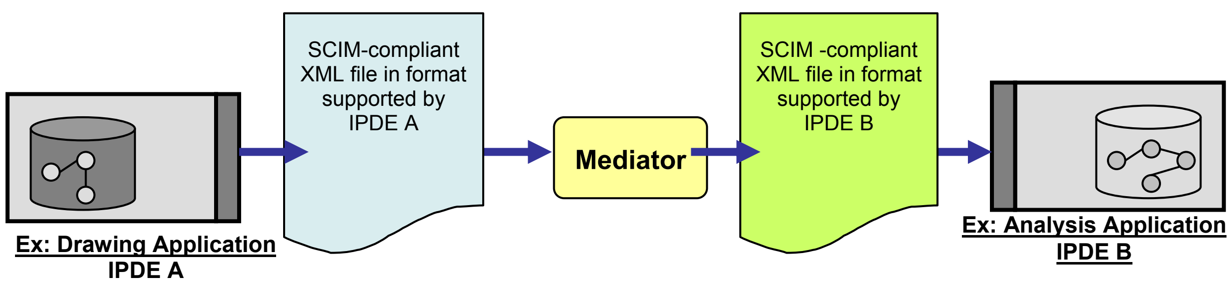


Figure 4-3 *SCIM-compliant Data Access/Exchange*

In fact, the file produced by the Mediator can be in STEP Part 28 format as opposed to a generic SCIM-compliant XML format. Thus, this "Mediator" technology can be used to support a standards-based neutral file exchange.

The Mediator is required for successful data exchange between the two IPDEs in the above figure, but it is not a deliverable required of either IPDE. Its development must be accomplished by the program, project, or shipyards involved.

However, development of the required Mediators can be facilitated by use of a "Mediator Generator" as illustrated in Figure 4-4 A "Mediator Generator" is a generic piece of software that reads the context schema and the XML Mapping Files for any two systems and creates a

"Mediator" between those mappings. A "Mediator", on the other hand, is an XSL stylesheet that reads a compliant XML file from the sending system and produces an XML file that can be imported into the receiving system. A "Mediator" can be written for an individual exchange, or can be developed automatically using a "Mediator Generator".

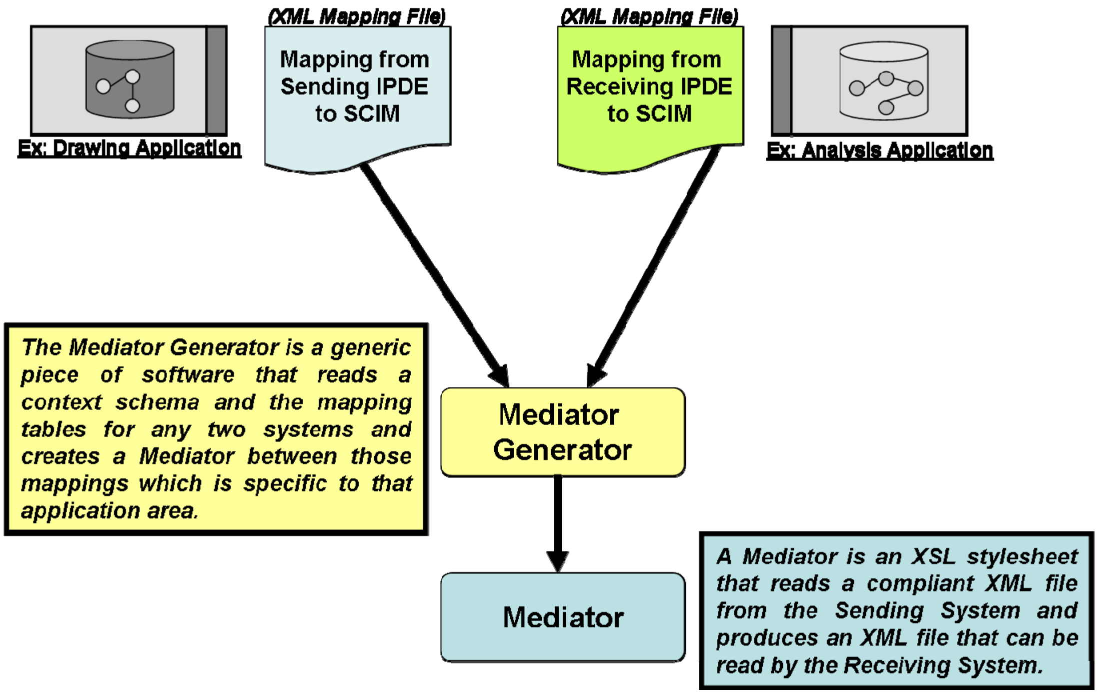


Figure 4-4 *Generation of Mediator Software*

4.2.4 Archive

Archiving of product models can be viewed as maintaining interoperability between systems over a long period of time. If the data must be available for the life of the ship, then its usefulness must outlive the software components and computer hardware with which it was created. Archiving the model in a neutral file format requires confidence that this format (and future revisions of the format) will be usable and viable for decades. To be approved as an archiving format for product models, the standard must:

- Capture Design Intent
- Preserve Construction History
- Include Geometric Dimensioning & Tolerancing
- Specify Analysis Results
- Produce an Efficient Format
- Maintain Upward Compatibility
- Provide Assurance of the Long Term Viability of the Standard

The version of the product model for an archive should be sufficiently documented to include the intent and proper use of the interface elements within it. This ensures that any future modifications to the corresponding specification do not adversely effect/influence the data within the archive.

Several of the above capabilities are under development as enhancements to STEP, but are not yet included in the International Standard (ISO 10303). It is hoped that eventually the STEP Standard can be invoked as a neutral file format to be used for archiving.

At the present time the SCIM models and the STEP Standard appear to provide the best mechanism for an archiving capability, but the requirements and needs for long term data retention have not been defined well enough to determine if these specifications will provide the necessary tools to satisfy this requirement.

4.3 Open System Compliance

4.3.1 Assumptions

It is assumed that any IPDE implementation will be constructed largely from integrated COTS-based and custom-built applications that serve the required functionality imposed by this specification and the needs of the environment in which it is deployed. At the time this document was written, predominant applications used in the shipbuilding lifecycle were developed and maintained by separate sources and based on disparate software architectures. Achieving the level of configurability that would enable efficient modification of IPDE functionality through 'swappable', standards-based components, where shipyards are afforded the freedom to choose from multiple vendor solutions, requires a common software architecture.[3] Defining an architecture at this level with associated requirements would require a considerable amount of effort and accordance from multiple disciplines. However, doing so has questionable benefit given that mandating such a framework may inadvertently exclude many systems vendors (if not all) from contention for future IPDE integrations, thus negating the value for such an extreme endeavor. Though the intent of this specification is not to exclude an environment at this level of maturity, the constraints set forth by it contents will not mandate it.

With that said it is crucial for future IPDE implementations to address the major cost drivers for the NPDI related to decreasing subsequent cost of change of the approved product model and increasing the ability of its constituent components to interoperate. Short of requiring a market where all application vendors implement a common infrastructure, there is still ample room for improvement in developing future IPDE implementations with comparatively little effort, but with substantial gains towards addressing these goals. It is therefore felt that it is in the best interest of future IPDE implementers to follow the guidelines specified by this section and in the referenced OACE – Design Guide.

This section defines architectural design guidelines consistent with Open Systems principles. The information contained here identifies those aspects of a component-based architecture that directly address interoperability and a system's ability to be efficiently maintained while handling modifications with minimal impact. Without delineating boundaries or defining the specifics of their interfaces this specification will assume that a candidate IPDE design will be composed of separate units of functionality (component, subsystem, system, etc.), which is a core structural element for an open architecture. Note that it is not within the scope of this specification to mandate any specific design methodology, system architecture, or development process. Nor is there any attempt to identify best practices with respect to systems engineering in general, although some aspects of it do have an impact on cost of change and interoperability.

This section borrows heavily from the Navy Open Architecture Computing Environment (OACE) Design Guide (DG). It includes references to specific content within the OACE-DG document to

[3] Refer to the definition of the OACE compliance level 4, which mandates the use of a common component architecture with standards-based interfaces and data exchange [OACE-DG, Section 3.1.5].

provide the reader with further related material when necessary. Although the OACE focuses on the design for real-time, mission-oriented systems, much of its content is applicable to an IPDE especially the material related to component design [OACE-DG, Section 3.2.1].

For the purposes of this section, the following definitions apply:

Component - A physical, replaceable software entity that performs defined functions and encapsulates data. A software component is a distinct, separable, and uniquely identified constituent part of a software system, with well-defined boundaries, capabilities, and interface to other software components. Both processes and libraries may qualify as software components [OACE-DG 3.2.1.2.1]. This term does not imply a specific implementation, programming language, technology, or granularity level.

Subsystem - A collection of components that have a degree of functional unity and utility that merit separate identification and tracking. The collections are smaller than the entire set that make up a software system [OACE-DG 3.2.1.2.1.1]. Examples of IPDE software subsystems include those that can be within a PDM application (part catalog management, presentation modules, etc.) or those that can be within any other large applications (e.g., a large 3rd party or legacy system).

System - An entire set of components aggregated for an overall purpose [OACE-DG 3.2.1.2.1.1]. Examples of IPDE software systems include a complete PDM application, a 3rd party application, or a legacy system. Within an IPDE, there are typically many software systems: one or more PDM systems, numerous 3rd party applications, and other software systems.

4.3.2 IPDE System Architecture Design Guidelines

The sections that follow identify guidelines that should be considered in the design of an IPDE. Figure 4-5 is a high level view of an IPDE system showing various levels of a software architecture - components, subsystems, systems - collectively referred to as units of functionality. Units of functionality, as defined in this section, refer to architectural elements that are likely to be updated or replaced in their entirety over time. Each unit of functionality has its own software interface (API) and is integrated with other internal and/or external units of functionality. Software elements that are self-contained but tightly bound to other software elements by unique interface or specific functional requirements are excluded. The goal of future IPDE architectures is to evolve towards an environment that partitions like functionality into self-contained software elements with the characteristics mentioned below.

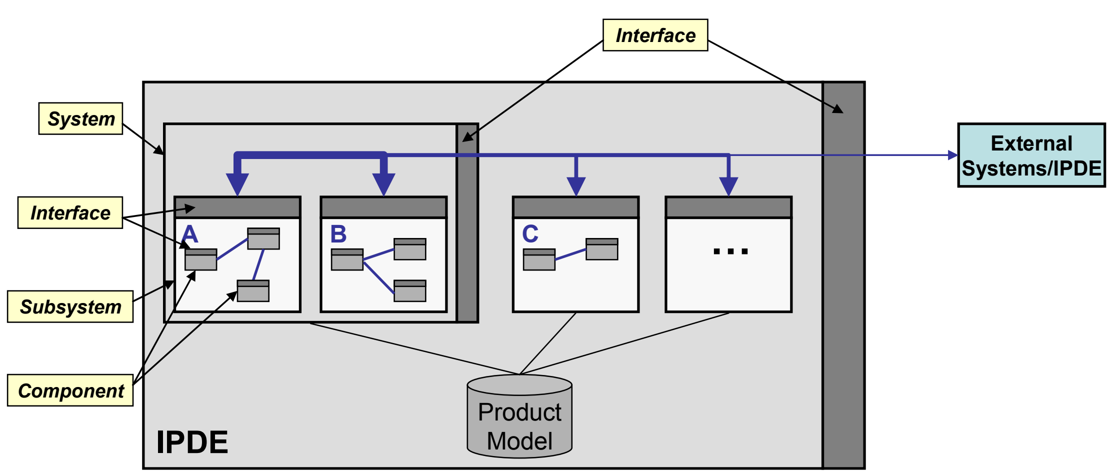


Figure 4-5 *IPDE Units of Functionality*

4.3.2.1 Functional Unity/Containment

At the lowest level each unit of functionality should be designed such that the functionality that it provides is as specific and logically coherent as possible. Doing so allows the designer to isolate pockets of capability to be used repeatedly and consistently throughout the system. Duplicate functionality unnecessarily bloats the size of an application and makes replacing it more tedious. Each unit of functionality should have stable interface specifications that remain consistent over time.

4.3.2.2 Loose External Coupling and Data Reliance

Each unit of functionality should be designed so that it limits reliance on external components for functionality and/or data. Any external communication should use well-defined interface entry points using explicit message passing. Low cohesion in this regard decreases the effort required to swap an element of an IPDE's capability and enables shipyards to achieve incremental improvements as requirements and technology evolve.

4.3.2.3 Interface Specifications

Each unit of functionality should include as part of its interface specification: a list of high-level functionalities, out-going dependencies (dependencies to other subsystems), and the current set of in-coming dependencies (other subsystems depending on it) [OACE-DG 3.2.1.2.1.2]. This allows IPDE management to reduce the risk and cost of incremental improvements and to enable these improvements to be spread across multiple programs and yards.

4.3.2.4 Use of public, consensus-based standards

Wherever practical, data elements exchanged as part of a public interface for a unit of functionality should use standards-based formats with format specifications that are publicly-available. In lieu of this, data elements should be able to be converted to a human-readable, self-describing format (such as XML). Doing so allows for increased interoperability because of the likelihood of cross-vendor support for common data formats.

4.3.2.5 Portability

IPDE implementations that incorporate COTS functionality should limit dependence on vendor-specific capabilities. Unique features that are tightly engrained in an integrated system and that become embedded in a yard's process are difficult to back fit should the application need to be replaced. If possible, it is preferable to segregate this functionality using software adapters with standard's based interfaces so that the underlying software can be retained.

4.3.2.6 Scalability

Integrated systems should be designed in a way that they can accommodate additional functionality (either through system upgrade, bug fixes, or security patches) and number of users given sufficient hardware support. This provides flexibility in the use of a system on various size programs with various user and functional requirements.

Appendix A. Configuration Management Scenario

This appendix illustrates the need to have part-based configuration management as opposed to document-based configuration management. The example used to illustrate the problem is a simplification of a small portion of the fuel oil and transfer system of the TWR 841. The Torpedo Weapons Retriever was selected because the drawings, product data, and associated data are freely available and Approved For Public Release with Unlimited Distribution.

This scenario examines a single pipe run (Figure A - 1) as it evolves over several years in a class of eight ships. In order to explain the issues a relatively simple example was necessary. This example is limited to the product model of the pipe run and does not show collateral impact on other systems, components, or spaces. Nor does it show the impact on items derived from the product model such as analyses, work packages, and manufacturing instructions. Each change is implemented as soon as it is accepted. There is no consideration of the workflow or approval process. The collaborative relationship between disciplines and between shops is not considered either.

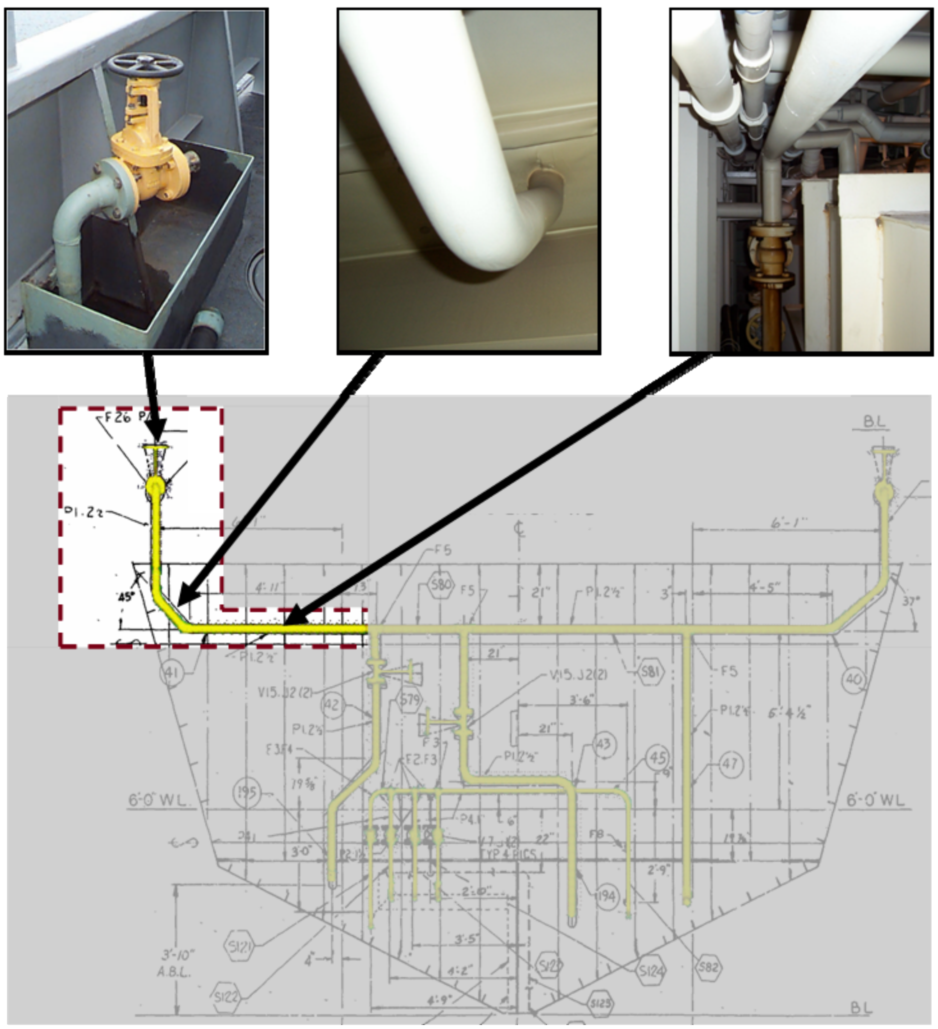
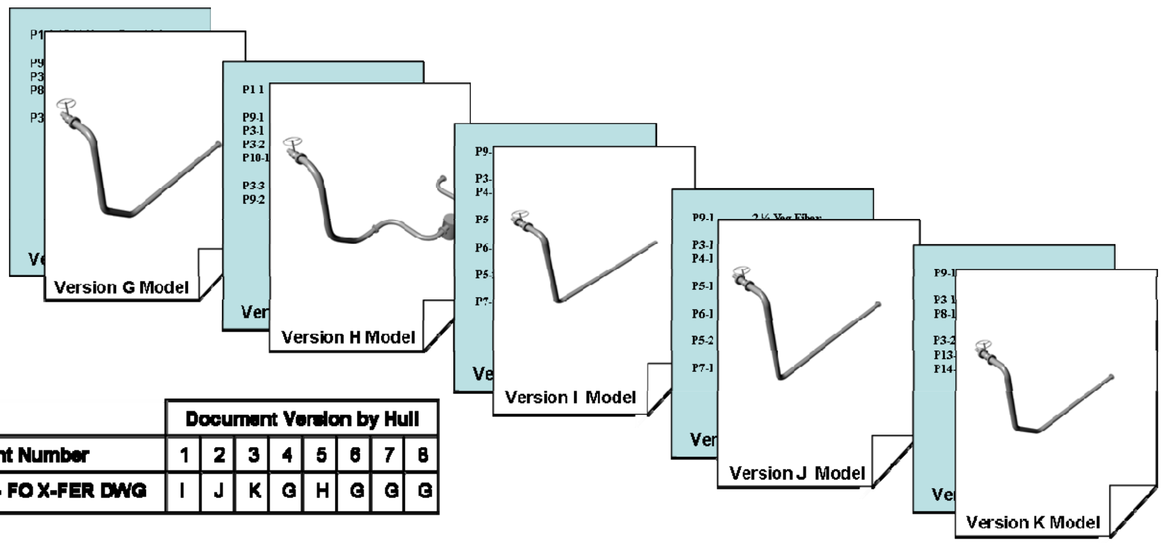


Figure A - 1 Configuration Management Scenario: TWR 841 Fuel Oil and Transfer System Pipe Run

The following terms have been defined for the purposes of this presentation: A part is an item that is defined and managed by a catalog. A part occurrence is a part that has been placed in the model. The terms are used interchangeably in this presentation. Configuration Management at the part level is referred to as part-based. Configuration Management at a level of a group of parts is referred to as document-based. Typically document-based implies a file based management system and part-based implies a file independent database driven management system.

In the document-based approach (Figure A - 2), multiple parts are configuration managed in a single document. This is the traditional approach to configuration management in Design/Engineering systems. Each unique configuration is represented in a separate document. A change that applies to more than one hull must be duplicated in multiple documents. This requires more labor to make a change to the class because each change must be repeated in the various document versions. A mapping of document versions to applicable hulls must be used to determine which document corresponds to each hull configuration.



	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	I	J	K	G	H	G	G	G

Figure A - 2 Document-based Approach to Configuration Management

In the part-based approach each part is configuration managed separately which eliminates the need to manage duplicate data.

While the part-based approach has been used for configuration management in Execution Systems for Ship Construction, it typically has not been commercially available in systems used for Design and Engineering. The part-based approach utilizes hull effectivity to identify which parts make up a particular hull configuration and allows the operator to dynamically build the necessary views.

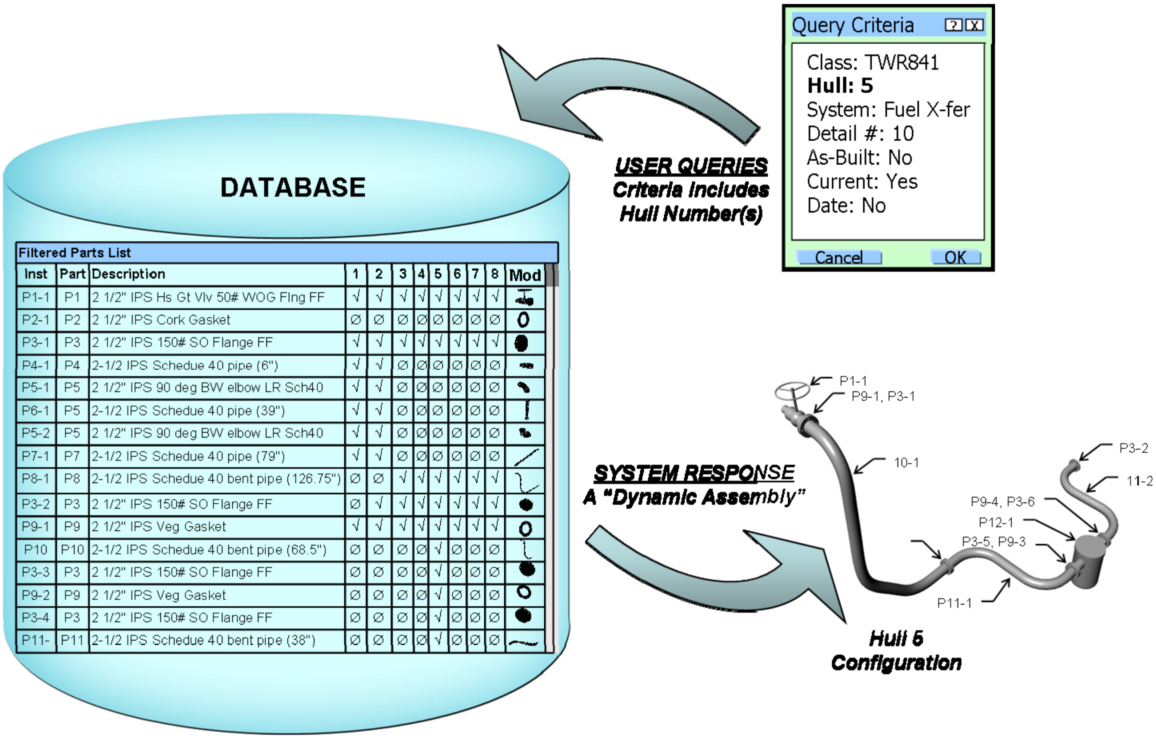


Figure A - 3 illustrates the part-based approach. If a user wants to work on the current hull 1 configuration, the operator defines a query, which will be run against the database to extract the necessary parts from the database and the system dynamically displays the appropriate version of the pipe detail for hull 1.

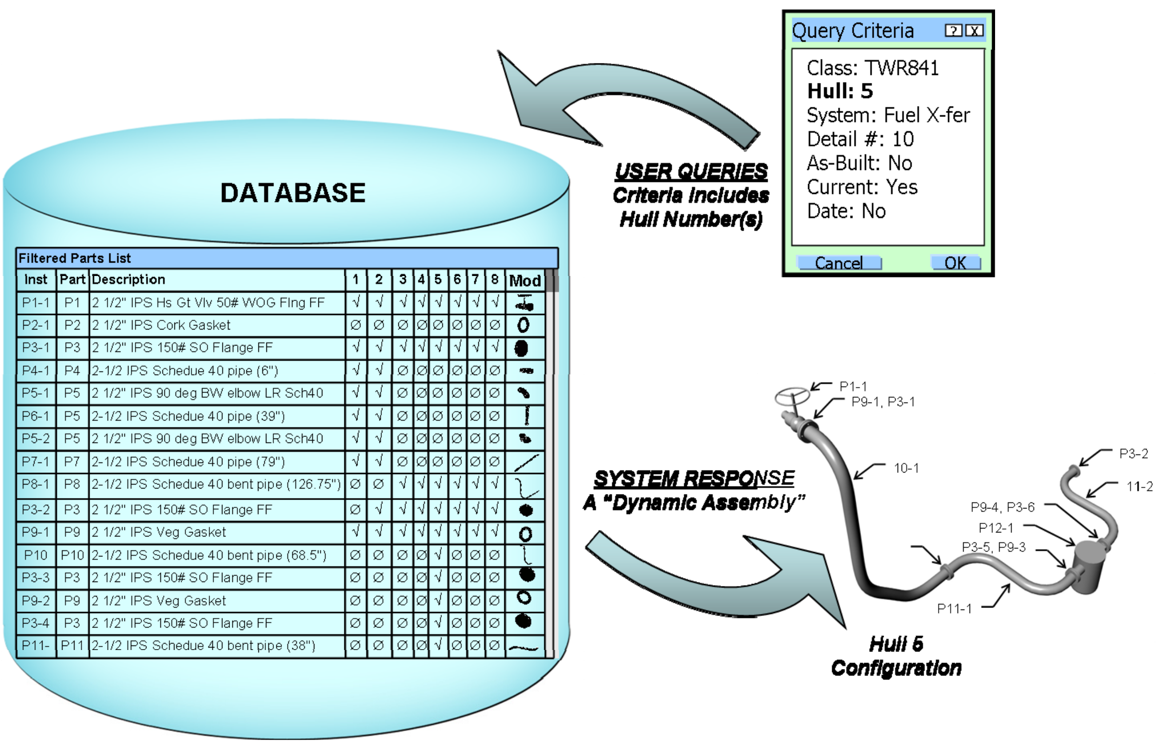
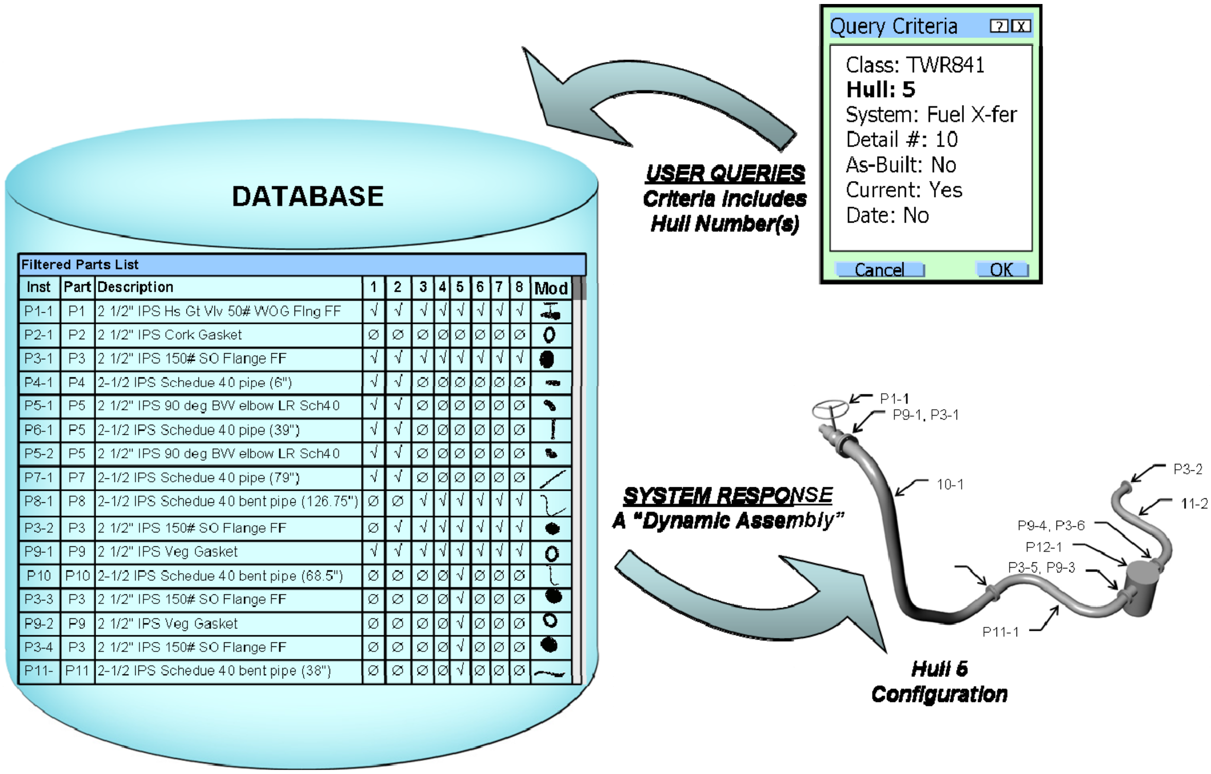


Figure A - 3 Part-based Approach to Configuration Management

If the user wanted to see the current hull 5 configuration, the operator would modify the query to reflect the appropriate hull (Figure A - 4). The query would be run against the database to extract the necessary part occurrences from the database and the system dynamically displays the appropriate version of the pipe detail for hull 5.



Inst	Part	Description	1	2	3	4	5	6	7	8	Mod
P1-1	P1	2 1/2" IPS Hs Gt Vlv 50# WOG Flng FF	√	√	√	√	√	√	√	√	
P2-1	P2	2 1/2" IPS Cork Gasket	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	
P3-1	P3	2 1/2" IPS 150# SO Flange FF	√	√	√	√	√	√	√	√	
P4-1	P4	2-1/2 IPS Schedue 40 pipe (6")	√	√	Ø	Ø	Ø	Ø	Ø	Ø	
P5-1	P5	2 1/2" IPS 90 deg BW elbow LR Sch40	√	√	Ø	Ø	Ø	Ø	Ø	Ø	
P6-1	P5	2-1/2 IPS Schedue 40 pipe (39")	√	√	Ø	Ø	Ø	Ø	Ø	Ø	
P5-2	P5	2 1/2" IPS 90 deg BW elbow LR Sch40	√	√	Ø	Ø	Ø	Ø	Ø	Ø	
P7-1	P7	2-1/2 IPS Schedue 40 pipe (79")	√	√	Ø	Ø	Ø	Ø	Ø	Ø	
P8-1	P8	2-1/2 IPS Schedue 40 bent pipe (126.75")	Ø	Ø	√	√	√	√	√	√	
P3-2	P3	2 1/2" IPS 150# SO Flange FF	Ø	√	√	√	√	√	√	√	
P9-1	P9	2 1/2" IPS Veg Gasket	√	√	√	√	√	√	√	√	
P10	P10	2-1/2 IPS Schedue 40 bent pipe (68.5")	Ø	Ø	Ø	Ø	√	Ø	Ø	Ø	
P3-3	P3	2 1/2" IPS 150# SO Flange FF	Ø	Ø	Ø	Ø	√	Ø	Ø	Ø	
P9-2	P9	2 1/2" IPS Veg Gasket	Ø	Ø	Ø	Ø	√	Ø	Ø	Ø	
P3-4	P3	2 1/2" IPS 150# SO Flange FF	Ø	Ø	Ø	Ø	√	Ø	Ø	Ø	
P11-	P11	2-1/2 IPS Schedue 40 bent pipe (38")	Ø	Ø	Ø	Ø	√	Ø	Ø	Ø	

Figure A - 4 Query for Hull 5 Configuration Using Part-based Approach

Figure A - 5 illustrates when the various ships in our example are built and modified as well as what changes are applied to each. The details of the changes will be described over the next several paragraphs. In the first year, hulls 1 and 2 are built to the initial design. In the second year, change number 1 is introduced and applied to hulls 3 and 4 during the construction phase. In year 3, change number 2 is introduced and applied to hulls 5 and 6 during the construction phase and applied to Hulls 2, 3 and 4 as a SHIPALT. In year 4, change number 3 which is a class change is applied to hulls 7 and 8 during the construction phase and all the other hulls as a SHIPALT. In year 5, change number 4 is applied as a SHIPALT to hull number 5 only. In year 8, change number 5 is applied as a SHIPALT to hulls 1, 2 and 3.

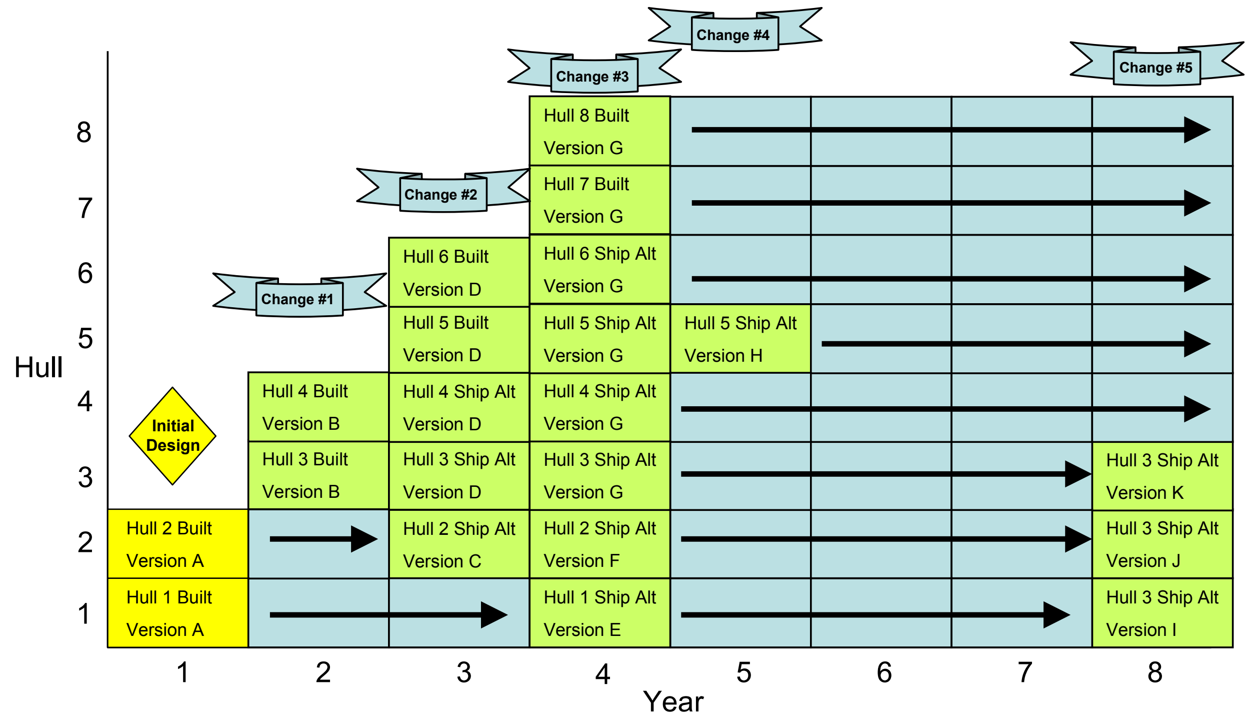


Figure A - 5 Change Applicability across 8 Hull Scenario

In the first year, a pipe run in the fuel oil transfer system is designed using a hose gate valve, straight pipe pieces, and elbows. This pipe run will be butt welded to an adjacent pipe run. At the end of the first year the class design has been completed and the first two ships are built and designated hull 1 and hull 2.

In the second year a design change is made to address a space constraint problem (Figure A - 6). A new construction process allows the pipe pieces and elbows to be replaced by a piece of bent pipe. Replacing those parts with a single bent pipe will reduce the cost to construct the system because fewer welds are required. This change is made in time to be incorporated into hull 3 and hull 4, both of which are currently under construction. This change will also be incorporated into all follow ships. Hull 1 and hull 2 will not be retrofitted with this design change. At the end of the second year hull 1 and hull 2 conform to version A, hull 3 and hull 4 conform to version B.

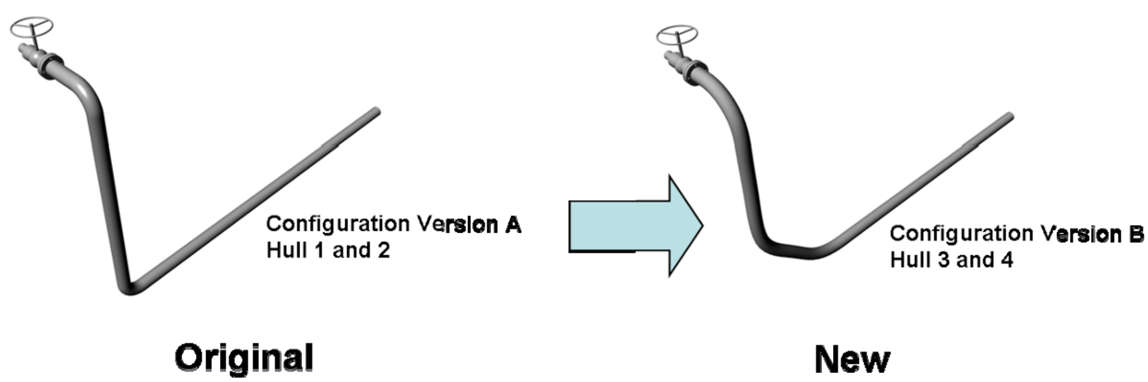


Figure A - 6 Configuration Management Scenario Change Number 1

During the third year a change is made to an adjacent pipe run (Figure A - 7). In order to accommodate the change a decision is made to replace the butt welded joint with a slip-on flange. A decision is made to apply this change to hulls 2, 3, and 4, and to incorporate it into the design of all future ships. The change will not be made to hull 1. A SHIPALT is required to change hulls 2, 3, and 4, and an engineering change order is proposed for hulls 5 and 6, the two ships currently under construction. Configuration management is now getting a little trickier.

The flange needs to be added to the version A configuration to accommodate the change in hull 2.

The flange needs to be added to the version B configuration to accommodate the changes in hulls 3 and 4 and all follow ships.

As a result, there are two new versions (C and D), version A no longer applies to the as-maintained hull 2, but is still valid as the as-built representation (a lead yard responsibility).

Likewise, version B no longer applies to the as-maintained hulls 3 and 4, but is still valid as the as-built representation for these hulls.

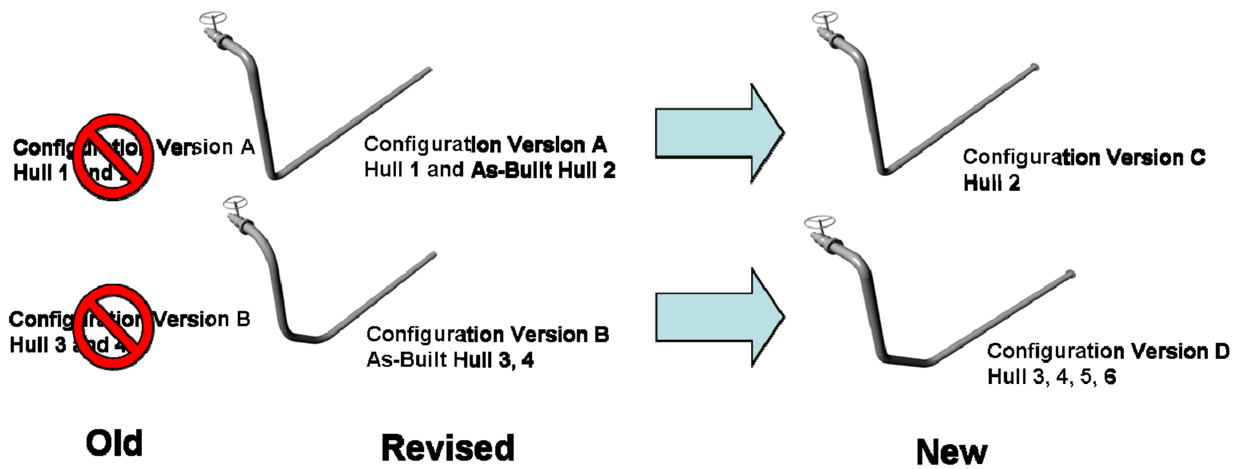


Figure A - 7 Configuration Management Scenario Change Number 2

During the fourth year, a class change is made which is applied to all hulls (Figure A - 8). The change involves replacing a gasket. Hulls 1 through 6 require ship alts while hulls 7 and 8 are built to the new class design.

The gasket needs to be replaced in the version A configuration to accommodate the change in hull 1. The gasket needs to be replaced in the version C configuration to accommodate the change in hull 2. The gasket needs to be replaced in the version D configuration to accommodate the change in hulls 3, 4, 5 and 6. This version also is used to build hulls 7 and 8.

As a result, there are three new versions (E, F, and G). Version A no longer applies to the as-maintained hull 1, but is still valid as the as-built representation for hulls 1 and 2. Likewise, version C no longer applies to hull 2 and is obsolete. Version D no longer applies to hulls 3, 4, 5 and 6, but is still valid as the as-built representation for hulls 5 and 6.

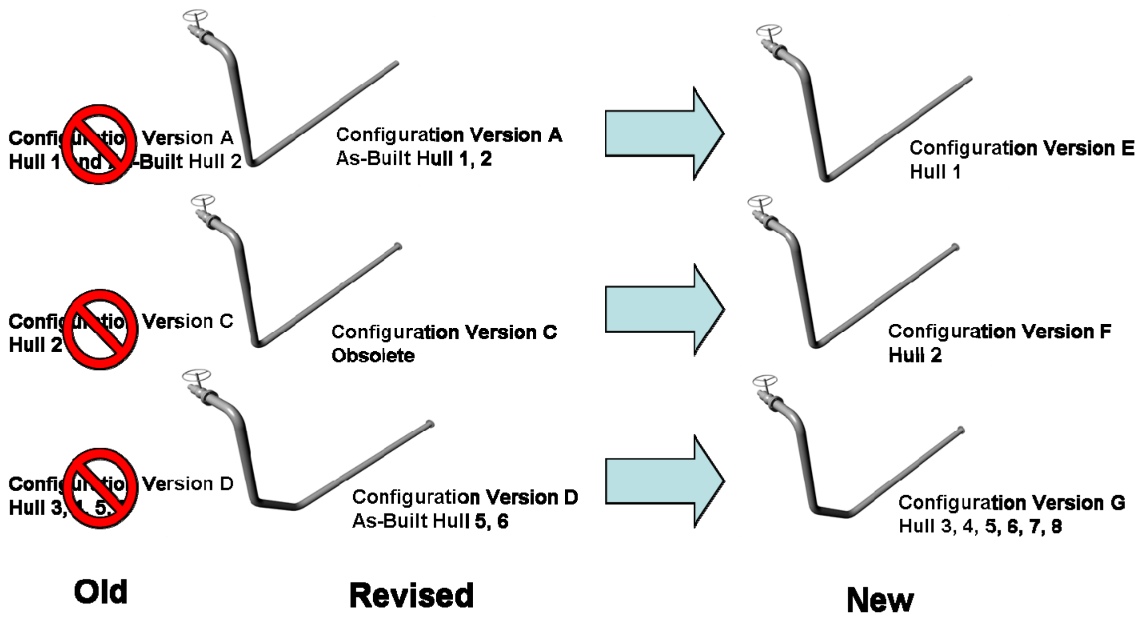


Figure A - 8 Configuration Management Scenario Change Number 3

In the fifth year, a mission change for hull number 5 requires the addition of a strainer and a pipe to be re-routed around a newly installed piece of equipment (Figure A - 9).

The re-routed pipe and strainer needs to be added to the version G configuration to accommodate the change in hull number 5.

As a result, there is one new version (H). Version G no longer applies to hull 5, but is still valid as the as-maintained for hulls 3, 4, 6, 7 and 8.

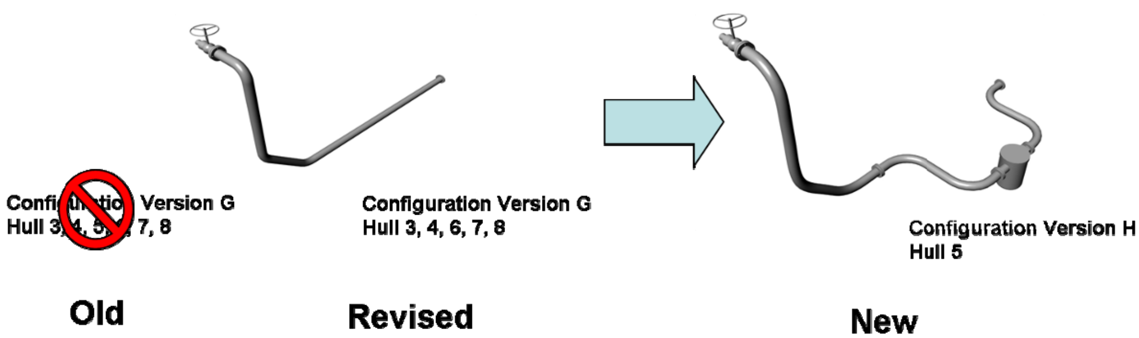


Figure A - 9 Configuration Management Scenario Change Number 4

In year 8, the last year of this scenario, the fuel fill valve in hulls 1, 2 and 3 show signs of premature wear. Since the hose gate valve is no longer available it is replaced with a hose bib, flange, gasket and gate valve on hulls 1, 2 and 3.

The new gate valve, hose bib, flange and gasket need to be applied to the version E configuration to accommodate the change in hull 1. The new gate valve, hose bib, flange and gasket need to be applied to the version F configuration to accommodate the change in hull 2. Likewise, they need to be applied to the version G configuration to accommodate the change in hull 3 (Figure A - 10).

As a result, there are three new versions (I, J, K). Version E no longer applies to hull 1 and is obsolete. Version F no longer applies to hull 2 and is obsolete. Version G no longer applies to hull 3, but is still valid as the as maintained for hulls 4, 6, 7 and 8.

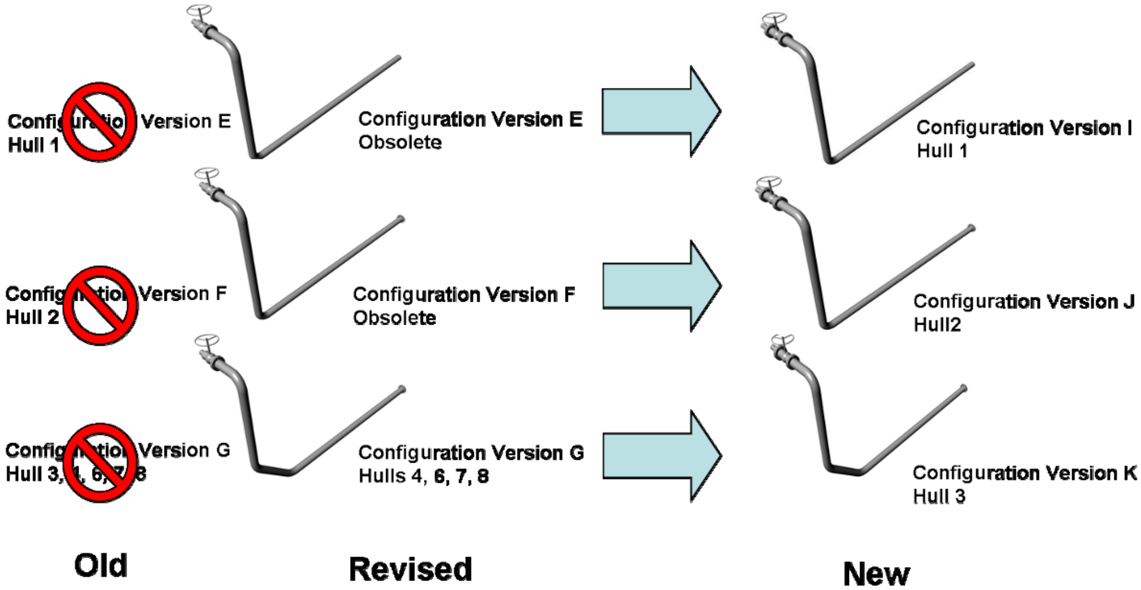


Figure A - 10 Configuration Management Scenario Change Number 5

Figure A - 11 illustrates the various hull configurations at each point in time. The number of parts required using the document-based approach vs. the part-based approach over time is also shown. The details in this figure are too small to be read easily; however, a higher resolution wall chart is available. Each column represents a single year in the life cycle. The bottom row contains a description of the changes that are being made at that time. The next row up lists the part-based configuration units along with corresponding part counts. Above the part-based configuration units are graphical displays of each hull configuration at each point in time. At the top are the document-based configuration units and corresponding part counts. You should notice that the part counts for the document-based approach begin to grow much more rapidly than the part-based. This is significant because when a change is made to a part using the document-based approach; it needs to be made in several locations, whereas when a change is made using the part-based approach it only needs to be changed in one location and the hull effectivity defined accordingly. The Green background indicates an addition to that configuration unit at that time period. The Red background indicates a deletion to a configuration unit at that time period. The Gray background indicates the configuration unit is no longer the current version. The "X" indicates the configuration unit is obsolete.

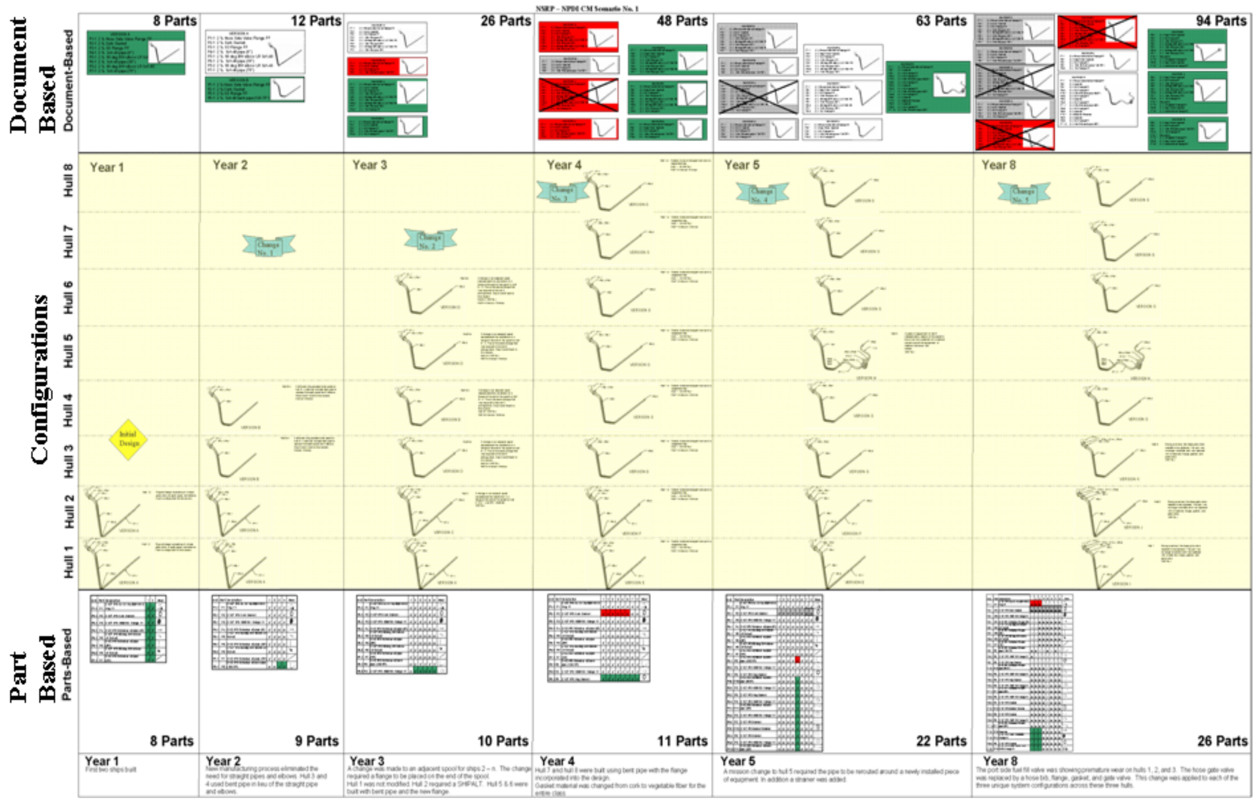


Figure A - 11 Comparison of Document versus Part-based Configuration Management

The next several paragraphs describe each change in the context of the document and part-based approaches.

For version A of the design there is virtually no difference between the document-based and the part-based configuration management process. At the end of the first year, both processes require 8 occurrences of 8 parts to define the system and manage the configuration of the class (Figure A - 12).

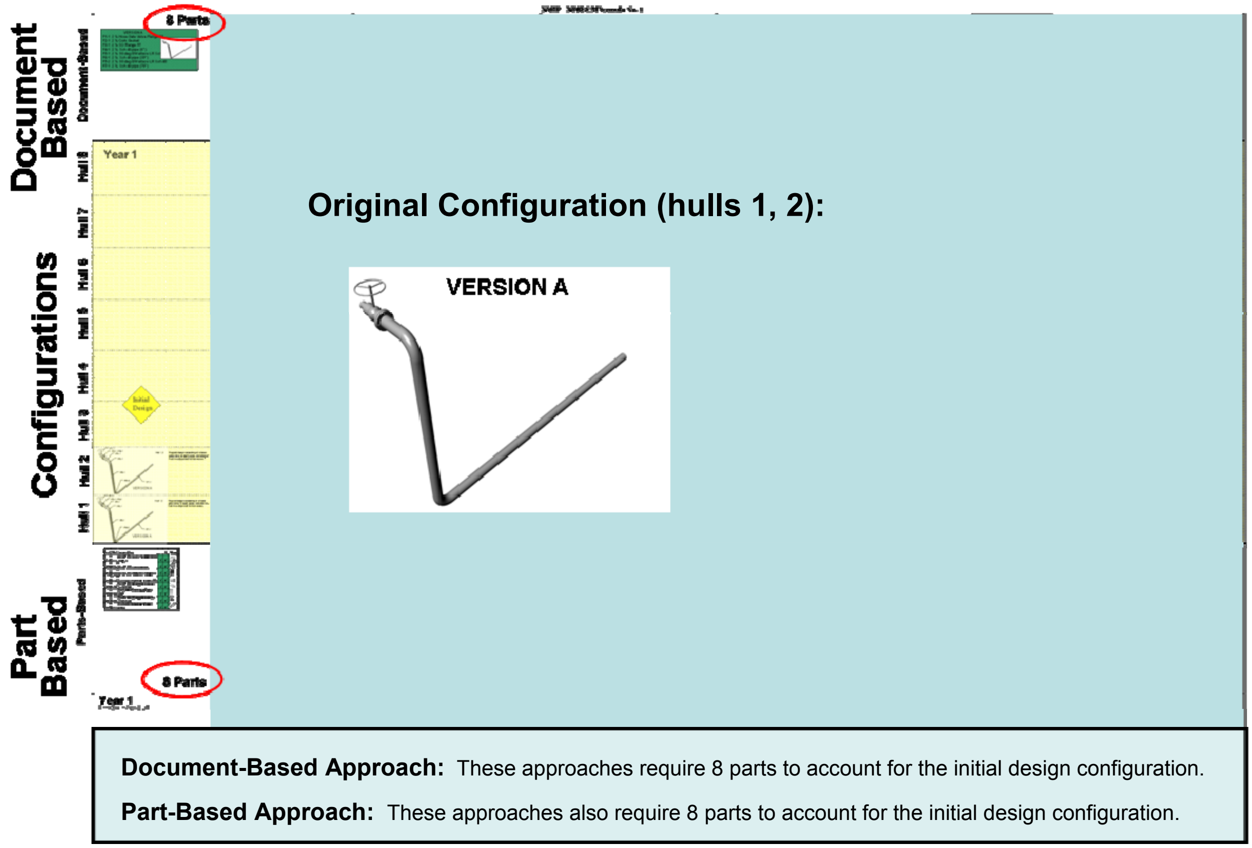


Figure A - 12 Original Configuration: Document versus Part-based Approach

At the end of the second year, two different versions are required to define 4 hulls (Figure A - 13). Hull 1 and hull 2 conform to version A. Hull 3 and hull 4 conform to version B. Configuration management becomes a little more complicated because 2 different configurations are being managed. The document-based configuration management process requires two models having a total of 12 occurrences of 12 parts. The model copy results in 3 duplicated parts. The part-based configuration management process requires nine occurrences of 9 parts.

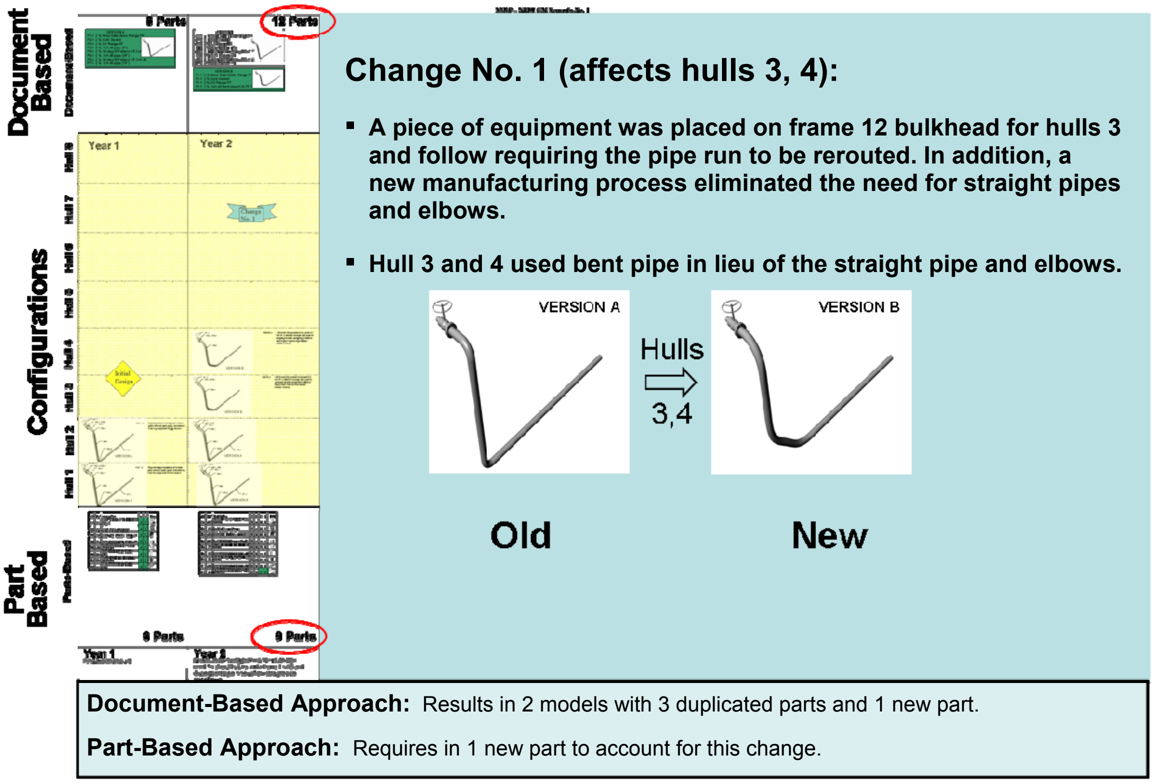


Figure A - 13 Change Number 1: Document versus Part-based Approach

At the end of the third year, three versions are required to define 6 active hulls (Figure A - 14). Hull number 1 conforms to version A. Hull number 2 conforms to version C. Hulls 3, 4, 5 and 6 conform to version D. The document-based approach requires four separate models. One of the models no longer reflects an active ship, but is required for design history of the as-built configuration. The document-based approach requires the same change to be performed on two models, resulting in 13 additional duplicate parts. The part-based approach just required one additional part with appropriate hull effectivity.

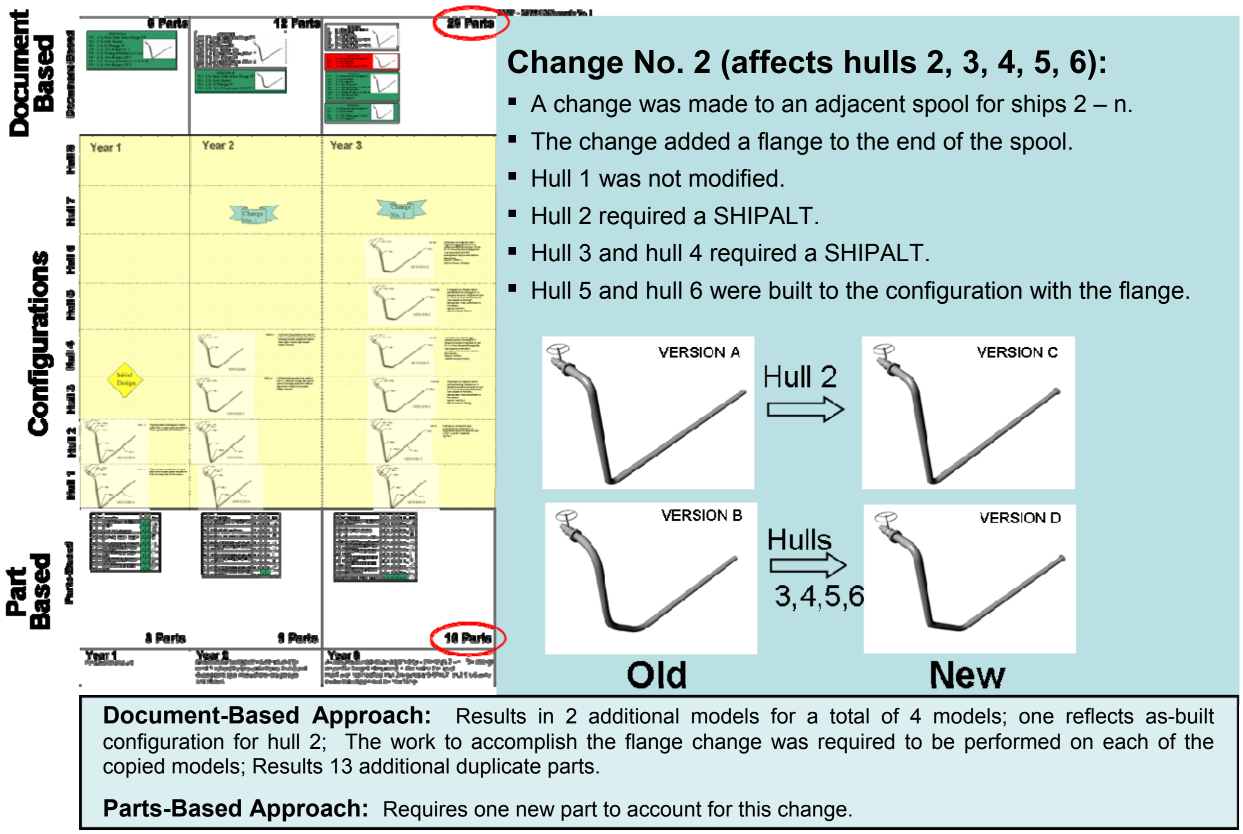
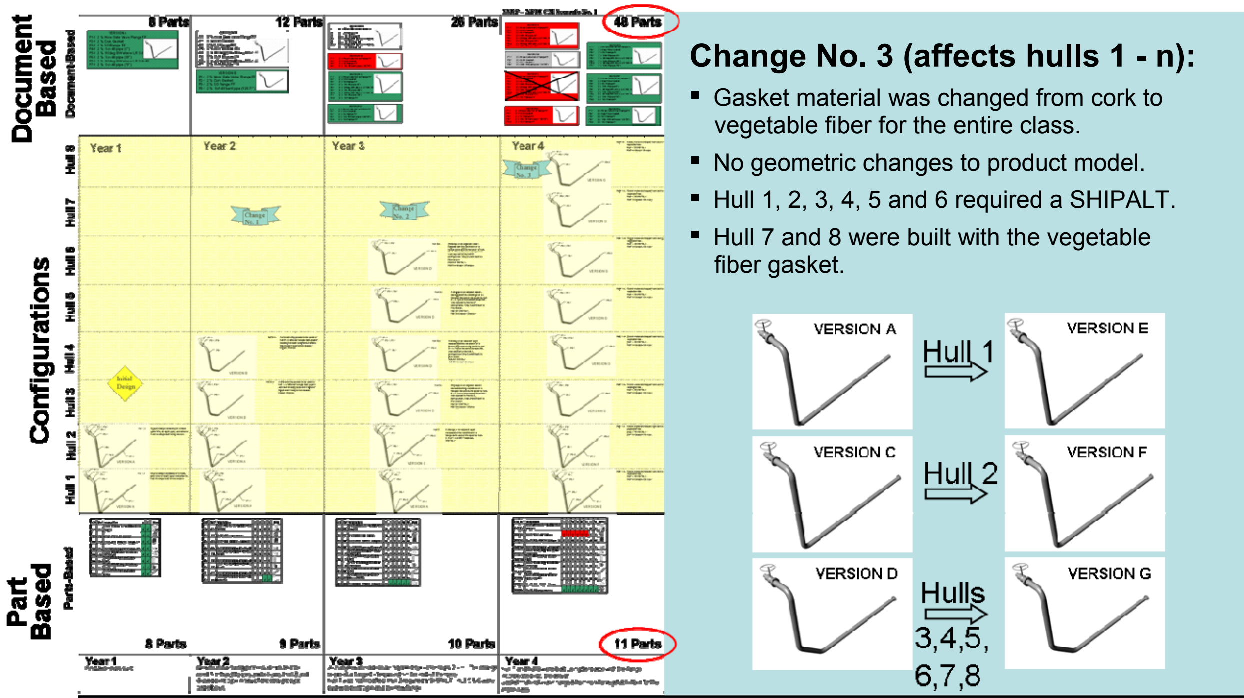


Figure A - 14 Change Number 2: Document versus Part-based Approach

During the fourth year, a class change is made (Figure A - 15). At the end of the fourth year, three versions are required to define 8 hulls. Hull 1 conforms to version E. Hull 2 conforms to version F. Hulls 3, 4, 5, 6, 7 and 8 conform to version G. The document-based approach requires seven separate models, four that no longer reflect an active ship but are required for design history. The document-based approach requires the same change be performed on three models resulting in 21 additional duplicate parts vs. the one additional part with appropriate hull effectivity in the part-based approach.



Document-Based Approach: Results in 3 additional models for a total of 7 models; 4 models reflect historical as-built or as-maintained configurations; The work to accomplish the flange change was required to be performed on all 3 models representing valid configurations; Results in 21 parts duplicated across multiple models.

Parts-Based Approach: Requires one new part to account for this change.

Figure A - 15 Change Number 3: Document versus Part-based Approach

At the end of the fifth year, four versions are required to define 8 hulls (Figure A - 16). Hull 1 conforms to version E. Hull 2 conforms to version F. Hulls 3, 4, 6, 7 and 8 conform to version G. Hull 5 conforms to version H. The document-based approach requires an additional model with 4 more duplicated parts, in addition to the 11 new parts required by both the document and part-based approaches.

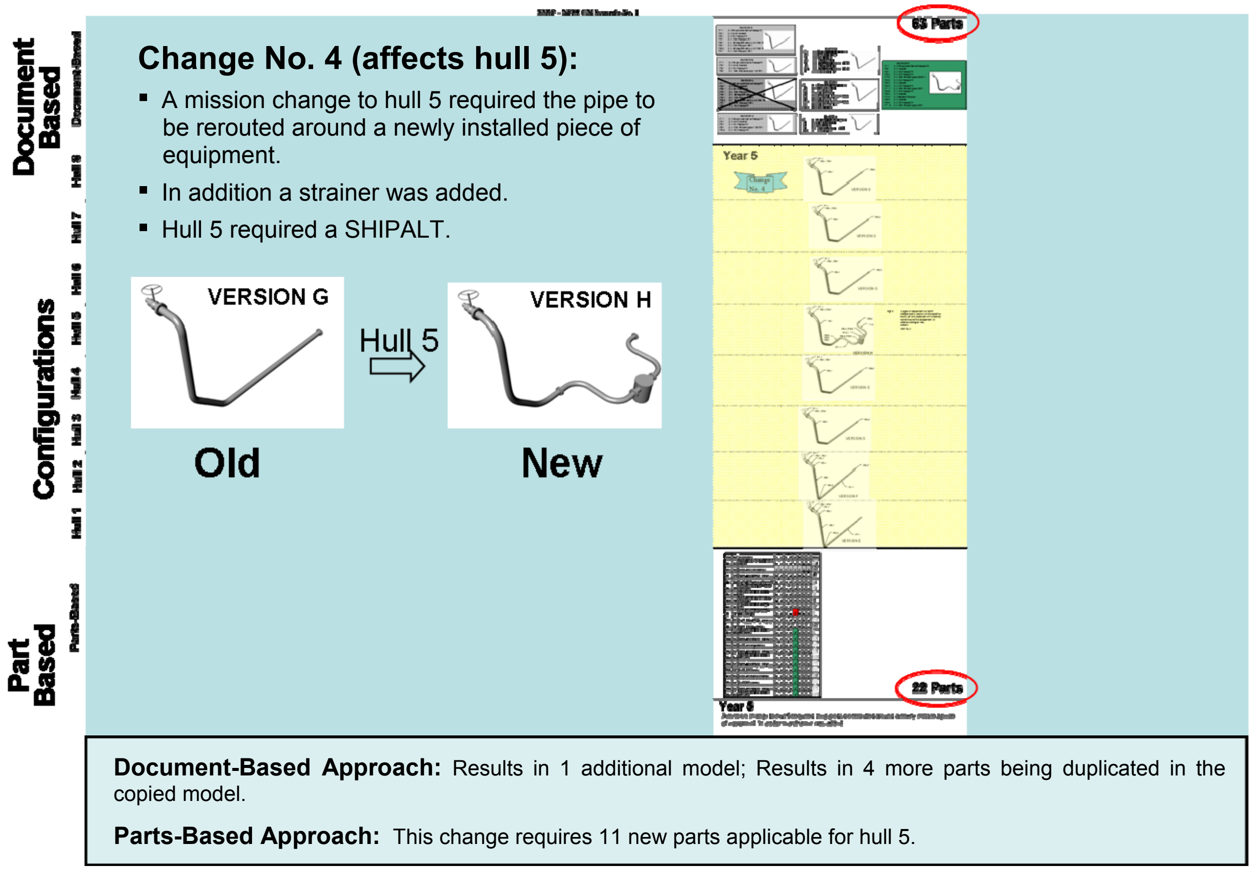


Figure A - 16 Change Number 4: Document versus Part-based Approach

At the end of the eighth year, five versions are required to define 8 hulls (Figure A - 17). Hull 1 conforms to version I. Hull 2 conforms to version J. Hull 3 conforms to version K. Hulls 4, 6, 7 and 8 conform to version G. Hull 5 conforms to version H. The document-based approach requires the same change be performed on three models resulting in 27 additional duplicate parts and three additional models. Using the part-based approach only one change is necessary with hull effectivity applied at the part level.

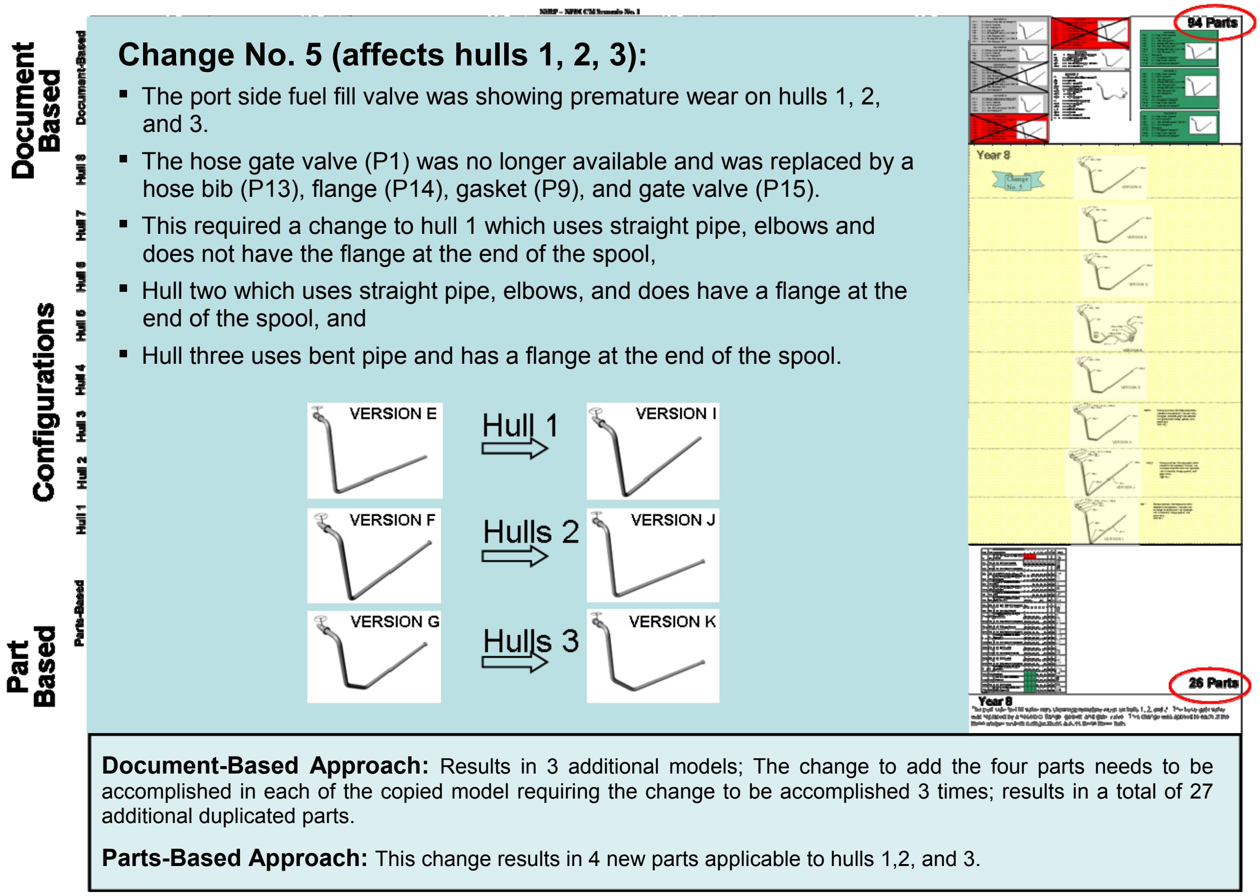


Figure A - 17 Change Number 5: Document versus Part-based Approach

The previous sections described the five changes in the scenario and compared and contrasted the different part counts and configuration unit counts that need to be maintained between the document-based and the part-based approaches. In the following paragraphs, change number 5 will be used as an example to illustrate the level of effort required to manage the configuration of the product model.

In change 5, the hose gate valve is replaced with a gate valve, gasket, threaded flange and hose bib (Figure A - 18). In the document-based approach, the 4 new parts need to be defined separately for each of the three configurations. However, in the part-based approach, the 4 new parts are defined once in the product model and the hull effectivity is defined accordingly in the database.

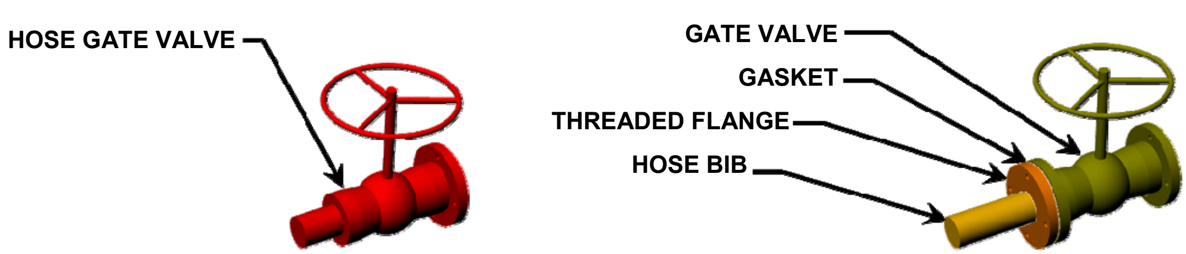


Figure A - 18 Change Number 5 Illustration

Figure A - 19 illustrates a sample Engineering Work Order which defines the change and the hulls to which the changes apply.

ENGINEERING WORK ORDER DEFINITION

DATE: 3/8/07 **CLASS:** TWR 841 **CHANGE APPLICABILITY:** Hull 1, 2, 3 (ONLY)

SUBJECT: Replace port side hose gate valve with hose bib valve Pipe Spool #10

CHANGE DESCRIPTION: The port side hose gate valve, part number P1-2 is showing premature signs of wear. Engineering evaluation has determined that this valve needs to be replaced for the designated hulls. This change is to modify the Engineering BOM and model configuration only. It does include work to modify the System Diagram to reflect this change also. A separate EWO will be issued to accomplish the associated Tech Manual and manufacturing Work Packages for Rip out and Replacement Ship Alteration Documentation and instructions to implement this change on the ship.

SYSTEM: Fuel Oil Fill and Transfer **DOCUMENT:** Not Applicable

WORK ORDER: TWR– 078 **MAN-HOUR ESTIMATE:** 1.75Hrs.

LEAD GROUP: Piping Distributed Systems Group

ASSOCIATED DRAWINGS: Diag078;Dwg078;Tech Manual078;ShipAlt-078

TECH IMPACT: WTB Y **NOISE** N **SPARES** Y **WTS** N **SAFETY** Y

APPROVALS:

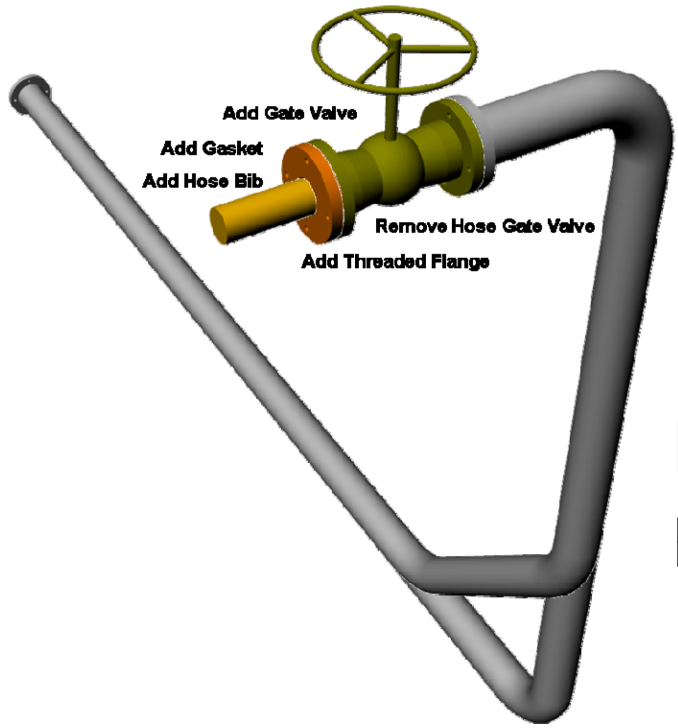
JoeDesigner	StewartSteakholder	MichaelMicroscope
ORIGINATOR	SHIP MGMT	SUPSHIP
Program Peter	CharlieConfigurator	InspectorClewseau
PROGRAM OFFICE	ECM	TEST

Figure A - 19 Change Number 5 Example Engineering Work Order

In the document-based approach, this hull effectivity is applied at the document level. To make the change for hull number 1, we need to bring up the configuration of hull number 1 which is version E. Once completed, this will be saved as version I. The following steps are required (Figure A - 20):

- The gate valve is removed from the configuration.
- The new gate valve is instanced.
- The gasket is placed.
- The threaded flange is added.
- Finally, the hose bib is instanced.

The document is saved and the document hull effectivity table is updated to reflect the new document version (I) applying to hull 1. At this point, the change has only been made for hull 1 in the document-based approach.



Open the file containing the model of the pipe run for Hull 1. This is version E.

AUTHOR CHANGES
Remove the Hose Gate Valve and replace it with a gate valve, gasket, flange, and hose bib.
SAVE CHANGES
A new document (file) will be saved as version I.
UPDATE EFFECTIVITY
Update the document effectivity table to reflect the new file being applicable to Hull 1.

	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	E	F	G	G	H	G	G	G

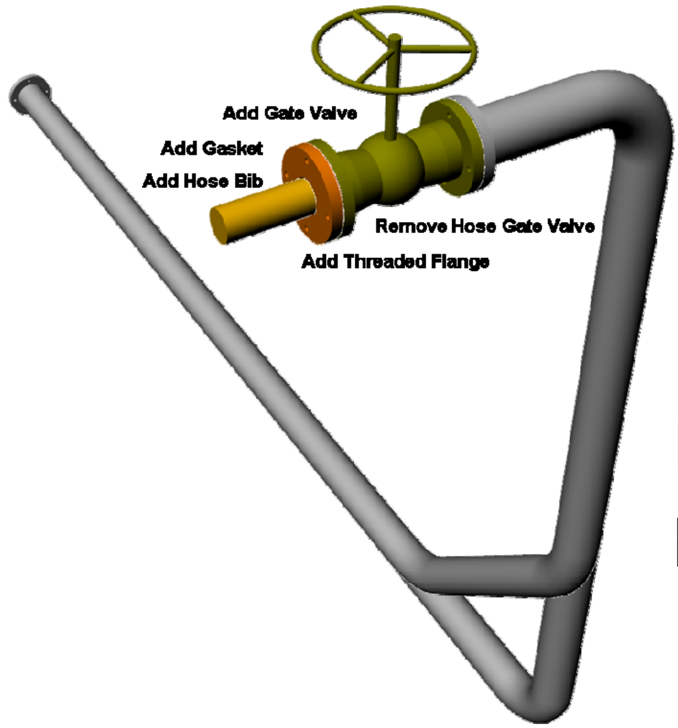
	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	I	J	K	G	H	G	G	G

Figure A - 20 Change Number 5, Hull Number 1: Document-based Approach Changes

To make the change for hull number 2, we need to bring up the configuration of hull number 2 which is version F. Once completed, this will be saved as version J. The following steps are required (Figure A - 21):

- The gate valve is removed from the configuration.
- The new gate valve is instanced.
- The gasket is placed.
- The threaded flange is added.
- Finally, the hose bib is instanced.

The document is saved and the document hull effectivity table is updated to reflect the new document version (J) applying to hull 2. At this point, the change has been made for hulls 1 and 2 in the document-based approach.



Open the file containing the model of the pipe run for Hull 2. This is version F.

AUTHOR CHANGES
Remove the Hose Gate Valve and replace it with a gate valve, gasket, flange, and hose bib.
SAVE CHANGES
A new document (file) will be saved as version J.
UPDATE EFFECTIVITY
Update the document effectivity table to reflect the new file being applicable to Hull 2.

	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	I	F	G	G	H	G	G	G

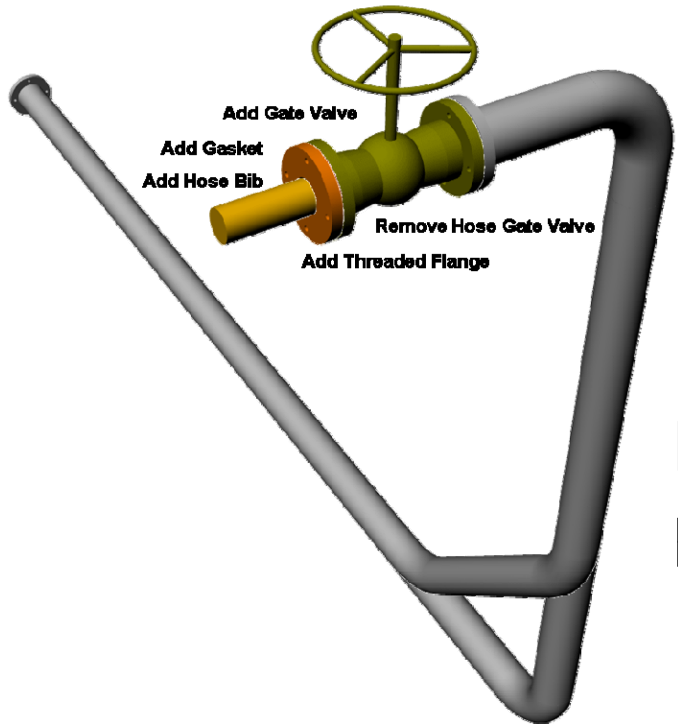
	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	I	J	G	G	H	G	G	G

Figure A - 21 Change Number 5, Hull Number 2: Document-based Approach Changes

To make the change for hull number 3, we need to bring up the configuration of hull number 3 which is version G. Once completed, this will be saved as version K. The following steps are required (Figure A - 22):

- The hose gate valve is removed from the configuration.
- The new gate valve is instanced.
- The gasket is placed.
- The threaded flange is added.
- Finally, the hose bib is instanced.

The document is saved and the document hull effectivity table is updated to reflect the new document version (K) applying to hull 3. At this point, the change has been made for hulls 1, 2 and 3 using the document-based approach.



Open the file containing the model of the pipe run for Hull 3. This is version G.

AUTHOR CHANGES
Remove the Hose Gate Valve and replace it with a gate valve, gasket, flange, and hose bib.

SAVE CHANGES
A new document (file) will be saved as version K.

UPDATE EFFECTIVITY
Update the document effectivity table to reflect the new file being applicable to Hull 3.

	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	I	J	G	G	H	G	G	G

	Document Version by Hull							
Document Number	1	2	3	4	5	6	7	8
TWR841- FO X-FER DWG	I	J	K	G	H	G	G	G

Figure A - 22 Change Number 5, Hull Number 3: Document-based Approach Changes

Using the part-based approach, the same change request (Figure A - 19) can be used to reduce the complexity and repetitive user actions. The information in the change request is used to define the work content or query. The query is run against the part-based database which returns a filtered parts list meeting the query criteria (Figure A - 23).

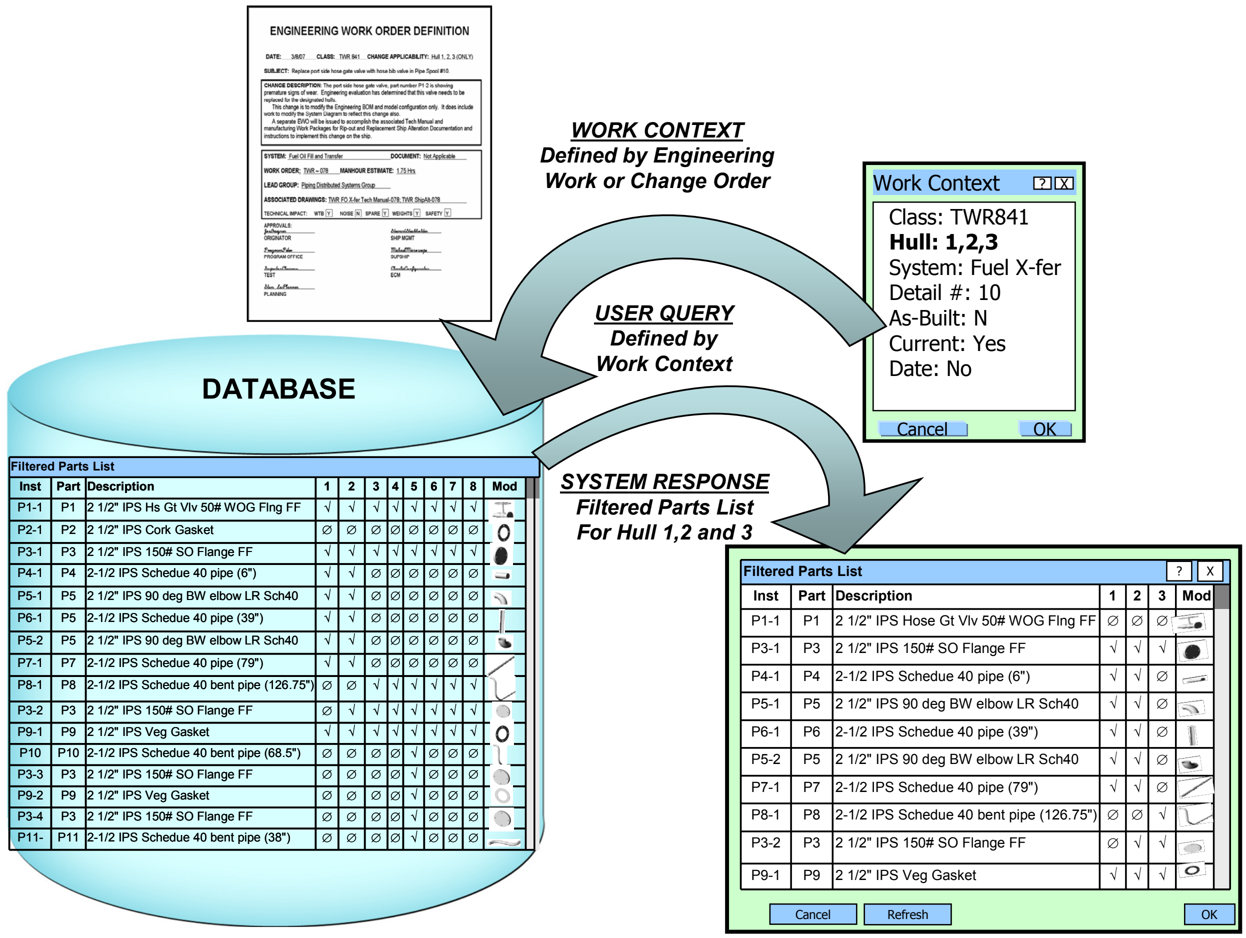
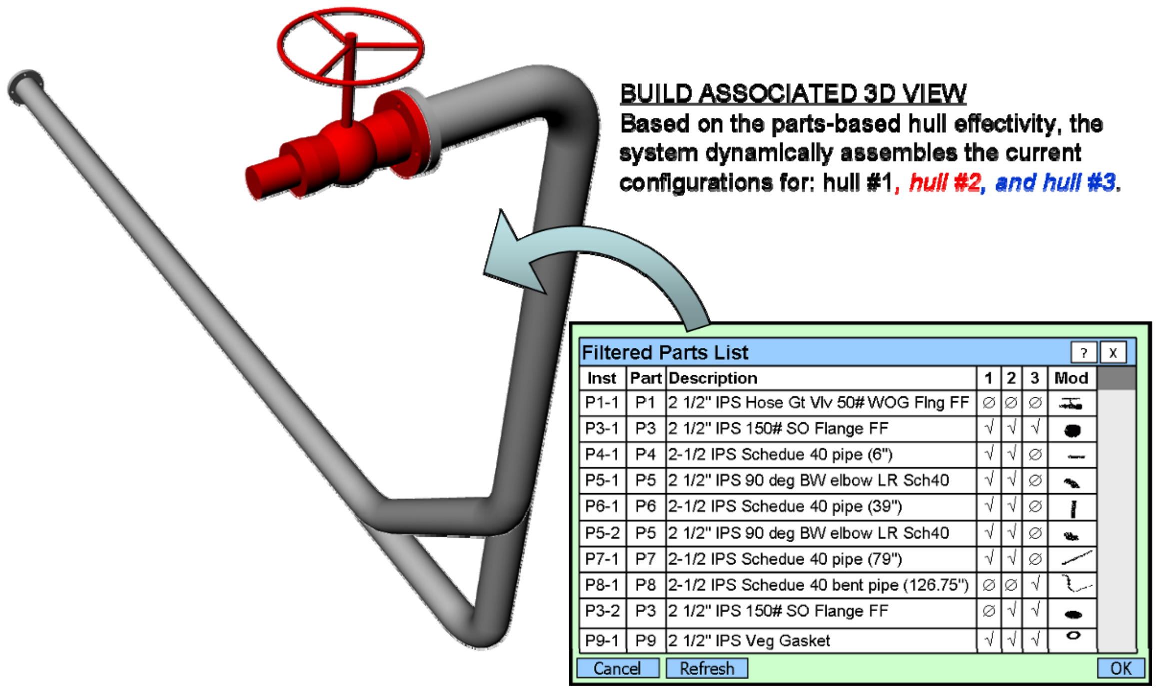


Figure A - 23 Change Number 5, Hull Numbers 1, 2, and 3: Part-based Approach Query

The parts-based approach uses this filtered parts list to dynamically display the configurations required to accomplish this change. The results of this query are the overlay of hulls 1, 2 and 3 (Figure A - 24).



Inst	Part	Description	1	2	3	Mod
P1-1	P1	2 1/2" IPS Hose Gt Vlv 50# WOG Flng FF	∅	∅	∅	
P3-1	P3	2 1/2" IPS 150# SO Flange FF	√	√	√	
P4-1	P4	2-1/2 IPS Schedule 40 pipe (6")	√	√	∅	
P5-1	P5	2 1/2" IPS 90 deg BW elbow LR Sch40	√	√	∅	
P6-1	P6	2-1/2 IPS Schedule 40 pipe (39")	√	√	∅	
P5-2	P5	2 1/2" IPS 90 deg BW elbow LR Sch40	√	√	∅	
P7-1	P7	2-1/2 IPS Schedule 40 pipe (79")	√	√	∅	
P8-1	P8	2-1/2 IPS Schedule 40 bent pipe (126.75")	∅	∅	√	
P3-2	P3	2 1/2" IPS 150# SO Flange FF	∅	√	√	
P9-1	P9	2 1/2" IPS Veg Gasket	√	√	√	

Figure A - 24 Change Number 5, Hull Numbers 1, 2, and 3: Part-based Approach 3D View

Now that the user has defined the work content from the Engineering Change Order and the system has dynamically displayed the associated configurations, the CAD operator uses normal CAD modeling techniques to incorporate the changes. The following steps are required (Figure A - 25):

- The hose gate valve is removed.
- The new gate valve is instanced.
- The gasket is placed.
- The threaded flange is added.
- Next, the hose bib is instanced.

The operator then saves the changes. The parts-based system uses the work content that was defined at the beginning of the user's session to update the filtered parts list.

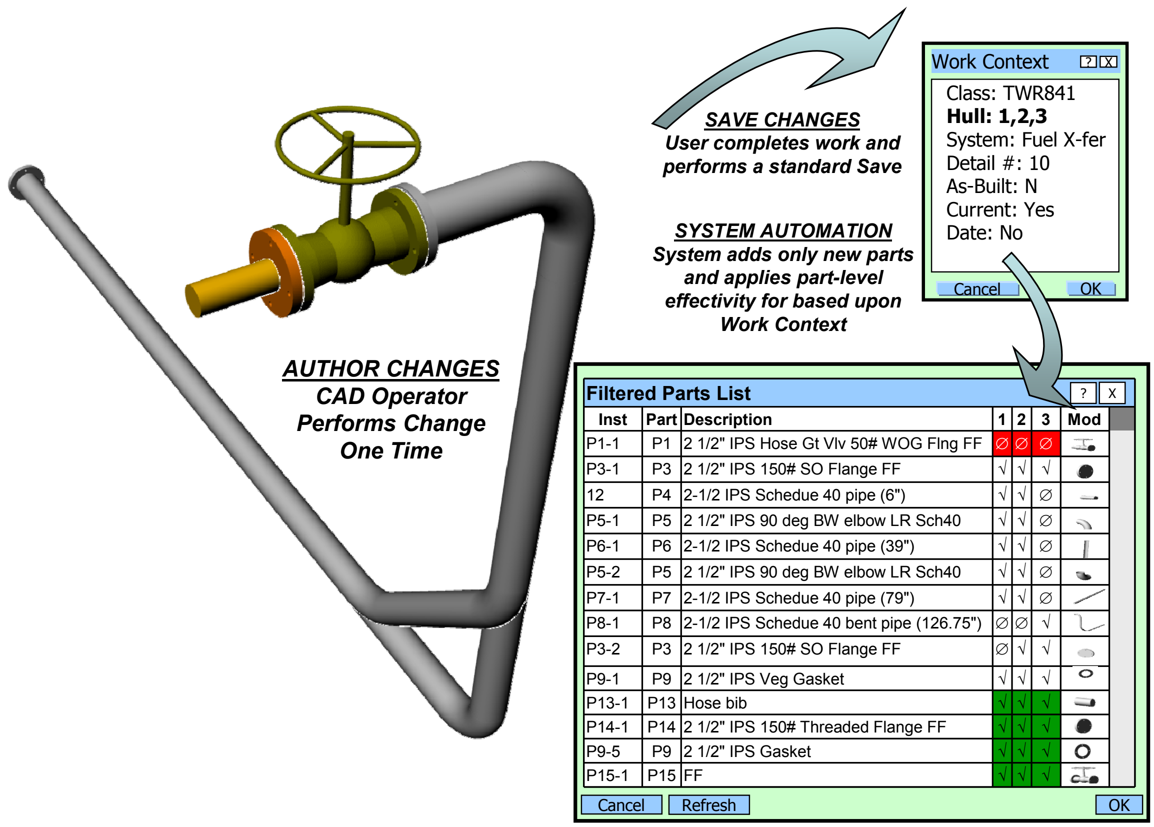


Figure A - 25 Change Number 5, Hull Numbers 1, 2, and 3: Part-based Approach Changes

Using the part-based approach the changes are stored in the database (Figure A - 26). This change was accomplished in the parts-based approach for all three hulls simultaneously by instancing each of the components once. In contrast, the document-based approach required the operator to perform the same instance operations repetitively for each configuration. This additional labor is compounded when you factor in the creation of downstream products such as drawings and manufacturing data and processes such as design review and approval for each configuration.

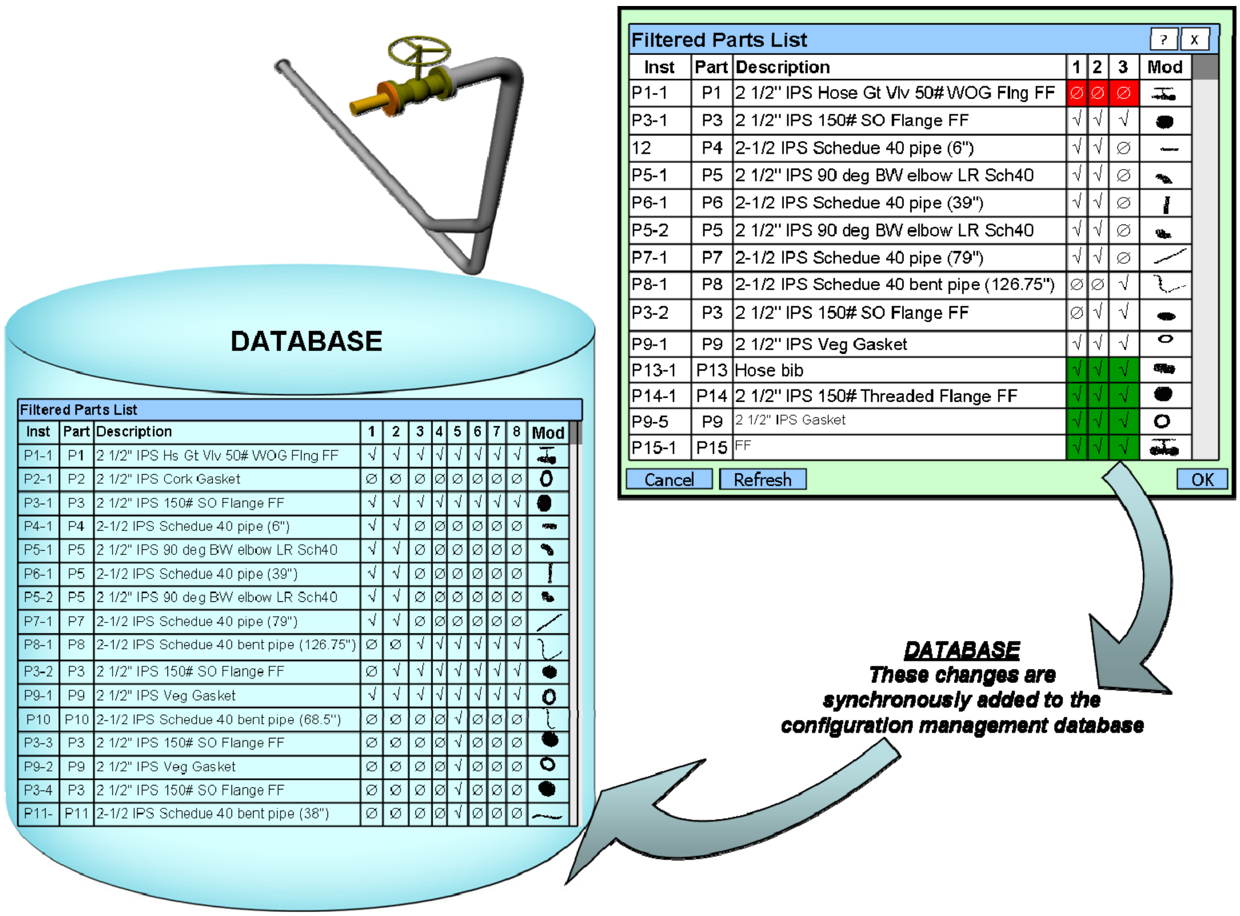


Figure A - 26 Change Number 5, Hull Numbers 1, 2, and 3: Part-based Approach Results

This scenario was limited to a single pipe run in a class of 8 hulls over an eight year time period. It was limited to some Bill of Material and geometric data changes. It does not include the impact of these changes on all of the other downstream products. The changes discussed in this scenario do not represent the volume of change that occurs for a ship class. To help put this in perspective, a recent class of 5 large auxiliary multipurpose supply ships which are not as sophisticated as a combatant or sub averaged about 150 class changes a year. Even with all of these simplifications, this scenario still demonstrates the inefficiency and complexity of the document approach vs. the part-based approach.

This basic change scenario using the document-based process illustrates the negative impacts associated with the ever-increasing data duplication across multiple documents. The part count rapidly increased to 4 times the actual number of parts with only 8 ships. Even more important is the fact that the labor required to affect change increases rapidly over time as a result. When different document-based models are created for each product configuration, every class change must be made in each different document. In addition to the designer having to make the change in each document model, the checking, reviewing and approval of each change also needs to be performed for each and every document model. The effort to manage the correct configuration of data using the document-based approach quickly cascades when considering: System interfaces, Drawing sheets and tech pubs, Bill of Materials, Attribute changes, Pipe spool sketches, Work packages, flow analysis, structural analysis, schematics, manufacturing plans, and NC data. Remember, this scenario only addressed a portion of a single pipe system in a space. A real ship class will have hundreds of such systems in each area or zone and over a thousand systems for an entire ship. The high cost of changes using the traditional document-based systems has resulted in the product models not being maintained for the ship's lifecycles. These issues also inhibit technology insertion in the ship class.

The benefits of the parts-based Configuration Management approach include the elimination of data duplication by managing change at the part level. This in turn reduces redundant, non-value added effort required to affect change. Parts-based management allows for dynamic assembly of part views including: hull specific applicability, integrated design, system design review, and work package graphics.

There are significant issues to be resolved with the parts-based Configuration Management approach. One is the lack of vendors and products supporting parts-based configuration management. Another is the requirement for authoring tools to have an open architecture to enable cross-vendor parts-based configuration management of product model data.

This scenario provided an explanation of a document centric approach and a part centric approach to product model data configuration management. The advantages of the part-based approach were illustrated using a simplified example, with the conclusion that the part-based approach is the way we need to go for efficient configuration management.