# PDM-CAD/CAM INTERFACING IN SHIPBUILDING

Development of a generic PDM solution, tested on NUPAS-CADMATIC hull and outfitting

Steyn Schiltkamp
Marine Technology
Ship Production
1376578
steynschiltkamp@gmail.com
+31 6 40807501

16 SEPTEMBER 2015

TU DELFT
Numeriek Centrum Groningen

## Acknowledgements

From November 2014 to September 2015 I have worked on my graduation project concerning the use of product data management together with CAD/CAM software in shipbuilding. From this project I have learned a lot about product data management and CAD/CAM software which was fun but I have also learned that writing a thesis is probably one of the least favourite things I like to do. The people I would like to thank that helped me acquire this new knowledge are my two supervisors Jenny Coenen (TU Delft) and Simona Stoica (Numeriek Centrum Groningen (NCG)). This project was done in collaboration with Numeriek Centrum Groningen and therefore I would like to thank all the people at NCG and CADMATIC who have made this possible.

Roughly half of the time spent on this project was spent in Groningen. I would like to thank my grandparents for letting me stay with them in Ter Apel and that I could use their car. Also I would like to thank my parents and the rest of my family for all their support. Last but not least many thanks to my girlfriend Elles, housemate Job, and the rest of my friends and acquaintances that have somehow helped and/or supported me during this project.

Thesis for the degree of MSc in Marine Technology in the specialization of *Ship Production*

# PDM-CAD/CAM interfacing in shipbuilding

By

Steyn Schiltkamp

Performed at

Numeriek Centrum Groningen

This thesis (SDPO.15.024.m) is classified as confidential in accordance with the general conditions for projects performed by the TUDelft.

16-09-2015

Company supervisors
Responsible supervisor: Dr. S. Stoica
E-mail: stoica@ncg.nl

Thesis exam committee
Chair/Responsible Professor: Prof Ir. J.J. Hopman
Staff Member: Dr. Ir. J.M.G. Coenen
Staff Member: Dr. Ir. H.P.M. Veeke
Company Member: Dr. S. Stoica

Author Details
Studynumber: 1376578
Author contact e-mail: steynschiltkamp@gmail.com

## Abstract

The aim of this thesis is to make clear what the possibilities are towards the implementation of Product Data Management (PDM) in combination with CAD/CAM software in the shipbuilding industry. Even though in many industries there is already an extensive PDM integration in combination with CAD/CAM software, the shipbuilding industry is in its initial phase of integrating PDM in its standard way of working. The main focus of this research is to Investigate the use of PDM together with CAD/CAM software and design a concept solution that meets the requirements of an optimal integration collaboration between PDM and CAD/CAM software.

In order to make clear what these possibilities are, a literature study has been done. From literature the theory of PDM has been described and the UML modelling language has been chosen for a description of the PDM possibilities in shipbuilding. Before description of these possibilities a generic engineering process of shipbuilding is described to function as a basis to identify PDM possibilities. A virtual block of a not existing ship has been used to delimit this engineering process to a surveyable size. Within the described engineering process, activities are identified that not only have a connection to PDM but may also benefit from PDM. Using the described engineering process and PDM theory, the PDM functions have been described. These functions have been directed to assist the activities that are linked to PDM and also benefit from PDM. The PDM functions are not only described for their functionality but they are also placed in a system architecture to give a representation of how the PDM functions are executed when used by the described engineering process. Based on the description of the engineering process and the PDM functionality, a qualitative measurement has been performed to give a clear indication of the benefits that the PDM solution has on the described engineering process. In order to show that the described engineering process can also handle the engineering of a complete ship, the process has been expanded to describe the engineering of two existing ships. Finally the functionality of the PDM solutions has been tested for its feasibility on the NUPAS-CADMATIC hull and outfitting packages.

The implementation of PDM consists of two main aspects. The main perspective is taken into consideration from a system point of view where PDM functionality is executed by a PDM system and makes re-use of engineering data possible. Moreover, the data storage needs to be redesigned to a central data vault that facilitates this re-use and other benefits. The other perspective is from a managerial point of view, where configuration management and a different decision making is implemented in the way of working on the work floor of the engineers. An important part of this research are the changes to the system regarding the NUPAS-CADMATIC hull and outfitting packages. At this moment, the proposed PDM solution is not feasible with the packages. This means that should the packages decide to adhere to the proposed solution, changes need to be made to the systems for them to have a shared central data vault and therefore benefit from proper re-use of engineering data as described by PDM.

# Table of contents

# 1 Introduction

Since the transfer from information storage on paper to digital information storage, there have been different methods and systems to organize and save data. Shipyards use of these systems as well. In a production process like shipbuilding, the need to streamline and control the process is high. During the production process, lots of documents are needed, which can be generated at any stage. Existing documents can be modified according to different requirements of each stage, or to changes in the design of a ship.

A trend in modern shipyards is to use PDM (Product Data Management) systems. The use of a PDM system should improve the shipbuilding process, from initial design, preliminary design, basic design, detailed design, ship production and support and maintenance. Advantages of the PDM theorem and systems are known and well described.

The use of 3(2)D CAD/CAM systems that define the geometry of a ship have become standard in the complete production cycle of a ship. All shipyards have their own way of working and therefore require a different solution on the collaboration between CAD/CAM and PDM. The fact that the shipyards do not use the same software packages magnifies the problem.

The implementation of PDM in a complex production system is difficult and the optimal or practical solution is not yet defined. These systems should have a close interaction with a PDM system. Does a PDM strategy or PDM-system match the different storage and data management systems used in the production process?

## 1.1 Product Data Management

The information generated and used in the complete manufacturing process of products, needs to be stored and shared among people working on the engineering and/or production of certain products. The use of Product Data Management (PDM) can help organize, track and control data used in the complete manufacturing process. PDM does not stand on its own, it is often compared and linked to Product Lifecycle Management (PLM) and Enterprise Resource Planning (ERP).

Product Data Management is a strategy that gives access to the right product and process related information to the correct person at the right moment in the product lifecycle. Most of this information is about pieces of a ship, these pieces are referred to as parts or components, but when a piece of a ship is built up of parts and/or components it is referred to as an assembly. PDM makes it possible to create processes and secure the data influenced and used by these processes. PDM generates a secure and managed environment where all relevant information is saved without redundancy, but with conservation of history. (J.P. la Fontaine, M.G.R. Hoogeboom, & J.S. Könst, 2008).

PDM is not necessarily a software package. The different software packages make it possible for a company to carry out their PDM strategy. PDM systems in the form of software packages are used to control information, files, documents, and work processes required to design, build, support, distribute, and maintain products. Typical product-related information in a PDM package includes: geometry, engineering drawings, project plans, part files, assembly diagrams, product specifications, numerical control machine-tool programs, analysis results, correspondence, bills of material, engineering change orders, and many more (D. Tony Liu & X. William Xu, 2001). The storing and sharing of all these different kinds of information makes PDM very suitable for the different aspects of the engineering process. PDM is made up of five basic functions:

1. Data vaulting

   All the product information needs to be stored and retrieved during the engineering process. Within the vault some kind of document management system will arrange all the data according to a predefined structure.
2. Workflow and process management

   The engineering process that uses PDM needs workflow and process management to make sure the right procedures are used when working with all the stored product data. The most important procedure is the status handling of the product, changes to documents need to be approved or declined.
3. Product structure management

   Makes sure that not only information of separate parts is stored. Separate parts may form a designed product or assembly, this product configuration with its different designs and variations need to be well managed. Products and assemblies may contain bills of materials, which in turn need to be handle in a way that the information they contain is well distributed.
4. Parts management

   To be able to re-use standard components in the design of a product, the standard components or parts need to be managed separately.
5. Program management

   To keep track of the engineering process, program management provides work breakdown structures and project- tracking, scheduling, and planning.

Evidently PDM has a lot of advantages, but what does a company implementing PDM want to achieve? At the end of the day, a company wants to reduce the costs and/or time needed to deliver a product to market when manufacturing a certain product. This cost reduction or decreased time to market can be measured by improved productivity, reduced cycle times, reduction of errors, improved product quality etc. These kinds of improvements are what providers of PDM packages advertise:

- Improved business productivity
  - Optimization of operational resources
  - 3D-CAD integration
  - Use of Bill of Material management
  - Facilitate collaboration between global teams
  - Reduced cycle time
- Improved data accessibility
  - Provide visibility for better business decision taking
  - Broaden access to product information
  - Easy data vault use
  - 3D-view for non-CADusers
  - Browser ready for Internet/Ethernet use
  - Excellent part search engine
- Improved file management
  - No data conflicts during concurrent engineering
  - Export data downstream to be used by non-CADusers
  - User right management
- Improved workflow management

- Compliance with business regulatory requirements
- Re-use of design data
- Part attribute records
- Engineering change control

Looking at all these advertised improvements or functions that should lead to improvements, it appears that Product Data Management is similar to Product Lifecycle Management (PLM). However, PDM is not the same as PLM but it is part of PLM. Most PDM providers also provide PLM and have combined the packages as one PLM package. As the name implies, PLM focuses on the complete lifecycle of a product with all its containing phases. The product is managed from the first stages of design to the deconstruction of the product. PDM mainly focuses on the design, engineering and production of a product, these phases are just a view in lifecycle of a product, thus PDM is part of PLM. The focus on these previously mentioned phases mean that there needs to be a good interaction with 2/3D CAD/CAM software, as these are the phases where PDM software is used. Furthermore, in the design, engineering and production phases, parts or assemblies are designed, revised, changed and/or approved, which means that these actions need to be well managed in PDM. It is important that the information belonging to parts and assemblies is correctly saved and transferred. Each unique item and/or item version must be identified as such, meaning that the PDM package needs to prevent users from copying an assembly and changing it but saving it under the same name.

Another software system that manages processes within a company is Enterprise Resource Planning (ERP). Different company processes make use of the ERP system: sales, human resource, finance, storage, etc. All these processes and their accompanying data can be integrated in one system. This integration of data means that data from PDM also needs to be integrated to the ERP system as some other company processes need to be able to use that information.

How is PDM linked to PLM and ERP? When designing an assembly there will most likely be parts of that assembly that need to be ordered. These parts are mentioned on the bill of materials, which is managed within the PDM system. Just as the information whether parts need to be ordered, have been ordered or if they are already delivered. The information from the bill of materials needs to be available for other departments that do not use the PDM system. The inventory management has information about the parts in storage, this needs to be compared with the information from the bill of materials. The parts that are not in storage need to be purchased which is done either by procurement or the inventory management. Maintenance data is produced during the engineering of the ship and can be part dependent, this data needs to be shared with the PLM system.

It is clear that PDM, PLM, and ERP are highly interconnected which makes it hard to separate the subjects. This research will only focus on the PDM-CAD/CAM connection because it lies closest to the actual engineering actions during the shipbuilding process. Furthermore as described before PDM is part of PLM which means that the use of PDM should be optimal before an optimal PLM solution can be developed.

## 1.2 Research question

What is the current situation of, the use of PDM together with CAD/CAM software? And how should a concept solution that meets the requirements of an optimal integration collaboration between PDM and CAD/CAM software take shape?

### 1.2.1 Sub questions

1. What is the current role and impact of PDM systems on ship design, engineering and production processes?
2. How are commonly used PDM systems in the shipbuilding industry suitable to use in combination with CAD/CAM systems and what are the needs and expectations of shipbuilders with respect to the connection to, or integration with, their existing CAD/CAM packages?
3. What are the requirements for an optimal integration/collaboration between CAD/CAM software and a PDM-system or strategy? And by what factors can a successful integration/collaboration be measured?
4. Design a concept solution that meets the requirements of an optimal integration/collaboration. Test the designed solution using NUPAS-CADMATIC as CAD/CAM system.

## 1.3 Methodology

A literature research has been done to investigate PDM in theory and because PDM is already used in the shipbuilding and other industries the major existing PDM packages have also been researched to get a grip on the current state of the use of PDM. After this, a choice for modelling language is made, which is described in Chapter 3. The selection of a modelling language was necessary for the development of a PDM solution, because there was no existing shipbuilding process available and therefore a generic process was modelled, as described in Chapter 6. The generic engineering model is used as starting point for the formulation of the PDM solution and how this solution should be used in connection with the generic engineering process. Both the PDM solution and the usage solution of the solution modelled by this language which is described in Chapter 7 and 8. Because the described engineering process is generic and the connected PDM solution is a concept solution, quantitative measurement of the improvements could not be made. Instead a qualitative measurement of the improvements has been done, this measurement is described in Chapter 9. To finalize the research the PDM solution has been tested for its feasibility with the CAD/CAM software: NUPAS-CADMATIC hull and outfitting.

# 2 PDM packages

As explained before PDM is more a strategy than a system. Different PDM package suppliers have used that strategy and transformed it into a workable system that a company can use on its engineering/manufacturing process. The reason that not all PDM packages are the same is that the strategy on which the packages are based is not always interpreted the same. In the following part of this chapter an overview of some major and minor PDM package suppliers will be given.

## 2.1 Siemens Teamcenter

The Teamcenter package of Siemens is a PLM package that has the ability to do PDM. The PDM part of Teamcenter is called Rapid start PDM. According to Siemens the PDM package contains the following processes:

- Product knowledge capture
- Visualization
- BOM management
- document management
- electronic workflow

PLM will extend that content to: change management, global collaboration, program and project management, procurement, compliance, portfolio management, manufacturing, and systems engineering. Teamcenter makes use of ISO-standard JT-files which means that through a 3D-viewer analysis can be done on 3D assemblies without requiring access to CAD-systems. The PDM system with its processes addresses the following business needs.

**Improve productivity and reduce cycle time:** The PDM system ensures that users have access to the right information at the right time, by making sure that the product and process information across the information is synchronized. This synchronization tries to reduce time searching for the right information.

**Reduce lifecycle errors and costs:** the provided visibility to accurate up to date information reduces errors which lowers the cost of correcting them. The single source of product and process information allows multiple applications access to the correct information without the probability of duplication errors when information foes from application to application.

**Facilitate collaboration with anyone anywhere**: The system makes sure that individuals are working on the correct versions of the product information. Which individuals are doing work to certain information is visible to other across the organization.

**Improve value chain orchestration:** Processes that involve internal and external participants are consistently managed. The changes in the product and requirements, that affect these parties are well managed by aligning the processes that cross the value chains of all the parties.

**Provide greater visibility to products and processes:** The product and process data, including CAD data, parts information, documents, requirements, 2D and 3D data etc. is made visible to users across the organisation. This means that the users that see the data that is meaningful to them are seeing information that is up to date and accurate.

## 2.2 Solidworks Enterprise PDM

The PDM package of Dassault systems is called Solidworks Enterprise PDM. According to this package there are seven steps to intelligent PDM.

**Intelligent data relationship knowledge:** the data that is stored within the vault tells you it's relations to other data e.g. when a part in the data vault is selected, the information about the assemblies that contain it becomes available.

**Controlled check-out:** if a file is taken from the vault it can only be accessed when it is checked out from the vault. After this procedure the file will be locked under the users name and other users cannot change its contents.

**Globally known check-out status:** as explained before checked out files get a checked out status. This checkout status can be seen by everyone in the team working on the product containing that file. Thus everyone will know what other users are working on, thus being able to avoid redundant work.

**Absolute overwrite protection:** in extension to the controlled check-out the vault secures that when a user is making changes to a document, someone else cannot overwrite that. Furthermore a revision management is maintained by keeping track of all the revisions and logging the actions done to change the files.

**Controlled check-in:** when changes to a file are finished, the file need to be checked in to the vault. This makes sure that the file is not locked anymore and other team members can see that the file has been changed and has become available again.

**Intelligent version/history branching:** When checking files in and out over a time period it can happen that some changes made to the file are not as the designer would like and that these changes need to be reverted. The vault acknowledges and tracks the generation of four versions. The designer can go back to version 1,2, or 3 whenever he wants. This tracking is done automatically to prevent having different versions of the same assembly saved under different names.

**Simplified revision nomenclature:** to make sure that the names of the files stored in the vault are not complicated long names which reflect all features of the file. The file is given a basic name and the information about the features of the file is stored within the vault.

Not only is Solidworks enterprise PDM a big organized vault to store files, it also has the possibility to manage view and print about 250 different file types including major CAD formats, MS Office, images, and animations. To streamline the engineering process the package contains a visualization of the workflow. It shows the order of tasks and also weather some designs are approved or waiting for approval.

## 2.3 PTC Windchill PDMLink

The PTC Windchill is a PLM package that contains a PDM module called PDMLink. The key benefits of PDMLink according to PTC are described below.

**Improve product quality:** within all the levels of content, from final assembly to individual components the information is managed and stored in one central repository. This repository eliminates mistakes associated with duplication or incomplete data. Potential mistakes can be identified earlier in the design process and everyone within the organisation can access the same product information

**Reduce process errors and engineering rework:** To eliminate problems and/or delays caused by miscommunication or product changes, an extensive change management process is included in the package. It has the ability to fast track small changes and full-track major modifications. This basically means that making small modifications to the contents of files requires less work steps than big modifications. During modifications, automatic documenting revisions and iteration histories are kept to prevent communication problems when product changes are made.

**Support global product development:** PDMLink is designed to support the product development in a web-based architecture. This architecture can coordinate replicated databases around the world. This feature makes it possible for anyone within the organisation that is anywhere to have access to the database and contribute to the product development process.

Outside these key benefits the PDMLink package also manages the bill of materials and can identify differences between multi-level bill of material structures. PDMLink has the ability to store and view models from other systems than PTC-Creo. Other than the ability to store other CAD files, PDMLink can also be integrated with legacy and enterprise system such as ERP.

## 2.4 PTC Windchill PDM Essentials

The PDM Essentials package is a simplified product data management tool with focus on easy to deploy, easy to configure, and easy to use. It can enter data for most of the Popular CAD systems. The package can be very well used by non-CAD users as it provides easy access to the correct information.

**Better control:** making changes to the design can only be done by authorized users. All changes have to be approved and they are documented. A change becomes visible for downstream departments (e.g. purchasing and manufacturing) only when it has been approved.

**Improved engineering productivity:** the web-based architecture avoids the need to have the PDM package running on all the computers and is accessible from an internet browser. The architecture improves collaboration and distributed product development. It is possible to quickly find and reuse existing designs. When using PTC Creo all data management can be done directly from within PTC Creo.

**Reduce risk:** The risk of users working with the wrong files is reduced by making sure the correct files are easy to find. Minimizing delays and cost by assuring that manufacturing does not use wrong parts or drawing versions. Teams are able to work without the fear of overwriting changes made by others.

**Growth potential:** the PDM Essentials package is suited for the need of a smaller team and company but it can be used as a foundation for future requirements. The package is a basis for a future transfer to the complete PTC Windchill PLM package.

## 2.5 PDXpert

The PDXper package is a PLM package that uses PDM but unlike other packages there is no separate PDM module. PDXper gives ten reason why the package can increase sales revenues, reduce product cost, and lower administrative expense.

**Cut development time in half, and get to market faster:** the "free-form text search" will enable to find parts more quickly. In-process design can be reviewed while the designer is working on the concerning product time is saved. By storing in-process and finished files in a central server rework from file los is reduced. Within the package it is also possible to re-use previously approved designs. Re-using files and storing them on a central location may reduce errors on the bill of material.

**Shorten release and change cycles:** the centralised storage of product data allows reviewers to simultaneously access the data. A configurable change workflow will identify approving departments and show their approval sequence with their authorized viewers. Notifications about the reviewing are automatically send.

**Enjoy more comprehensive, yet less intrusive collaboration:** all product plans, drawings and procedures for procurement, production, inspection, service, repair, and disposal are immediately available when they are created. This means that there is no reason for other employees to ask for them.

**Increase production experience:** the ability to reduce change cycles means that products can start to be designed earlier. The available time to produce a product increased because by time saving in the engineering phase production can start earlier which can deliver predictable cost reductions

**Slash production rework and scrap:** an audit trail of all changes to the bill of materials will be kept. The change workflow as described before makes sure that product modification are reviewed before the release to production. The bill of materials are consistent and may include information about production and inspection processes.

**Minimize excess and obsolete (E &O) inventory:** by maintaining accurate bills of materials, the reviewed products before entering production, have fewer new parts because of re-use of existing parts. The re-use of existing parts keeps the E&O inventory less full and the need for a change in workflow is brought down.

**Streamline compliance tasks:** the PDXpert package supports commercial-grade documentation and system configuration reports like ISO 9000, ISO 10007, FDA Part 820, EIA/IS-649, and MILHDBK-61 which are standardized requirements on which a company can achieve a quality policy to meat standardized customer requirements.

**Assess and report on a product's environmental impact:** the task of calculating and report product material composition for a particular item on the bill of materials is simplified an automatized.

**Reduce process administrative and clerical costs:** the data management of the PDXpert package together with its workflow management and email notifications will reduce a paper administration. Because the workflow can be configured a well-controlled process arises. When the processes are well defined users can perform more tasks in less time.

## 2.6 CIMCO PDM

The CIMCO PDM package is a simpler PDM package than the packages explained above. CIMCO PDM provides a SQL database with no limit to the amount of files and file size. The data base does not work with folders but with levels that can have sub-levels. Production information from NC file data can be created, this file extension is common used for CNC-machines. The structure of the data base can be user-configurable. An extensive and flexible user and group permission administration is used to maintain the control on who can access and modify information. Furthermore CIMCO PDM has a seamless integration with CIMCO NC-Base, CIMCO Edit, and CIMCO DNC-Max.

## 2.7 PLM xpert, PRO.FILE

As the name implies PRO.FILE is a PLM package that is also able to do PDM. The challenges of managing product data that PRO.FILE says it can overcome are explained below in seven steps.

**CAD data management:** all model data is saved in an electronic vault with a controlled check in and out system to prevent conflict occurring during concurrent engineering. A high value integration of CAD and CAE supported multi-CAD environment makes PRO.FILE suitable for mechatronics (mechanic, informatics, electronics) engineering.

**Classification and managing file variances:** a search engine on the basis of functional search will make finding of parts and thus re-use easier. eCl@ss is a cross-industry product data standard for classification and clear description of products and services (The eCl@ss standard, sd). The classification and taxonomy functionality on the basis of eCl@ss will result in a uniform terminology. PRO.FILE supports a copy based administration to realize customer specific products by the use of copied standard parts.

**ERP integration:** data from bills of materials are automatically transferred to the EPR system. An ERP server connector offers the ERP users the necessary information for production, planning, and procurement. PRO.FILE will make sure that the production, planning, and procurement information is accessible for the designers.

**Mechatronics:** the PRO.FILE system supports different CAx systems the information generated by the different system will be brought together and PRO.FILE will product correct bills of materials. A built in process management will coordinate all the workflows and change will be automatically notified to the right parties.

**Product structure and bills of materials:** the bills of materials are automatically generated from CAD files and also kept up to date during the lifecycle. When bills of materials are finished they are automatically shared with the ERP system when they are approved. The bills of materials are compared with older products and their bills of materials and may be altered if necessary.

**Visualization:** the visualization possibilities of PRO.FILE make it possible for users to access documents from every location and make mark-ups and do change requests. The drawings made are unlocked for all the users in the organisation.

**Change management:** all change steps are documented, version and revision management makes sure the correct and most actual data is used in the complete process. Process management and compliance tries to secure process ending.

## 2.8 Aras Innovator, PDM software

Aras Innovator is a PLM package with the PDM software integrated in the package. According to Aras, their PDM package is the predecessor to PLM, it will organize, track, and control the product data in a central data base. The features of PDM within Aras are listed below.

Aras uses a secure online collaboration data exchange to make sure that re-use of data is possible across the entire organisation. The online collaboration and data exchange is also used by project and program management, engineering and design, quality assurance, procurement.

The configuration and change management make use of CMII- and CMPIC-certified best practices. Which are configuration management institutes that give out configuration management certificates.

The Aras PDM is integrated with the following features: bills of materials management, CAD file, document, and engineering change management. The use of global product development the possibility to work according to the lean philosophy is integrated as well. Furthermore the PDM is integrated with part traceability, product lifecycle management, project management, quality planning, compliance, requirements management, and supply chain management.

## 2.9 Auto-trol KONFIG

KONFIG provides a set of configurable software and toolkits specially to model dynamic product data and processes. The features of Electronic Data Management (EDM),product data management, and electronic workflow are integrated in the system.

**KONFIG architecture advantages:** the system is internet ready which means that a internet browser is enough to use the system. The EDM, PDM, and workflow are combined in one product which has the ability to have a 10 million-item data base. There is virtually no limit to class definitions. The creation of data and security model is possible at any time without programming or change in database schemas, but the robust toolkits support programmatic customization. A graphical workflow designer makes it possible for simple workflow creation.

**KONFIG Technology advantages:** the system is proven to be capable of being used by the complete enterprise. Not only is it internet ready it is optimized for the internet and intranet. The system supports standard Java web server load balancing technologies, Java servlets, Java serverpages, and tags. The system is easily internationalizable and built on an Oracle database which supports an advances distributed vault technology. Finally the system supports business process and workflow standards.

## 2.10 Cmstat PDMPlus

PDMPlus PDM system manages the complete flow of product data information. All product information is stored and maintained in a central database which is called the Database of Authority. The system provides a configuration history from initial development to field maintenance, and supports and orderly transfer of product information through each stage of the product life cycle for the entire organization. The PDMPlus system is equipped with a complete configuration management package which defines in-depth relationships between related product objects.

PDMPlus gives ability to find and identify all components of product associated documentation. In every phase of the lifecycle the status of product information is monitored and records can be produced for verification audits. Product configurations such as designed, built, maintained, tested etc. Can be tracked because of the build in workflow management. This workflow management makes it possible for a paperless change and document revision-roll process.

## 2.11 PDM package comparison

To give a clear overview of how the above described PDM packages compare to each other a comparison matrix is made which is shown in Table 1. Most of the information about the PDM packages is obtained from the free accessible websites that advertise the PDM packages. This made it difficult to give a good comparison of the features of the PDM packages as the majority of the websites only mention the resulting advantages when using their product. The comparison depicted by Table 1, compare the important features of the packages that are mentioned in the advertising. These features are system dependant, this means that they emanate from the PDM package and should result in improvements. Other than that these features show differences between the packages where features like data vaulting and workflow management are provided by every PDM package.

| | BOM management | Check in/check out system | Windows explorer coupled data vault | Web-browser ready | Visual (3D) way of working possibilitie s | Use of neutral format data |
|---|---|---|---|---|---|---|
| Siemens Teamcenter | X | X | | | X | X |
| Solidworks Enterprise PDM | X | X | X | | X | X |
| PTC windchill PDM link | X | | | X | | |
| PTC windchill PDM essentials | | | | | X | |
| PDXpert | X | | | | | X |
| CIMCO PDM | | X | | | | |
| PLM xpert, PRO.FILE | X | X | | | | |
| Aras Innovator, PDM software | X | | | X | | |
| Auto-trol KONFIG | | | | X | | |
| Cmstat PDMPlus | X | | | | | |
| Enovia | | | | | | |

Table 1, PDM comparison matrix

## 2.12 PDM overview

The previous section explains the functions of PDM as described by literature and features of software packages that either are a dedicated PDM package or have a PDM part within the software package. The different software packages execute their own interpretation of a PDM strategy. The goal of PDM package providers, by implementing this strategy in a software package is to provide a system that benefits the engineering and manufacturing process of a shipyard. The implementation of a PDM system is costly for a shipyard, so implementing should benefit a shipyard afterwards. How does a PDM system benefits a shipyard?

When implementing a PDM system shipyards want to make sure there is something to gain when the costs of implementing have been made. Looking at the functions described in the first part of this chapter the key feature of a PDM system is data storage. However, data storage on its own is not a feature from which a company will benefit because every company in the world stores data. This data storing which is called data vaulting in this chapter is done according to a strategy on which the use of the stored data goes more efficient than before. A well-organized data vault by the use of workflow management, product structure management, parts management, program management, and configuration management as explained above, may result in the effectuation of the ultimate goal for a company to gain higher profits. Most of the gains are in the form of time reduction, time is reduced because documents that engineers need to work on are found more quickly. The engineers can use a search function or find documents quicker because of an organized data structure. Within this data structure the use of parts management will make clear the assemblies and the parts they consist of. If those parts and assemblies are correctly stored they can also be re-used which then saves time because they don't have to be re-engineered. Not only is data just easier found but by the use of workflow management it is visualized who is working on a certain document and who has to work on it next. This also makes it visible to every user who has worked on which document, making it easy to know which user to ask should one need information. Not only is made visible which user is working on which document, but also the status of the document. When a document has a visible status it becomes clear for users what is the next task to do on that document. These clarifications on the tasks that need to be done to documents and which user needs to do them streamlines the engineering process which results in time saving, not only by speeding up the process but also by reduction of errors. To extend data vaulting program management adds tools like work breakdown structure, scheduling, and planning to track the engineering process. With the organized data structure makes it easier to track the engineering process and more errors are prevented.

PDM itself has advantages mostly in the form of time savings in the engineering department. But how do these advantages come to justice within existing PDM packages? Do all the PDM packages realize those advantages? Some features of PDM are contained by every PDM package, off course as data vaulting is the basis of PDM every PDM packages has a data vault. The data vault itself is not always the same, the one package uses a windows explorer environment the other has its own environment. Depending on a company's preference it is determined what the best data vault is but a data vault with a windows explorer environment should be easier for users to get to know, as it is a well-known environment. Even though not all PDM package providers advertise having a search engine within the data vault it is fair to say that within every data vault there is to possibility to search for certain documents.

Users of a PDM package need to get documents from the data vault and start working. Every package has its way of visualizing information about whom and when someone worked or is working on a document. Also, the PDM package has the functionality to give access rights to different users. In a manufacturing work floor environment it is common to make use of Bill Of Materials (BOM). Therefore every PDM package has some form of BOM-management. This BOM-management always goes in collaboration with parts and product management.

It may be clear the not all the PDM packages are the same. There are differences between the PDM packages mostly related to the magnitude of the packages and the links these packages have with PLM and ERP. The bigger and better known packages make use of a 3D-viewer which is something that other packages don't use. Some packages even have the possibility for engineers to do analysis on 3D-moedels from the 3D viewer. As every PDM package has some kind of workflow management some packages go one step further. The product data is synchronized with the process data and they also try to align their workflow with the workflow of third parties. A difference of the packages within the data vaulting is not only the environment. Some packages require a document to be properly checked out of the data vault before being altered. After alteration it needs to be checked in. Other packages do not have that requirement but will lock a document when it is in use. Finally some packages are internet based packages which makes it possible to work from different locations requiring only an internet browser.

# 3 Comparison of modelling languages for describing engineering process

This chapter gives an overview of existing modelling languages and techniques and finally argues the choice for Unified Modelling Language (UML) to make a complete as possible model to map the information flow during the engineering phase of a ship.

With the use of CAD/CAM software during the engineering process of products, information is constantly generated. PDM systems are used to manage the information generated in the engineering process. Furthermore, this information needs to be transferred to and from different persons and systems participating in the engineering process, the information flow and the participants of the engineering system will be discussed in Chapter 6. To get a good picture of the kind and amount of information flowing from system to system, system to person, or person to person a fitting model needs to be made to map all those information flows, as a representation of a real life system to study the effects of PDM to it. To make an information model of a certain phase of the engineer of a ship the correct modelling language needs to be chosen. The goal of the research towards a selection of different modelling languages/techniques is to find a suitable modelling language that can describe and visualize the process in a way, that it is possible to identify and understand the problems occurring when a PDM system is introduced in an engineering process. A process model that describes information flows is necessary for this research because it needs to become clear where the connections with PDM are and if the PDM benefits the engineering process. The chosen modelling technique/language will be used to model an engineering process and PDM solution which is described in Chapter 6 and 7. This model should be able to reflect the connection between the physical process of engineering and building a ship and the information that is generated and shared by the participating people and systems. For this research it is important to select a modelling language that is able to handle the description of only the engineering process instead of describing the engineering process in context of the complete shipbuilding process. The modelling language must handle a great amount of different information flows, moreover the model needs to be with such a perspective that it is quickly understandable for others that have no knowledge of the complete context of the model.

Several existing models, all with their own advantages and disadvantages, have been analysed and investigated. Within the boundaries of this research, the model should be suitable to visualize a delimited part of a complete ship engineering process. In other words, the model will be of the engineering and not design and production. The modelling language/method needs to be very flexible and not so much fixed to a modelling construct which needs a complete business process. The engineering process to be described contains a lot of information flows, therefore it is useful that the modelling language makes information flows as opposed to flow of mass or entities because an extensive description of what kind of information is described by an entity will be unnecessary. In a visualization of a process modelling, information can flow both ways where mass or entities can mostly flow in one direction only.

## 3.1 DEMO

Dynamic Essential Modelling of Organisation (DEMO) is a theory about the construction and operation of organisations, that is rooted in the communicative action paradigm regarding human communication and action (J.L.G. Dietz, 2001). Combining communication and action within a business model DEMO has four core concepts: communication, information, action, and organization, as shown in Figure 1.

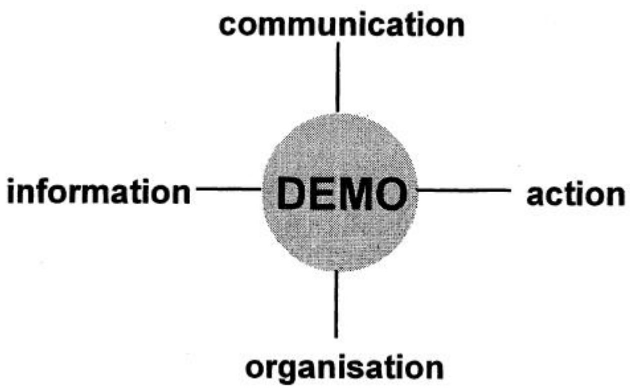



Figure 1, DEMO's four core concepts: communication, information, action, and organization

### 3.1.1 Communication

Communication consists of the sharing of one thought between two subjects, where the produced language expression is called information. The DEMO model distinguishes six kinds of communicative acts: question, assertion, request, promise, statement and acceptance. Thoughts are defined as triples <I,F,T>, where "I" stands for Illocutionary kind, "F" for Fact (i.e., an elementary state of affairs), and "T" stands for Time. The I can be a question, for example "do you have water?" The F is in this case the answer "you have water", or "you don't have water". The T represents the time period for which F is the case.

### 3.1.2 Information

Information is closely related to communication; there is no information without communication since information is produced only for the purpose of communicating. Information has three aspects: forma, in-forma and per-forma. Forma: means that it has some perceivable structure carried in some physical substance. In-forma means that to every forma belongs at least on in-forma. The in-forma is the meaning of the forma. Per-forma means that a piece of information is the effect on the relationship between the communicating subjects, caused by communicating thought. These aspects correspond with three information management levels which can be seen as glasses through which one can look at an organisation: essential, monitoring per-forma's, informational, processing in-forma's, and documental, handling forma's.

### 3.1.3 Action

DEMO distinguishes between essential actions, informational actions and documental actions. The most important are the essential actions, because these are the actual business actions, whereas the other actions serve to support them. The class of essential actions is further divided into objective actions and social or intersubjective actions. By executing objective actions, the members of the organisation fulfil the mission of the organisation. By executing intersubjective actions, subjects enter into and comply with commitments. Objective actions and their related intersubjective actions appear to occur in a particular pattern, called the (business) transaction. A transaction consists of three phases: the order phase or O-phase, the execution phase of E-phase and the result phase of R-phase

(OER-notation). In correspondence with the distinction between objective and intersubjective actions, DEMO distinguishes between two worlds in which each of these kinds of actions have effect: the objective world and the intersubject world.

### 3.1.4 Organisation

When describing the concept of organisation, DEMO takes an engineering point of view and uses the following definition of an organisation: *Something is an organisation if and only if it fulfils the next properties: it has composition, it has structure and it has a boundary.*

Analogous to the technique of functional (de)composition, a complex system is replaced by a structure of more understandable smaller systems. A white-box model of a system has system elements that interact on the one hand with the object world and on the other hand with the system world. There is a technique for composing and decomposing white-box models of a system. It is called constructional (de)composition. A central concept in the technique is the concept of subsystem. This concept is in DEMO defined as follows: S2 is a subsystem of S1 if and only if:

- S1 and S2 are systems according to the definition above.
- The kernel of S2 is a subset of the kernel of S1.
- The structure of S2 is a subset of the structure of S1.
- The environment of S2 is a subset of the composition of S1.

In conclusion, functional (de)composition and constructional (de)composition are similar techniques, but applied to very different system notions.

### 3.1.5 The DEMO methodology

As described above, DEMO is a business process modelling methodology. That offers a conceptual framework suitable as a basis for modelling information systems. Figure 2 gives a representation of the DEMO business process.

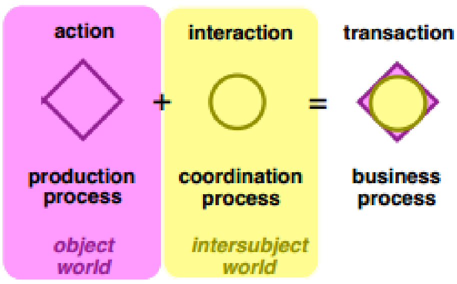



Figure 2, DEMO Business Process

Expanding this business process and placing it in the OER-transaction pattern as described in the action paragraph, a DEMO construct would look like the one in Figure 3. The figure shows a transaction. In the intersubject world, within the O-phase there is a conversation between Initiator "I" and Executor "E", red arrows represent communication of the initiator and blue of the executor. The result of the transaction is discussed in this phase. When there is an agreement between the two parties, state two is reached. The Executor will take necessary action for the transaction in the execution phase. In the result phase the actors have another conversation discussing the result of the transaction.

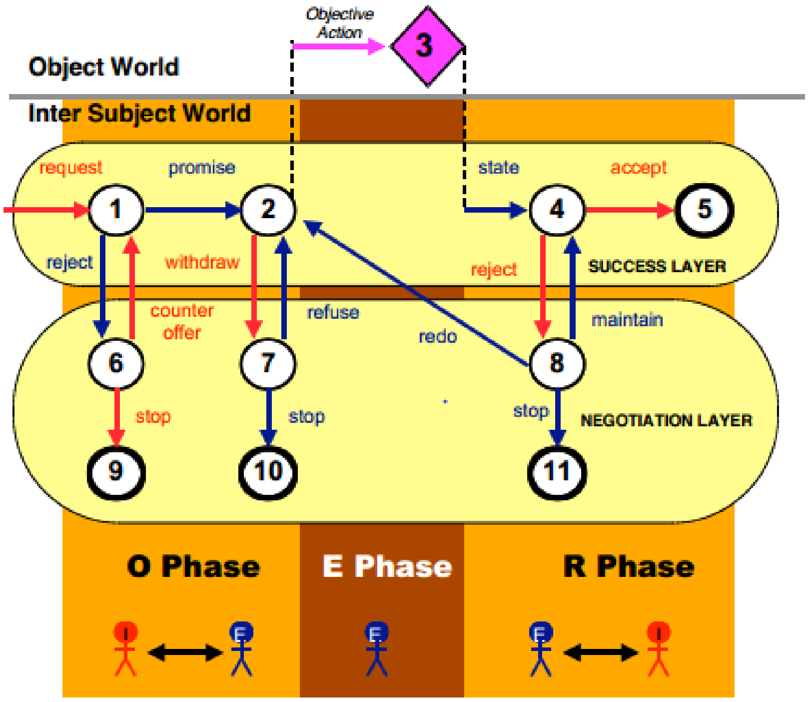



Figure 3, OER-Transaction Pattern Model

The DEMO framework is a good way to describe information systems where a lot of communication between persons is present. Notable is that most of the information flows described are in the inter subject world and that the object world lags behind. Because in the object world the actions are executed the world operates on a different tempo therefore splitting the information flows and the actions in two world.

## 3.2 Delft systems approach

In the book: The Delft Systems Approach, Analysis and Design of Industrial Systems by H.P.M. Veeke, J.A. Ottjes, G. Lodewijks a system is defined as a collection of elements within the real world and the methods a researcher wants to use for these elements to make the system discernible. These elements have mutual relationships and (eventually) relationships with other elements from everything in the universe “total reality”.

In the Delft systems approach, a system consists of elements (objects, components, entities) which are the smallest parts considered by a researcher in view of his goals. The sum of the collection of elements is called the content of a system. The elements of a system have certain properties such as physical, geometrical, aesthetic, social, etc. called attributes. If elements interact with each other, they have relationships; a collection of relationships is referred to as the structure of the system. The system representing a group of elements lies within the universe, known as the total reality. The elements of the universe under consideration of the system are within the environment of the system.

To make complex systems more clear, the Delft systems approach differentiates the system into subsystems and aspectsystems. A subsystem is a partial collection of elements in the system whereby all the original relationships between these elements remain unchanged. An aspectsystem is a partial collection of relationships in the system whereby all the original elements remain unchanged.

Furthermore the systems within the Delft system approach contain basic system concepts like: state, process behaviour, goal, function, task, and boundary conditions.

There are many reasons to model a system that can be eventually formulated in terms of problem solving. To make a correct model, a root definition of the problem and the surroundings should be made. A correct root definition satisfies the Customer, Actor, Transformation, Weltanschauung, Owner, and Environment (CATWOE) principle. A definition of a problem satisfies the CATWOE principle when the Customers are explicitly stated. The Actors are of the concerning activity are described, the Transformation performed by the activity and its World view (Weltanschauung) is described. Finally the Owners and the accepted preconditions of the Environment need to be identified. The root definition is used to construct conceptual models. A strong feature of these conceptual models is that they contain a control loop either feed- back or –forward loop.

Conceptual models that are described within the Delft systems approach are described below. The conceptual models consist of "functions", which means that they are black boxes with an input and a transfer function that changes the output. Different ways of structuring these boxes gives the conceptual models.

**Formal system model**: represents a human activity system. To be a formal system a system must have: some mission, some measure of performance, a decision making process, mutual interaction between the elements, a wider system or environment, decision making resources, and stability of ability to recover.

**Viable system model**: consist of functions to be present in any viable system. Viable systems all have the same pattern of functions. This pattern should not be considered an organization structure but a function structure.

**Steady state model**: it consists of a structured set of functions, expressing which contribution is repeatedly delivered to the environment in a controlled way but not how this contribution if achieved in a concrete way. The set of functions is itself a function again, which makes the model recursive.

**PROPER model**: the whole system model function delivers some kind of performance and therefore the model is called PROcess PERformance model or "PROPER" model. The performance will be expressed in terms of productivity, effectiveness and efficiency. These criteria are used during real operation of the system, but also during the design of a system to evaluate design alternatives.

The PROPER model, which can be used by all disciplines, is a strong model, however the focus lies on logistic processes or process which contain actual material. The flow of information is not as well described as flow of material and the measurement of effectiveness and efficiency is much harder for information flow as the efficiency is measured on the amount of mass/entities going in and out over time. Even though information can be described as entities it is hard to give it a value compared to other entities that describe a physical object which has for example a mass as a quantification measure.

## 3.3 ARIS

Architecture of integrated information systems, The ARIS-house of business engineering encompasses the whole life-cycle range: from business- process design to information- technology deployment, leading to a completely new process-oriented software concept.

Within the ARIS model, there are two fundamental ways of (re-)engineering information systems: the formal driven approach, based on the goals of developing and implementing a technical correct running system, and the content driven approach, based on the goals of developing and implementing an organizational correct running system. ARIS can be seen as a way to describe business process with the goal to add value to an organisation. This process is modelled using event-driven process chains.

ARIS-House of business engineering (HOBE) enhances the ARIS process architecture by addressing comprehensive business process management, not only from an organizational, but also from an IT perspective. Figure 4 gives a visualisation of HOBE, with its four phases.

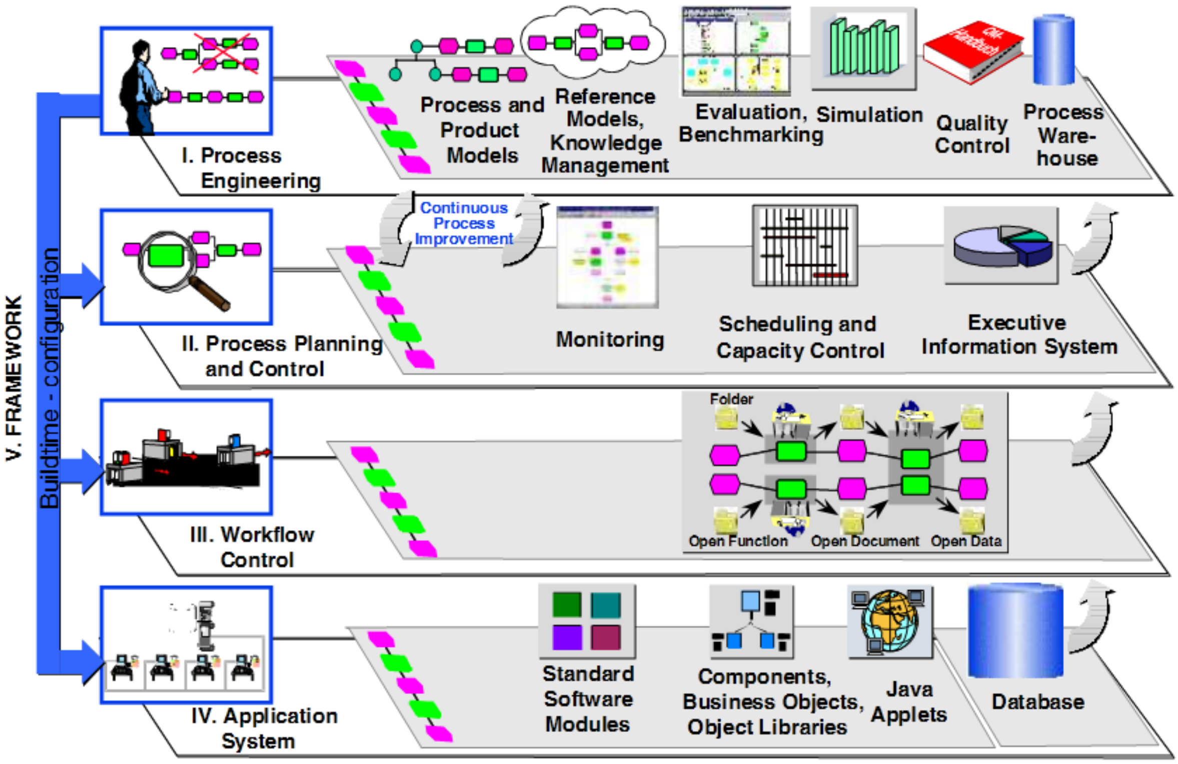



Figure 4, ARIS-House Of Business Engineering (HOBE)

**Level I, process engineering:** The manufacturing work schedule determines the models for which ARIS provides a framework. This framework is able to fit various methods for optimizing ,evaluating, and ensuring quality of the processes.

**Level II, Process Planning and Control:** This level is meant for business owners and process managers to monitor, plan, control and schedule various processes in accordance with suitable methods

**Level III, Workflow Control:** The workflow control stores objects and information that need to be processed. It makes sure that the stored objects and information is sent to the right receiver through the right route.

**Level IV, Application System:** Information that needs to be processed before it reaches its destination goes through an application system. Furthermore this level contains all the computer-aided application systems.

The four levels of HOBE are independently connected and form a framework for managing business processes. It is important to note that a variety of systems is used to cover the whole process, and that none of these systems is capable of determining the status of the entire process. The systems used for running a business process together with the reference models to describe the business process should fit within the framework described by ARIS. The ARIS-House of Business Engineering describes business process by using multiple modelling techniques and only combined it can form a complete business process

## 3.4 Bunge-Wand-Weber ontological modelling constructs

Wand and Weber have developed a series of models based on the ontological theory of Mario Bunge, the Bunge-Wand-Weber (BWW) models. Ontology is a strict conceptual scheme/data structure of all relevant entities and their relations and rules within a domain. Wand and Weber have developed and refined a set of models for the evaluation of modelling grammars and the scripts prepared using such grammars. These models are based on an ontology defined by Bunge and referred to as the BWW models. Two concerns have arisen during the development of the ontological BWW models: lack of understandability because of the use of rigorous set- theoretic language, and lack of comparability, which arises because the grammars used for he set-theoretic definitions are loose definitions.

The relationships between the constructs, things, properties of things, and attributes of things are described in table 1. The world is made up of things. We know things in the world via their properties. All things have properties; there are no property less things. It is assumed that things and their properties exist in the world.

| Ontological construct | Explanation |
|---|---|
| Thing | A thing is the elementary unit in the BWW ontological model. The real world is made up of things. Two or more things (composite or simple) can be associated into a composite thing. |
| Property<br>In general<br>In particular<br>Intrinsic<br>Non-binding mutual<br>Binding mutual<br>Hereditary<br>Emergent<br>Attributes | Things possess properties. A property is modeled via a function that maps the thing into some value. For example, the attribute "weight" represents a property that all humans possess. In this regard, weight is an attribute standing for a property in general. If we focus on the weight of a specific individual, however, we would be concerned with a property in particular. Other properties are properties of pairs or many things. Such properties are called mutual. Non-binding mutual properties are those properties shared by two or more things that do not "make a difference" to the things involved; for example, order relations or equivalence relations. By contrast, binding mutual properties are those properties shared by two or more things that do "make a difference" to the things involved. A property of a composite thing that belongs to a component thing is called an hereditary property. Otherwise it is called an emergent property. Some properties are inherent properties of individual things. Such properties are called intrinsic. Attributes are the names that we use to represent certain properties of things (normally abstract properties). |
| Class | A class is a set of things that can be defined via their possessing a characteristic property. |
| Kind | A kind is a set of things that can be defined only via their possessing two or more properties. |
| Coupling<br>Binding<br>Mutual<br>Property | Two things are said to be coupled (or interact) if one thing acts on the other and vice versa. Furthermore, those two things are said to share a binding mutual property (or relation); that is, they participate in a relation that "makes a difference" to the things. |
| System | A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets. |
| System composition | The things in the system are its composition. |
| System environment | Things that are not in the system but interact with things in the system are called the environment of the system. |

**Table 2, ontological construct and explanation**

An existing language is used to form a model according to the BWW construct, to form a so called metamodel. This model uses both the rules of the language and of the BWW construct thus forming models that are suitable for different applications.

## 3.5 Extended entity relations-model (eERM)

To tackle the two concerns mentioned above, lack of understandability and lack of comparability, a metamodel in the form of entity relations approach is used. Three aspects of the selected language used should be stressed:

1. Minimum-maximum cardinalities, which are used for further characterisation of the relationship types. It states how many (at least and at most) cases of an entity type participate in the relevant relationship type.
2. Relationship types can have own attributes (e.g., the price in a relationship type between car and customer) relationship types can be part of a relationship type. Thus such relationship type has the characteristics of an entity type.
3. Recursive relationships describe a relationship type that an entity type has to itself. Such relationships are used mainly to depict is-part-of relationships.

### 3.5.1 Entity relationship diagrams

Entity relationship diagrams are used to visualize a data models. This visualization helps identifying information in data systems. The diagrams not only identify information but also give a representation of the rules connected to the information within the data system. The components of an entity relationship diagram are entities, an entity can be seen as an abstract object, i.e. a car or a person but also an area or an event. These entities need to have a clear and singular name. Properties of entities called attributes, an employee has a name and/or an employee number. If entities are linked they have relationships, i.e. an employee and the workplace. In this case the relation would be "works' when two entities have more relationships the cardinality of the relationship goes up. Relationships can have a cardinality of 0, 1, or n.

## 3.6 Unified modelling language (UML)

The UML is a visual modelling language for specifying, visualizing, and constructing software systems. It unifies the object-oriented methods of Booch, Rumbaugh, and Jacobson (Oh, Han, & Suh, 2001).

During the modelling of the process and the data, UML provides relevant diagrams for describing the link between the object to be implemented in the system and the data handled in the company. This step is very useful because, if every link is clearly identified and specified, the implementation step will be easier and faster (Eynard, Gallet, & Roucoules, 2006).

UML models can raise the level of abstraction by hiding or masking details making sure the bigger picture of the models is not lost. If details cannot be hid or masked, the use of aggregation layers with different levels of detail can be used. To show the environment where an application is executing it is possible to zoom out to higher aggregation layers. Zooming out enough, the business process that the application is automating becomes visible. This makes it possible to nest new modelling elements. Using the thirteen standard diagram types of UML, models of software systems can be specified, visualized including their structure and design. It is also possible to use UML for business modelling and other non-software systems.

The thirteen types of diagrams are divided in three categories. Six of the diagrams represent a static application structure, three represent general types of behaviour and the remaining four represent other aspects of interactions:

**Static application Structure Diagrams**: Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram.

**General types of Behaviour Diagrams**: Use Case Diagram, Activity Diagram, and State Machine Diagram.

**Other aspect of Interaction Diagrams:** Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram. (Introduction To OMG's Unified Modelling Language (UML), 2015)

UML consists of a lot of different diagrams which do not necessarily need to be used together but they can supplement each other. The diagrams used to describe the engineering process and the connecting PDM system will be described in Chapter 6.

## 3.7 Conclusion

As mentioned in the introduction UML has been chosen as modelling language. The system to be modelled is a delimited part of the shipbuilding engineering process. This delimited part may be non-chronological, and other phases of the engineering process may be pulled in-to the delimited part to make sure the model is as complete as possible. This means that a very flexible modelling language/technique is necessary and, moreover, that a modelling language that focuses on a complete business process will not suffice. These models like ARIS, DEMO, and BWW are more a framework to describe businesses with the use of different modelling languages. Taking all this into consideration, a modelling language like UML is more fitting. By use of the different diagrams UML describes different information flows in different ways, where a defined information entity always remains the same. Which means that every different information flow should have its own specified information entities. Finally, the model should contain different aspects: information flow, data storage, activities, etc. which can be easily described by make use of the different diagrams types. The aspects that make UML a suitable modelling language/technique for describing the engineering process of shipbuilding is summarized below.

- The diagrams of UML do not make use of flowing entities, even though information can be described as entities it is unfeasible to described all the different information entities that will be necessary in the information process description of the engineering of shipbuilding and the use of PDM.
- UML is not bound to a business framework. One part of a complete business will be modelled and the focus should be on that part, the engineering process and not the contribution of that part to a complete shipbuilding process.
- The flexibility of the different UML diagrams makes it very suitable for the description of a fragment of a complete process which is the engineering process within the shipbuilding process.
- Getting to know the necessary UML diagrams is not time consuming because the complexity does not lie with the modelling language but a user independently decides to add complexity.
- A suitable modelling environment (Enterprise Architect) was available at Numeriek Centrum Groningen.

## 4 Ship engineering activities

The modelling language that will describe the engineering process will be UML. Before modelling can start the structure of the shipbuilding engineering process and the extend of the model should be determined.

The model of the engineering process needs to contain the main engineering activities. Figure 5 gives a global overview of the phases that make up the engineering process until the production. From the detailed engineering phases the engineering is focussed on zone of block oriented engineering. At these phases the engineering for production also starts. The pre contract design and systems engineering is focussed on the performance of a certain ship, which means that the information generated at these phases are requirements and the detailed engineering can shape those requirements into actual systems and steel structures.

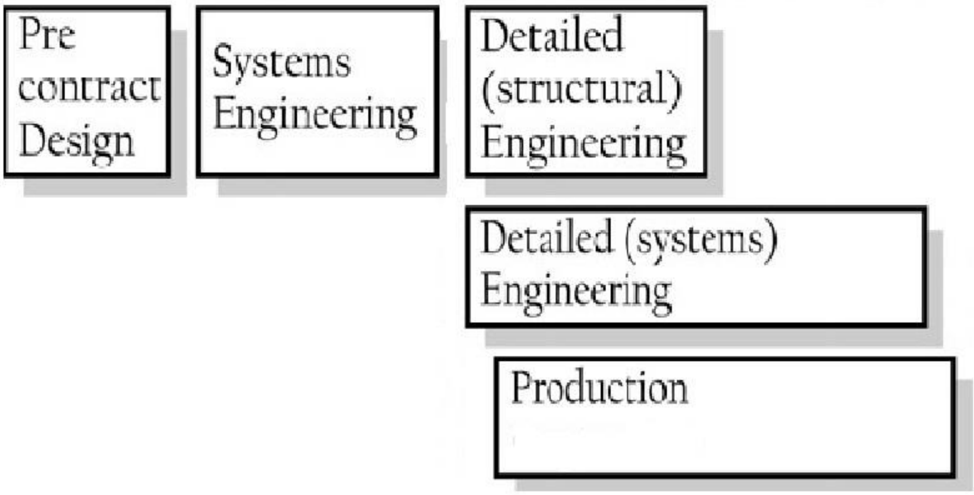



Figure 5, basic design/engineering sequence

Figure 5 presents a picture of the global stages in shipbuilding and the containing engineering phases. These engineering phases contain off course different engineering activities. Figure 6 shows the main activities within the engineering process, both system and structural engineering. A hull form in the form of a lines plan is provided by pre contract design and systems engineering. This hull form is engineered in to a structure.

The ship is made out of blocks and those blocks are made out of sections. Within a ship there are zones identified, these zones relate to the layout of the ship e.g. compartments where main components of the ship are placed. All the different outfitting phases are divided in main and secondary. This split depicts the difference in priority of work needing to be done. This means that main out fitting phase will result in pre-outfit drawing and that secondary outfitting phases will result in the outfit drawings. This difference in priority results from the different moments of production. pre-outfitting describes the outfitting work before hull erection, outfitting describes the outfit work while and after hull erection. All these concurrently executed engineering activities result in a 3D-model of a ship, in the case of this research a block of a ship.

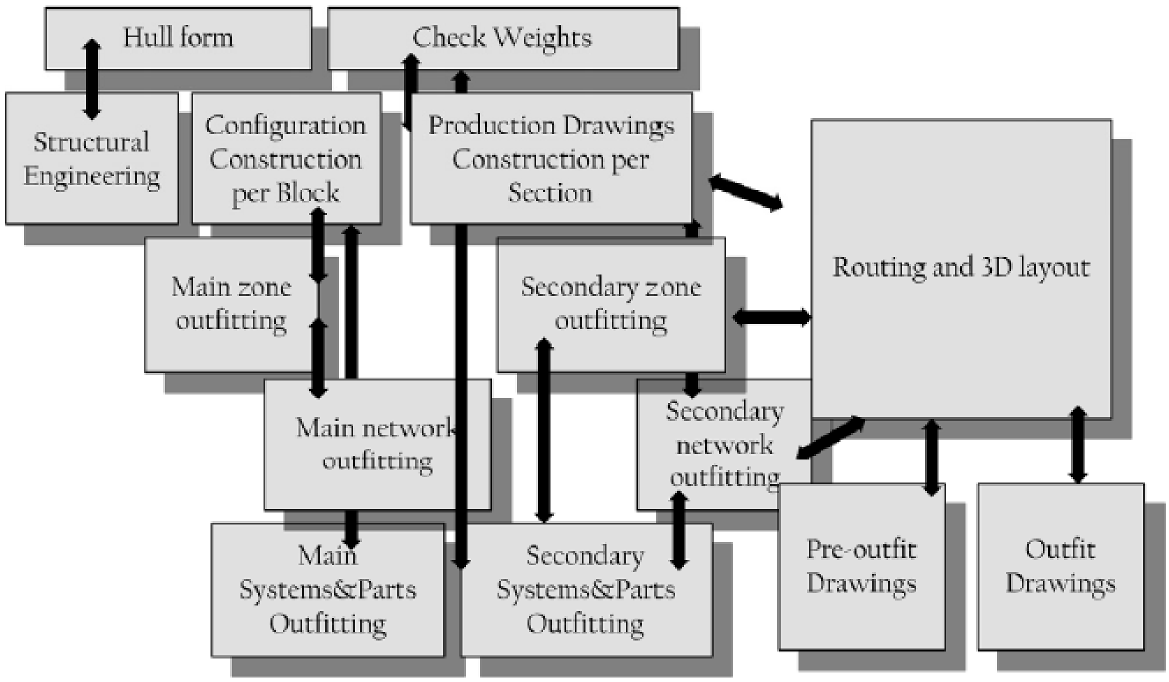



Figure 6, relationship between sub-processes in ship-engineering

The activities shown in Figure 6 are generic and do not show the activities related to the parts that are in the block considered for this research. The parts from the part list of the previous section will now be assigned to the activities of Figure 6.

- Hull construction, evidently these parts belong to the horizontal row of structural engineering, configuration construction per block, and production drawings construction per section.
- Systems engineering
    - Pump (mechanical), considering this pump as the main component of the section the pump will belong to the zone outfitting as it is a big component which influences the layout. When pipes of substantial sizes are to be fitted to the pump can influence the layout of the zone the pump is in. These big pipes will be engineered by the collaboration of zone and network outfitting. The routing of the pipes that belong to the pump need to be done in collaboration with network outfitting.
    - Switchboard (electrical), switchboard can vary in size but as a switchboard has low weight and is not bound to a certain location in the zone. This means the placing of the switchboard does not cause structural constraints. The switchboard belongs to the systems & parts outfitting.
    - Airco-unit (HVAC), as the air ducting of an airco-unit has a substantial size. The routing of the ducts needs to be taken care of thoroughly and on an early stage of engineering. The airco-unit and its ducting are within the network outfitting phase.
- Outfitting
    - Pump, the pumps foundation and driptray are to be considered in the systems & parts outfitting. The drain plug and the bolts, nuts require holes in the steelwork but the parts itself are not produced but are in stock or need to be ordered.

- Electrical, all the electrical wires that are in a ship lay in a cable tray. This tray is routed through the complete ship. This routing is done by network outfitting.
- Other, all these typical outfitting parts belong to the systems & parts outfitting. Some of these outfitting parts like hatches and manholes require holes in the construction. This means that it has to be done in collaboration with structural engineering. The making of holes can always be done afterwards bur the change of compromising the strength of the construction is present.

As sometimes mentioned above the different activity blocks need to collaborate on outfitting parts that do not really belong to one outfitting block. The same goes for outfitting and hull construction, the more pre outfitting in the production of a ship, the more the systems & parts outfitting and hull construction need to collaborate.

Just as the generic shipbuilding phases of Figure 5, are the activities in Figure 6 also quite generic. The structural engineering, configuration construction per block, and production drawings construction per section there are ship-building engineers working on the hull structure producing a 3D model of the hull construction and production information. Within all the outfitting blocks it is very much depending on the kind of component to be installed, which work needs to be done. The outfitting of a ship is a collaboration of mechanical, electrical, and ship-building engineers making 2/3D drawings of the components with their belonging outfitting parts (foundations etc.). This collaboration is very important because all systems and different routings interact.

## 4.1 Virtual block of ship.

To design an optimal integration between the CAD/CAM system and PDM package, the ship-engineering process with its workflow and steps need to be described. In order to not exceed outside the scope of this research the process described how to select a block of the ship as a case for this research that is sufficiently rich in its parts of required engineering processes, but not too complex for the purpose of this research. The following chapter gives an overview of the parts that are going to be in the section and the related activities belonging with engineering the section.

The model of ship-engineering should be delimited to a representative part of a ship. It was decided to look for a suitable block. Using only one, but representative block. The amount of required parts and engineering steps is brought down. Such a block needs to contain enough structural and mechanical parts to make the section representative of a whole ship. The focus of the model lies with the engineering of the block. The start status of the model is that the block is already pre-designed in a way that the requirements are finished.

### 4.1.1 Parts of the block

The engineering phase of a ship is split in two sub-systems hull construction and systems engineering. Therefore the description that describe the parts and components of the block will also be divided in those two sub-systems. A third part which is called outfitting is used for parts that either belong to hull construction and systems engineering, or are not really part of any group. The block is not an existing block from a ship. The components of the block are chosen in a way that the block represents a very small ship. Of course, a complete ship would have more components but in the case of the hull construction there would only be more of the same parts, which means this represents a complete ship very well. A real ship would have more systems than is under systems engineering below. On the other hand, these systems contain all the necessary disciplines. The virtual block consist of:

- Hull construction
    - Shell plates
        - Straight
        - Bended
    - Tanktop
    - Bulkhead
    - Girders
    - Floors
    - Framing
    - Brackets
    - (stringer)
- Systems engineering
    - Pump (mechanical)
        - Piping
    - Switchboard (electrical)
        - Cabling
    - Airco-unit (HVAC)
        - Ducting
- outfitting
    - Pump
        - Pump foundation
        - Driptray
        - Drain plug
        - Bolts, nuts
    - Electrical
        - Cable trays
    - Other
        - Piping/ducting mounting
        - Hatches
        - Stairs
        - Ladders/steps
        - Rails
        - Hatches
        - (watertight) doors
        - Manholes
        - Holes (holes for piping and cabling)

# 5 Top level use case diagram of design/engineering process

To investigate the use of PDM in a shipbuilding engineering process, a good description of the engineering process is made. From this description of the shipbuilding engineering process, the connections with PDM are identified, with their possible benefits and drawbacks. The virtual block that has been described before is now put into context with the use case diagram. The goal of the use case model is to have a description of a generic engineering process to investigate the consequences of the addition of a PDM system.

The use case diagram has four actors, a shipbuilding engineer, mechanical engineer, electrical engineer, and a ship owner (customer). These four are chosen as the actors shipbuilding engineer, mechanical engineer, and electrical engineer are required to fulfil all process functions, thus represent working profiles that are necessary to enable the complete process. The ship owner is not a working profile like the engineering actors but the input of design requirements from outside of the design/engineering process the ship owner represents the customer for which the ship is build. The actors are connected to four systems, global ship information, hull shape, hull construction, and outfitting system which means these actors have to execute the use cases. This system decomposition is based on the sub models of the ship information model (M. Welsh, J. Lynch, & B. Brun, 1992) shown in Figure 7.

This decomposition is chosen because the structural and outfitting information model are described separately, this means that the engineering process which belongs to these models is also described separately which makes it easy to focus on only the engineering. The separation of hull construction and outfitting corresponds with the NUPAS-CADMATIC hull and outfitting applications. The global information model represents the properties that characterize the ship and its functional capabilities. The shape and spatial organization in the ship is described by the spatial information model. A representation of the internal spaces and compartments and the moulded surfaces that bounds those spaces are defined in the spatial information model. The shape of the ship as described in the spatial information model is transferred into a construction in the structural information model. Finally, the ship and its spaces are filled with outfitting components and materials and this is what the outfitting information model is used for.

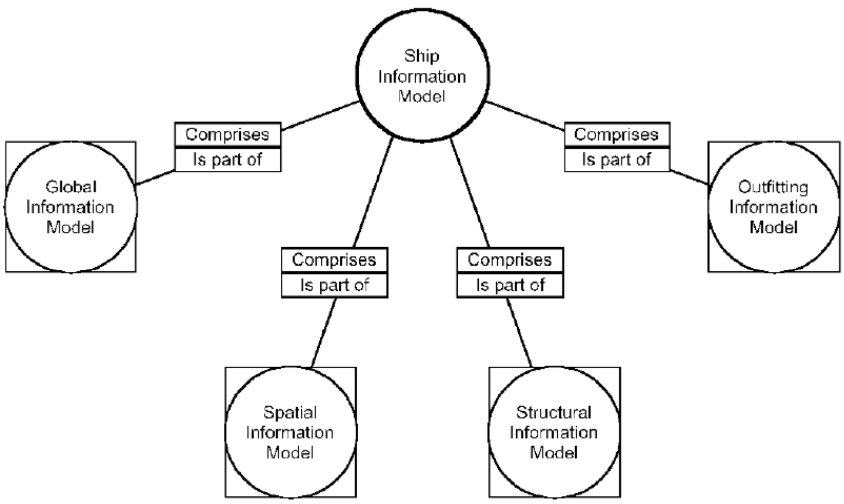



Figure 7, NIAM diagram of the NEUTRABAS high level model

This high level NEUTRABAS model which is modelling according to NIAM language is used as a structure for the description of the engineering process of shipbuilding. Only the top level is used because of its fitting structure. When zooming in a level deeper the use case model will differ from the NEUTRABAS high level model.

UML has many types of diagrams/models that help visualize the design of information systems. In this research three diagrams have been used. To model the engineering and accompanying PDM system a use case model is used, within the use cases activity diagrams explained the detailed actions executed by either a user of the system or a system functionality. To give a visualization of how the system takes shape a deployment diagram is used to visualize the systems architecture. The following conventions are used in the three UML diagrams.

- **Use case model:** the use case model captures the requirements of a system. Use cases are a means of communicating with users and other stakeholders what the system is intended to do.
    - **Actors:** a use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware of other systems. An actor is drawn as a named stick figure.
    - **Use cases:** a use case is a single unit of meaningful work. It provides a high-level view of behaviour observable to someone or something outside the system. The notation of an use case is an ellipse.
- **Activity diagram:** In UML, an activity diagram is used to display the sequence of activities. Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. Activity diagrams are useful for business modelling where they are used for detailing the processes involved in business activities.
- **Deployment diagram:** A deployment diagram models the run-time architecture of a system. It shows the configuration of the hardware elements (nodes) and shows how software elements and artifacts are mapped onto those nodes. (What is UML?, 2015)

Figure 8 shows the use case diagram containing the actors and their interaction with the four systems.

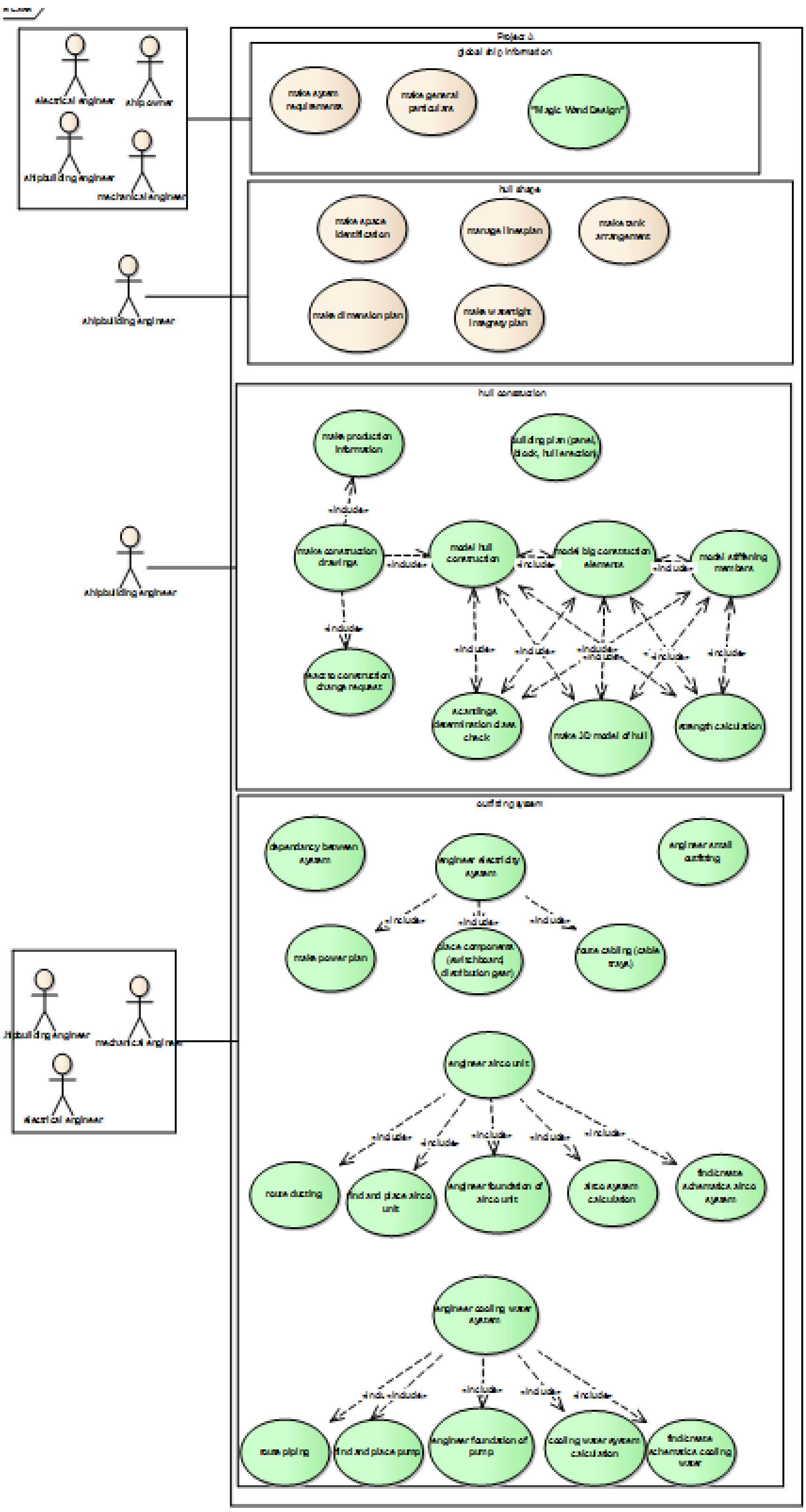



Figure 8, top level use case model

Appendix A provides a more clear picture of the top level use case model. From top to bottom the separate system will be described. The global ship information system describes the first design (mostly requirements) and provides the tools needed to describe the ship in a certain amount of detail, as specified by the ship owner. This description is both functional and has specific details about shape and components.

The hull shape system determines the hull shape and the internal division of the ship. An arrangement of every space inside the ship is made at the same time the hull shape is finalized.

When the shape of the ship is determined it has to be turned in to a construction. The purpose for this system is to make a construction model of the ship that represents a steel model and that depending on the construction application may or may not provide production information.

Finally, the spaces in the ship are filled with the systems that belong in the ship. This is done in the by making use of the tools provided by the outfitting system where the placed components are provided with piping, ducting, and/or cabling. Furthermore all the small outfitting/steelwork is also described in this system.

The complete process starts at the global ship information system. Once the process has started, all the systems are exercised parallel. On a more detailed level the activities of the different systems which are strongly coupled are sequential, e.g. the use case that takes care of a piece of hull construction is a predecessor for an outfitting use case that needs to place a component in that specific piece of hull.

## 5.1 Terminology

A lot of documents are produced when engineering a ship. The main terms used in the model when referring to those documents are **requirements** and **schematics**.

- **Requirements**: Consist of performance requirements that are ship specific, the component or system needs to satisfy those requirements and the actual performance of the component or system.
- **Schematics**: Consist of the calculations of system and component, specifications, characteristics, and the schematic drawings of the system and component.

The other important terms use in the activity diagrams are **collect**, **construct**, and **save**. These terms represent the activities that provide the link between the engineering system and the connecting PDM system.

- **Collect**: Consist of the searching the documents and information that are necessary for the task that the engineer needs to do.
- **Construct**: When the engineer is executing his task by engineering/modelling a system or piece of hull construction it will be indicated by a construct action. When the engineering model is connected to the PDM model the construct action makes use of the construct functionality of the PDM system.
- **Save**: When the engineer finishes his task and has created something it needs to be saved.

## 5.2 The actors

Four actors are involved in the use case diagram are not professions but job profiles, except for the ship owner which is an external influence but in the early design one that cannot be left out. The job profiles that are necessary to engineer a ship are the shipbuilding engineer, the mechanical engineer, and the electrical engineer. All the disciplines that are involved with engineering a ship are divided over these three profiles.

### 5.2.1 Shipbuilding engineer, mechanical engineer, and electrical engineer

The engineering tasks of a ship can be split into two main categories, the hull construction and outfitting thus the bottom two systems Figure 8. The shipbuilding engineers are responsible for the shape and arrangement of the ship, while the mechanical engineer carries the responsibility for the systems and their components. Those systems and their components need power, the electrical engineer works together with the mechanical engineer to design the corresponding electrical system within the ship. When those disciplines interact and conflicts occur, the disciplines need to work together and apply changes to either the construction, a system, or both. When the consequences of one of these changes result in a change in the construction, the shipbuilding engineer is responsible for the alterations in the construction; otherwise, when a system needs change, it is the responsibility of the mechanical engineer to apply the changes.

The engineering tasks of the different job profiles are not conducted completely separate. The engineering phase of a ship is a mutual effort of all the job profiles. The ship construction and its components are adapted to each other. The design compromises to resolve conflicts between ship construction and one or more components that are made may result in a change of ship construction, one or components, or both.

## 5.3 Important use cases and their containing activity diagrams

The point of the use case diagram is to have proper description of the engineering process that can be connected to a PDM system. The activity diagrams which are in the use cases consist of activities. The activities that have the above described **collect, construct,** and **save** actions trigger the PDM system to help the engineering system. The description of the following use cases and their containing activity diagrams are important to the connection between engineering process and PDM.

### 5.3.1 Model hull construction, model big construction elements, model stiffening members

When the global ship information and hull shape systems have produced enough information to start modelling the construction of the ship the model hull construction use case will start. Quickly after that the model big construction elements use case will start and after the model stiffening members use case. These three use cases form the basis of the hull construction system. Within these use cases are the actual modelling activities thus within these use cases the ship is transformed from a shape to an actual construction. Also within these use cases the PDM system is used. Figure 9, Figure 10, and Figure 11 show the three activity diagrams that are within the use cases. The use cases that are responsible for the modelling of the hull construction are broken down into these three subjects this does not mean that they are always executed in this order. They can be executed at the same time or all three can be executed at the same time by one person.

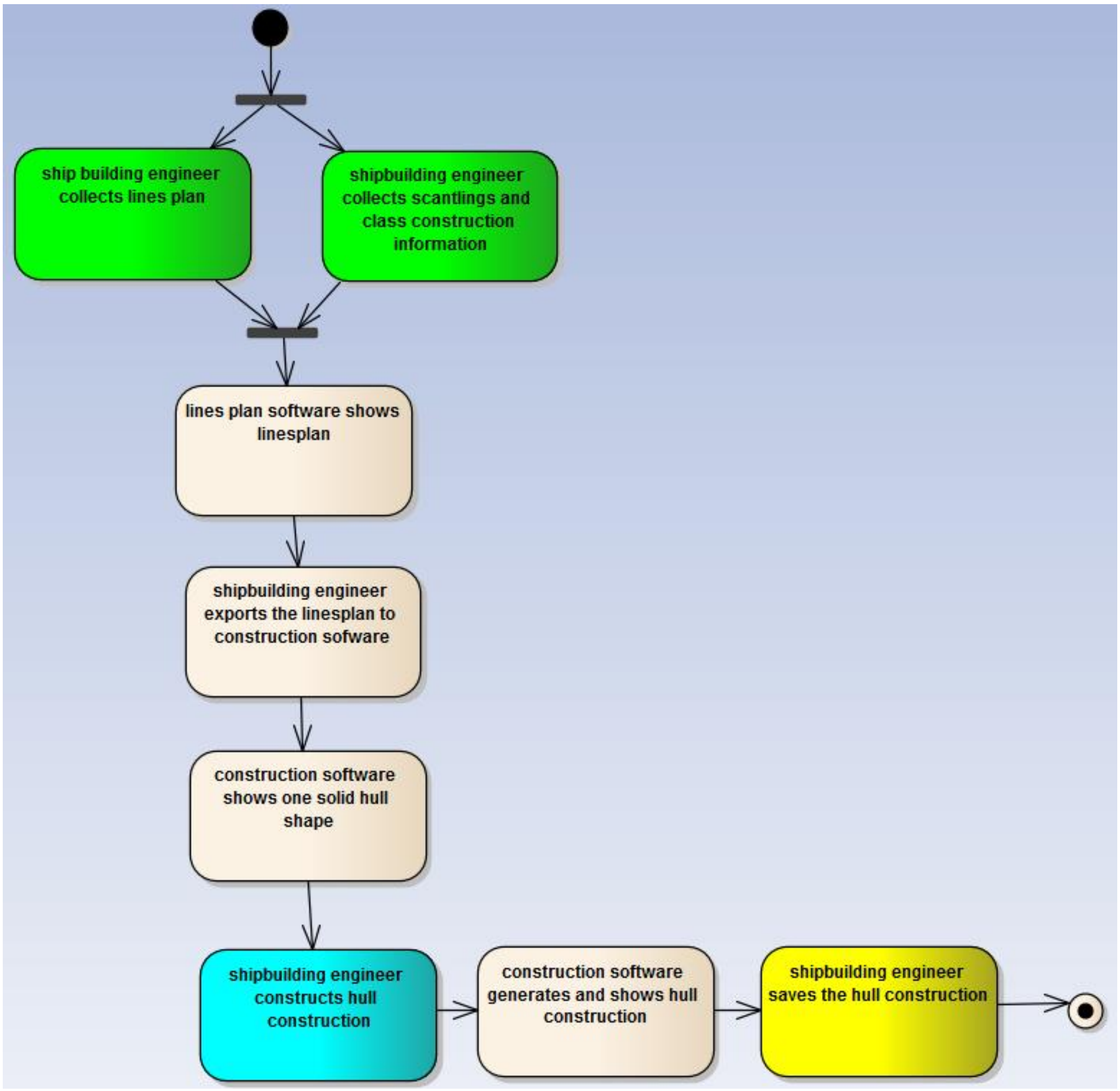



Figure 9, model hull construction

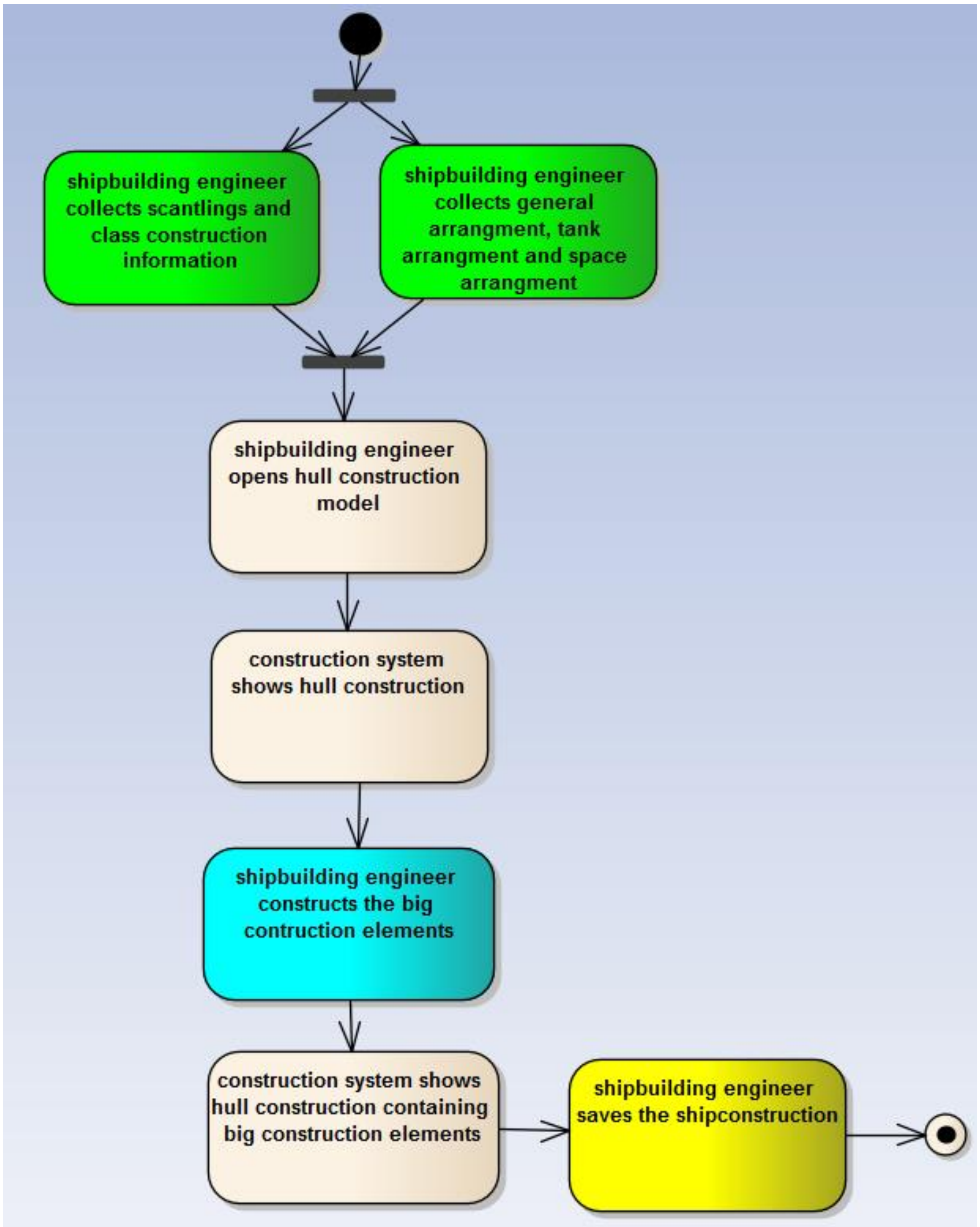



Figure 10, model big construction elements

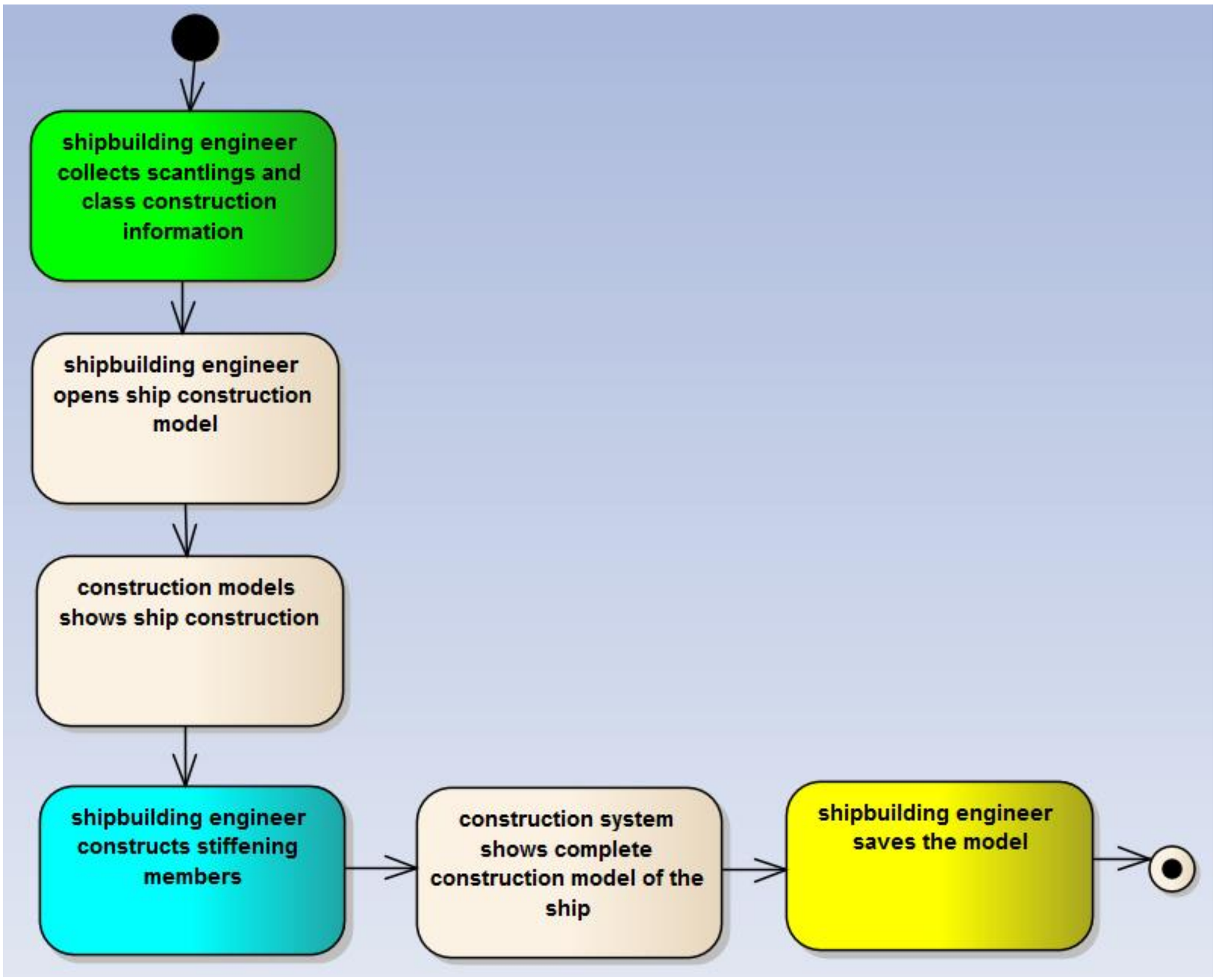



Figure 11, model stiffening members

The three activity diagrams have similar activities, these activities are coloured green, blue, and yellow. These activities are explained in Chapter 6.1.

- **Collect:** the green activities shown in the figures all contain collect. The collect actions is to gather the necessary information that the engineer needs to execute the complete diagram.
- **Construct:** the blue activities shown contain construct. The construct activity represents the actual modelling the engineer needs to do, basically it is the heart of the use case.
- **Save:** the yellow activities show when the engineer as contributed to a project by modelling for example a piece of the hull construction the it needs to be saved in some kind of project directory.

The engineering system is already equipped to be connected to a PDM system where these activities are the links between the engineering system and PDM system. In the situation where a PDM system is connected the **collect** activity will gather its information from the PDM system. The **construct** activity will use the PDM system to do the modelling. The **save** activity will trigger some different activities in the PDM system a detailed explanation of the PDM system and its use is done in Chapter 7. When a construction piece is finished and saved it is given a status by the PDM system to show other engineers that activities concerning surrounding construction elements can start and that the construction piece needs to be checked by someone who has the power to approve the construction. The assigning of statuses is done by the PDM system as well and explained in Chapter 7.1.3.

### 5.3.2 Make 3D model of hull

After the construction modelling of the hull construction there needs to be some gateway to the outfitting system. It is assumed that the outfitting is modelled in another application therefore the hull construction model needs to be imported in the outfitting application to create a working environment for the mechanical engineer that is going to be engineering the systems. The make 3D model of hull use case contains only the collect and save activity which are coloured green in Figure 12. When a construction piece is finished, it will be checked and the construction piece proceeds to the outfitting system. In the outfitting system the hull construction model is imported in the outfitting application. When the construction piece is imported and it is saved, a working environment for outfitting is created. Every time a piece of the hull construction is finished, this use case makes sure that it is transferred to the outfitting application.

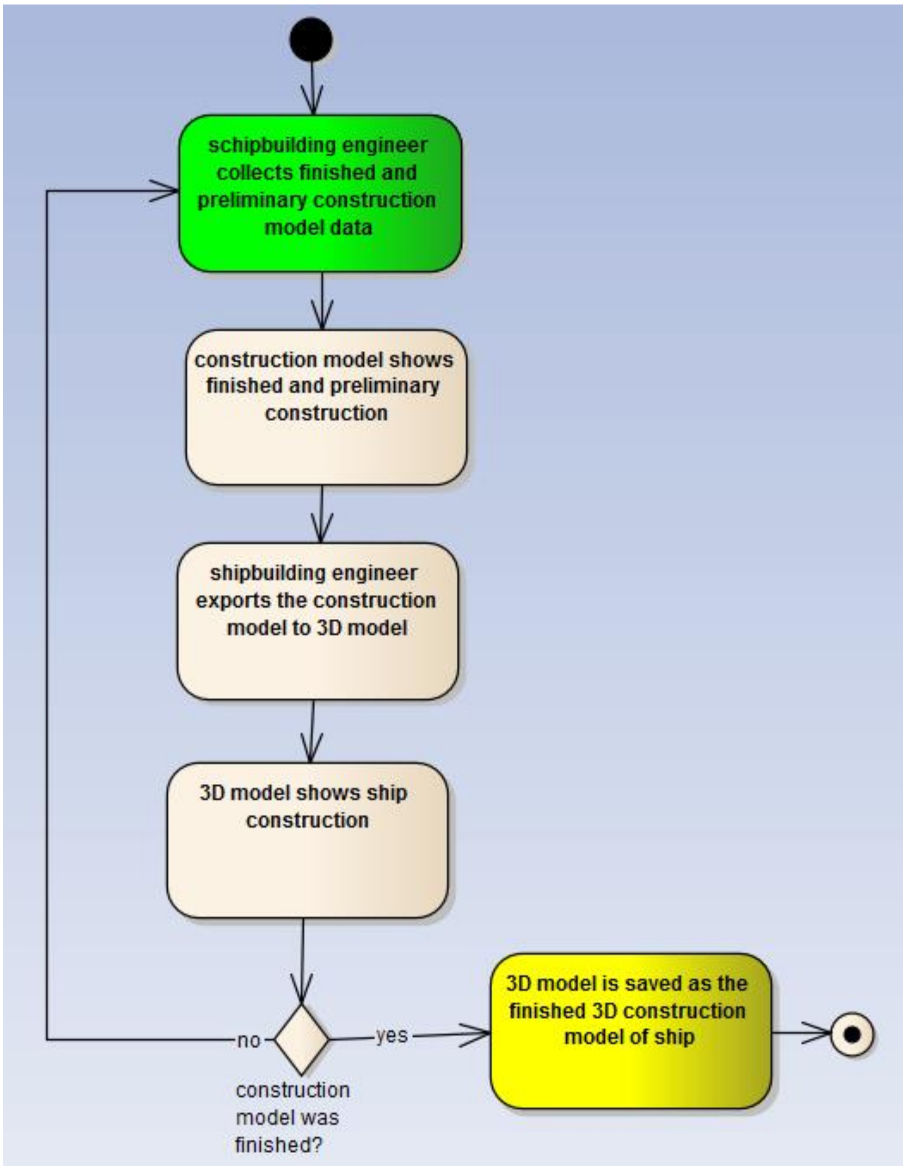



Figure 12, make 3D model of hull

### 5.3.3 Outfitting system

After a working environment is created the modelling activities of the outfitting system can start. Within the outfitting system exist use cases that that contain activities that produce information necessary for the modelling of the outfitting. This means that those use cases have already started because the modelling cannot start if a system has not been calculated correctly, even though a working environment is available. The outfitting system contains the engineering of three systems as can be seen in Figure 8. The engineer cooling water system will be explained in more detail to show the connection with PDM. A complete explanation of the use case model can be found in appendix A.

### 5.3.4 Engineer cooling water system

As shown in Figure 8, two of the use cases that include the engineer cooling water system use case, are cooling water system calculation and find/create schematics cooling water. These two use cases of which the activity diagrams are shown in Figure 13 and Figure 14, are very important because these two use cases specify how the system and its components will be specified. The calculations and schematic of the single components and even some complete systems are stored in the PDM system. The system and its components are divided in assemblies and sub-assemblies, which also contain the schematics concerning the components and systems.

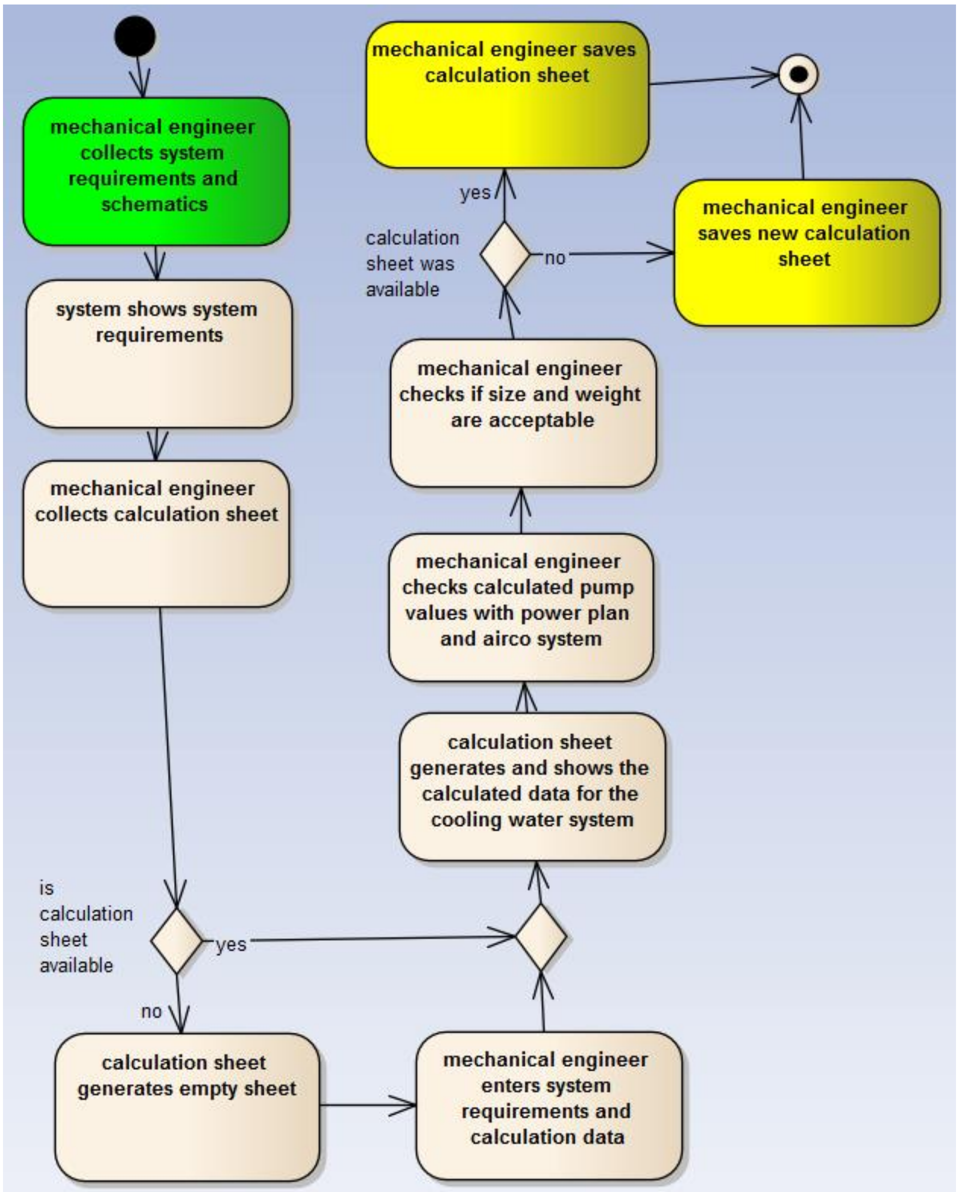



Figure 13, cooling water system calculation

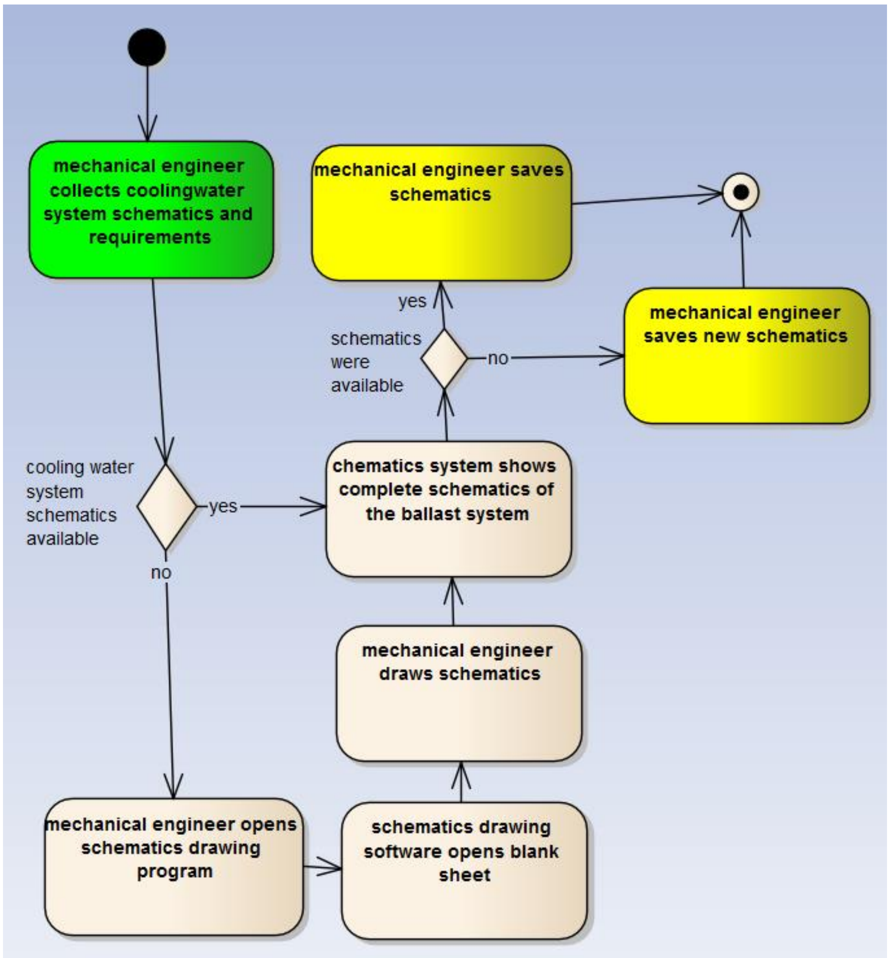



Figure 14, find/create schematics cooling water

Again, the shown activity diagrams contain green activities that describe the connection with the PDM system. These activity diagrams are kind of a PDM actions because the main objective of the diagrams is to find the schematics document of the cooling water system in the PDM system. If the system has been engineered before the schematics document which contains the calculations, of the single components will be attached to the assemblies that make out the complete system. When the system will be engineered for the first time the schematics document needs to be created. It is assumed that the calculations of a system are most important piece within the schematics document therefore it has a separate use case.

The following three activity diagrams; find and place pump, engineer foundation of the pump, route piping are shown in Figure 15, Figure 16, and Figure 17. They represent the actual modelling of the cooling water system and its components. The three different diagrams give the impression that the three use cases will always be separately executed but it is possible that they are combined. These activity diagrams are explained in a future situation when PDM is already connected. Without the PDM system connected the activity diagrams are executed straight forward as the activities state.

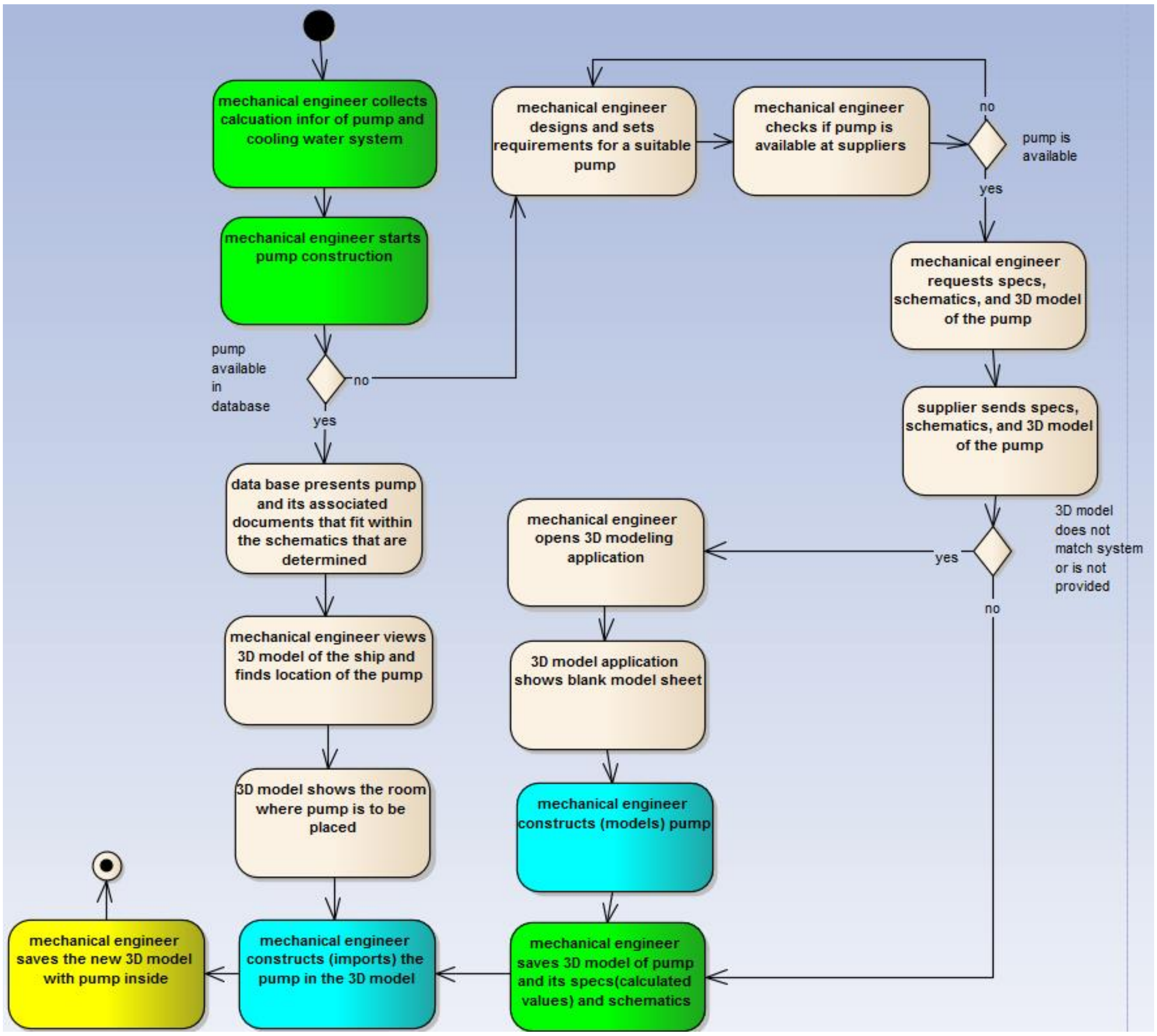



Figure 15, find and place pump

Even though the cooling water system calculation may have already determined the pump that needs to be in this system the pump needs to be found in the PDM system which is described by the green action "mechanical engineer starts pump construction". It may be possible that the pump is also described in an assembly which also contains the foundation. In that case, the use case "engineer foundation of pump" of which the activity diagram is depicted in Figure 16, does not have to be executed. When the to be used pump is not in the PDM system, should be ordered from a sub-contractor this workflow is shown in the right downward flow of actions in Figure 15. When a new object like a pump enters the system, it needs to be saved. When one is used, it does not need to be saved because it is already in the PDM system. Therefore, when new assemblies or parts enter the system they need to be saved. A new assembly is also created when an existing assembly is changed into a new assembly.

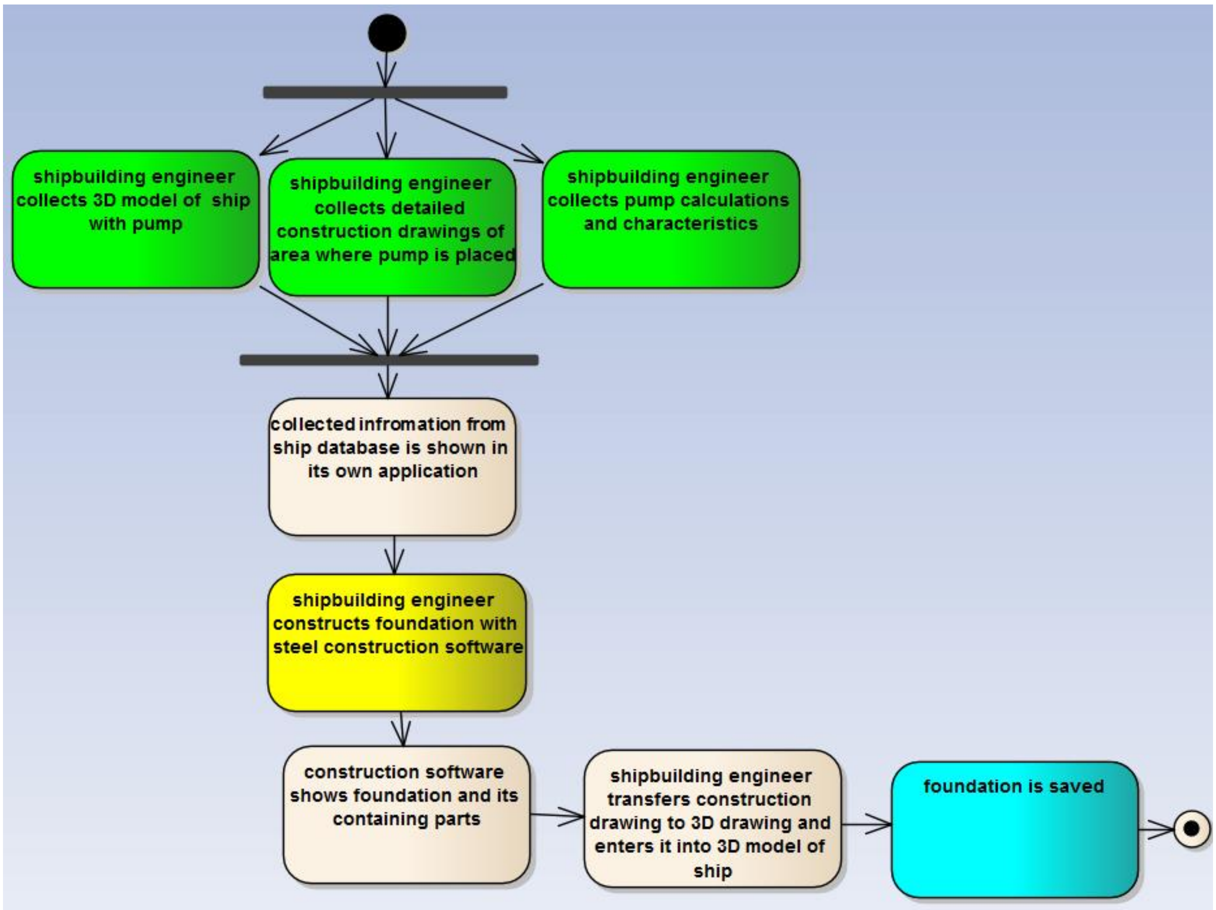



Figure 16, engineer foundation of the pump

When the foundation is already in the assembly of the pump, this activity diagram would not be executed. Although this activity diagram belongs in a use case that is located in the outfitting system, the application used for these actions is the hull construction application. This application is used because a foundation is constructed of steel and the construction possibly needs production information. Just as the activity diagrams explained before, this activity diagram uses the PDM system with the collect, construct and save actions. Even though the foundation of the pump is modelled in the hull construction application, it is part of the cooling water system assembly. This means that both the applications need to be able to at least show each other's models.

The route piping activity diagram as shown in Figure 17, also has the collect and save actions, instead of the construct there is the route piping action. This action allows for the same interactions with the PDM system as the "construct" activity but the "route" term fits better with the task the engineer executes. The exact practice of how the pipes are routed is not determined because it is highly dependent on the application used for routing pipes. The PDM system is capable of handling different ways of pipe routing because the structure of the stored assemblies is not defined, it is left up to the user

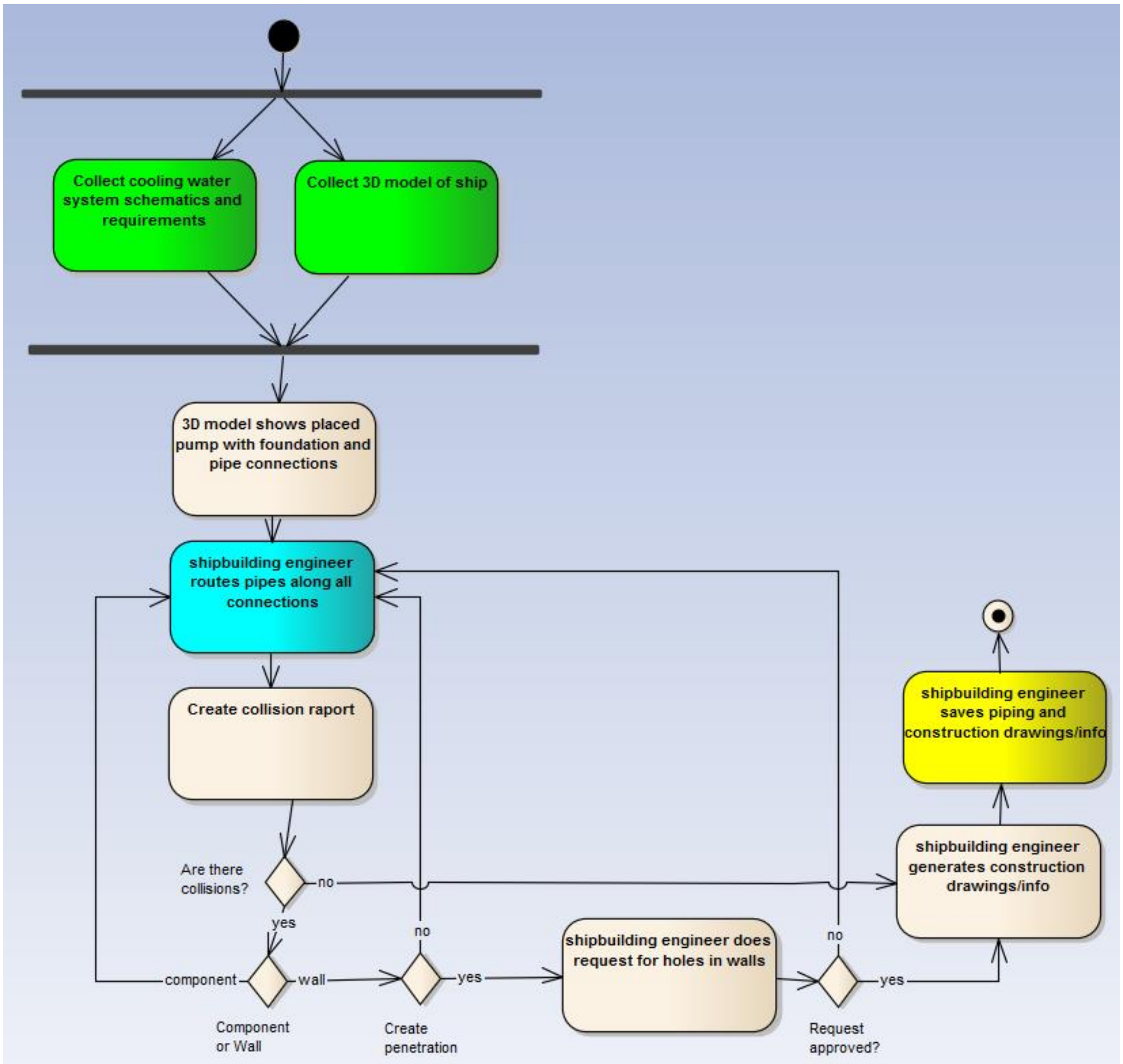



Figure 17, route piping

### 5.3.5 Engineering small outfitting

Next to all the parts and assemblies that are created when systems enter the hull construction, there are a lot of small steelwork parts that are necessary in the ship. As the title of the activity diagram suggests some of the outfitting parts are very small and don't require production information. Some do need production information and thus both the outfitting and the hull application will handle small outfitting parts and assemblies. This split can also be seen in Figure 18. It was already explained that the hull and outfitting applications need to be able to show each other models. When small outfitting has assemblies/parts that are modelled in both applications it may become difficult to keep a clear overview of what is engineered. A third application that combines the models of the two applications will be used to depict the completely engineered ship, also to check if the engineering is correct. In the Chapter: system architecture, the use of the applications in connection with the engineering and PDM systems will be explained in more detail. The connection with the PDM system is the same as described for the other activity diagrams. A difference is that the use of existing parts and assemblies for the small outfitting can be done to a much greater extent than with the engineering of systems. The parts and assemblies do not need to be very ship specific.

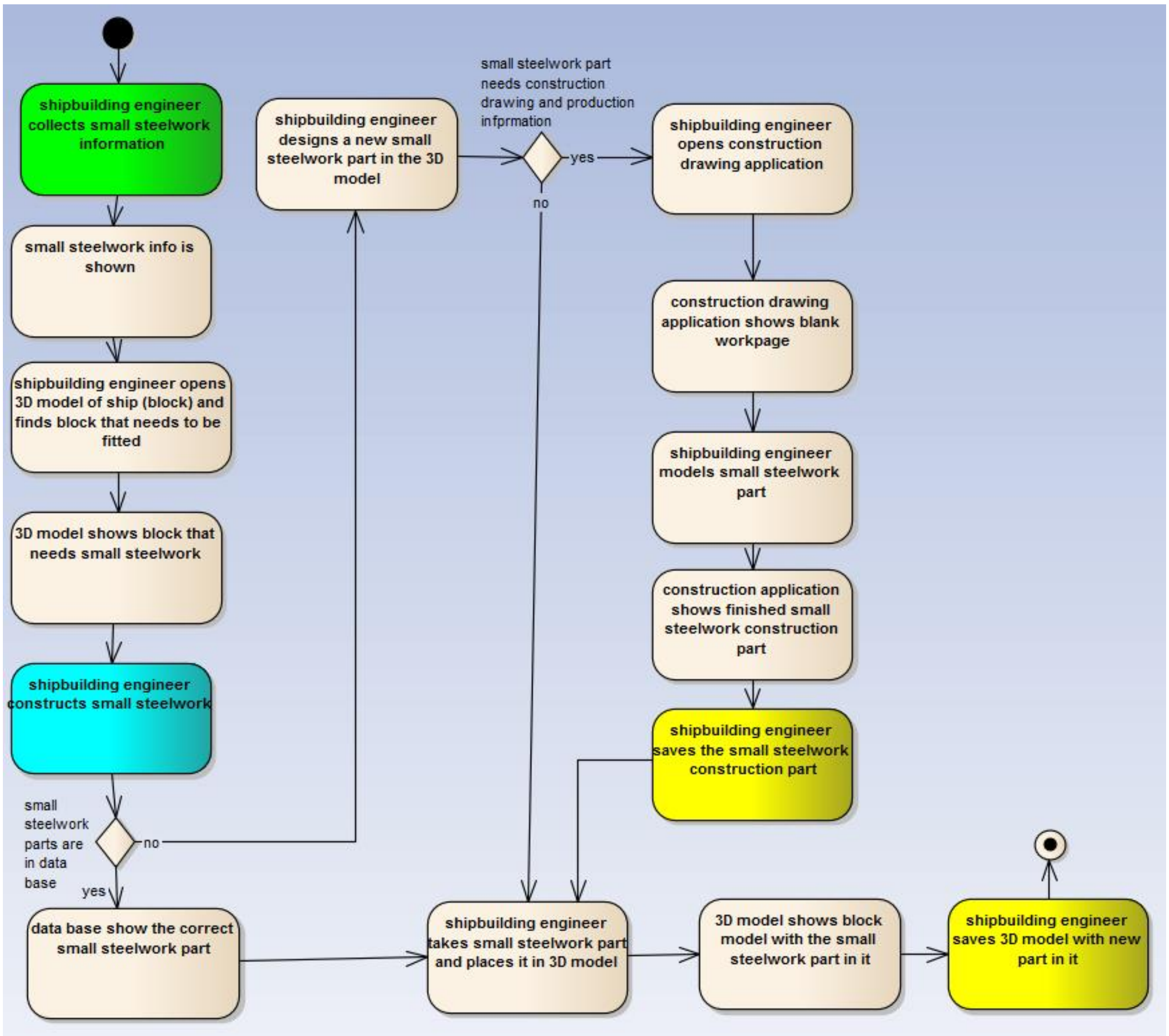



Figure 18, engineer small outfitting

# 6 PDM functions

For the engineering system to be executed together with PDM, the functions concerning PDM should be added to the engineering process, these functions are contained in an interface. This interface can be best explained as the control centre of the PDM system. It functions as an environment for the PDM functions. Through the interface the stored data is connected with the engineering system. The stored data is located in the data vault. Figure 19 shows the use case model of the PDM system and its containing functions. The interface contains all the executive and managerial functions, the data vault contains the information storage functions. The PDM solution is equipped to provide the use of assemblies when modelling the hull construction and outfitting components. An assembly is a piece of the ship that is made up of single parts. The size and shape of the assembly is not fixed there can be big and small assemblies. A more detailed description of assemblies follows in chapter 7.2.1. The model of the PDM solution is made up of two components the interface en the Data vaulting component which consists of two vaults; PDM vault and CAD vault. The functions that are executed by these components are displayed by the use cases. How these functions are executed is described by the activity diagrams that are contained within the use cases. These use cases are not connected to actors because it are the components ( interface and the data vaults) itself that execute the functions.

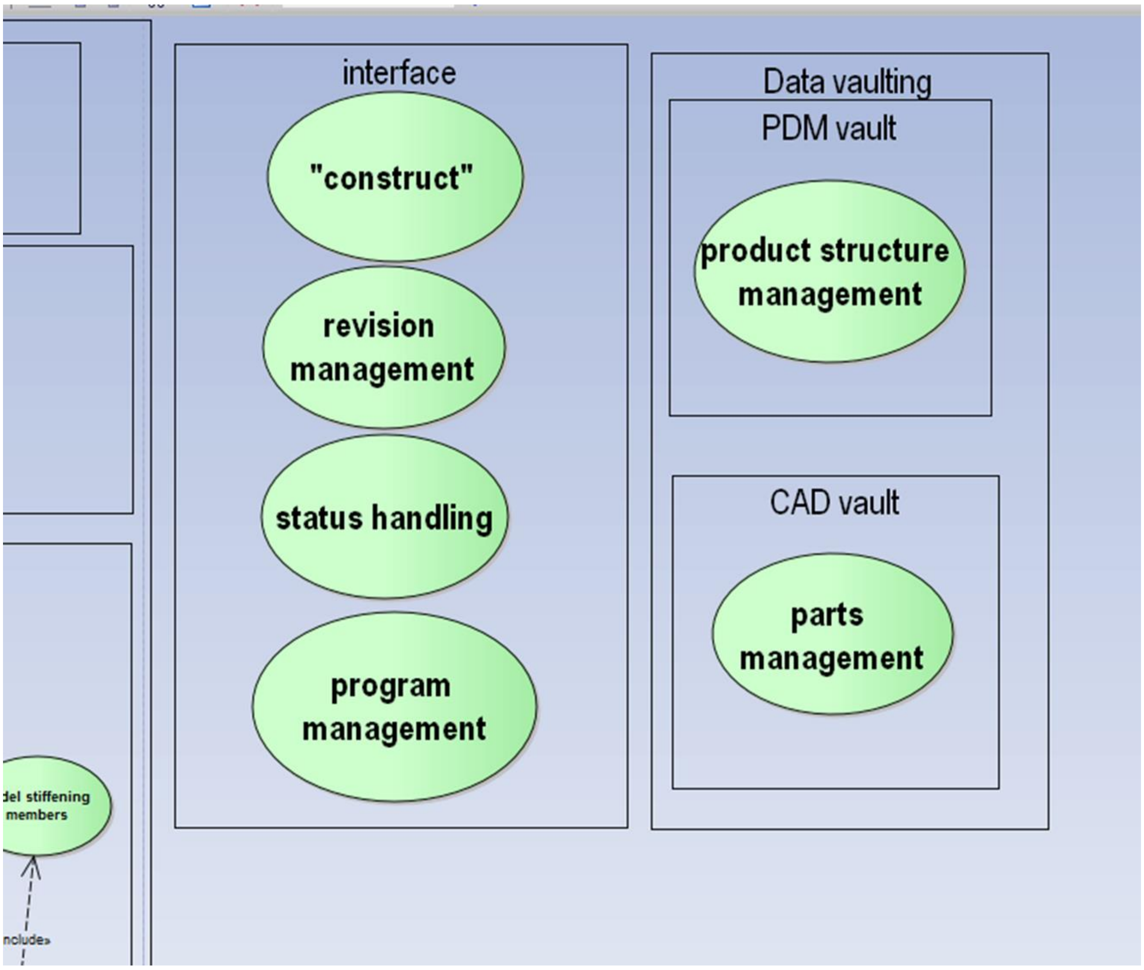



Figure 19, PDM system

## 6.1 PDM use, scenarios

After the investigation of the in Chapter 5 described engineering process in combination with the potential benefits of using PDM, PDM-related workflows have been envisaged. When using the engineering system and also making use of the PDM solution there are six scenarios that describe the possibilities that can occur when constructing the ships construction and systems. The six following described scenarios give and explanation of the workflows that an engineer can encounter while constructing the hull, component, or system of the ship. It is also possible that the engineer will encounter a choice between two workflows.

1. **Construct new assembly from scratch:** The scenario “construct from scratch” has initially no relation with the PDM system because it means that an assembly is constructed completely new. All the parts that make up an assembly are newly modelled. When an assembly is modelled new like this the parts arise in the local CAD database therefore the PDM system has nothing to do with the majority of this scenario. The connection with the PDM system comes in play when the newly formed assembly is finished and is entered in the PDM system for the reason that it than can be used by other projects.
2. **Construct new assembly from existing parts:** Just as scenario one, this scenario will create a new assembly. The parts that will make up the assembly are existing parts that are stored in the intelligent CAD vault. Because the parts are in the intelligent CAD vault the PDM interface needs to be used to assemble the parts together in order to form a new assembly.
3. **Use an existing assembly:** When using existing assemblies without changing them the PDM system directly re-uses engineering data which will results in the most time reduction within the engineering phase. The scenario consist of the use of complete assemblies that are in the PDM vault. When assemblies from the PDM vault are used almost all the information regarding the assembly is available thus resulting in time reduction. When this scenario is eligible there are two sub-scenarios. The assembly is in the PDM vault or the assembly is already in use by another project which means that the assembly is in the interface.
4. **Change of existing assembly with new modelled parts:** An existing assembly sometimes does not completely fit the concerning design. When an assembly needs to be changed it first needs to be used according to scenario three. After scenario three is finished the assembly needs to be copied en from that moment it will become a new assembly. Within the new assembly changes can be made, a part can be removed and replaced by a new modelled part or an existing part can be changed. When that is the case the separate part also need to be copied to form a new part, after the copying it becomes a new part.
5. **Change existing assembly with existing parts:** This scenario is very much like scenario four only no separate parts are changed or newly modelled. Just as scenario four, the assembly first needs to be copied to form a new assembly. When the new assembly is created parts can be changed for other existing parts. Also parts can be added to the assembly.
6. **Magic wand design:** In order to speed up the engineering process more than the PDM itself is already doing, the modelling of the ship is pulled forwards to the early design. Basically this scenario works the same as the use existing assembly scenario. The difference is that only the 3D model, that is also part of the assembly information, is used. This makes it possible to do quick “lego fashioned” engineering very early in the design cycle by modelling parts of the ship using 3D models of the assemblies that do not require the intelligent information from the CAD vault.

From the described five scenarios it becomes clear that the most reduction in engineering time is obtained when scenario three is executed. The reason why there are other scenarios is that existing assemblies are not always compatible with the design of a ship and need to be changed or a new one need to be modelled. Also an existing assembly is not always the best engineering solution for a system or construction detail. It can happen that an existing assembly fits the ships design but it is not an ideal engineering solution. The choice which scenario will be executed is done by the user of the PDM system. When a scenario is chosen that changes an assembly the revision management determines which scenario that changes assemblies needs to be executed.

When assemblies are re-used by different projects conflicts can occur when different versions are made of the assemblies. A common situation occurring when re-using and changing assemblies is described in Table 3. In Table 3, it is described how two users are intending to use and change the content of assembly J. Ending up with the situation where the original content J and the content produced by user A, $J^*$ is lost because user B saves the content at the last transaction time stamp $T_4$.

| Transaction Time $T_i$ | User A | User B |
|---|---|---|
| $T_1$ | Copy assembly J into A's workspace | - |
| $T_2$ | Update the content of Assembly J to $J^*$ | Copy assembly J into B's workspace |
| $T_3$ | Put assembly J with content $J^*$ back into the vault | Update content assembly J to $J^{**}$ |
| $T_4$ | - | Put assembly J with content $J^{**}$ back into the vault |

Table 3, time stamping in the use of PDM

The situation that is described in Table 3 causes the loss of data when the data is re-used. Involuntarily loss of data may never be the case. If this situation is places within the architecture the assemblies (files) are never copied to a workspace, a connection is made between the application and the data within the vault. This will at least prevent that the data is present at different locations and the data cannot be changed separate from each other, preventing these concurrent engineering conflicts. When the choice is made to change an assembly because it is necessary for the project, one of the scenarios described in this chapter will be executed and a new revision or completely new assembly will arise.

## 6.2 Interface

The interface manages the connections with and actions done to the data in the two vaults. An engineer works on a project and executes actions that are contained in one of the use cases that make up the engineering system. The interface manages which actions the engineer needs to execute and makes sure that the engineer has the correct data also the interface manages an assessment for the finished use cases. This section will describe in detail the functions within the interface shown in Figure 19.

### 6.2.1 Construct

The construct use case will be executed when the engineer executes an construct activity that is in the engineering system. The activity diagrams that are contained in the use cases of the engineering system have a construct action. This construct action is directly linked with the construct scenarios that are in the construct use case. These scenarios are shown in Figure 20. From left to right the construction scenarios will be explained. The construct function and its containing scenarios provides the connections to the data that is stored in the vault. Basically the execution of a construct scenario means that the engineer is modelling a piece of a ship, either by using previous modelled assemblies or the engineer models the piece new. The actions that make up the scenarios in the construct use case describe the functionality of the interface and not the modelling tasks of an engineer.

The scenarios are explained in Chapter 7.3 but the basic features of how the interface makes sure the scenarios can be executed will be explained. The construct function within the interface makes sure that the applications in the projects are able to use the correct assembly information. This means when an existing assembly is used thus scenarios 3: use existing assembly,4: change existing assembly with new modelled parts, and 5: change existing assembly with existing parts. The interface makes sure that the project receives the assembly information. The interface uses the location of the intelligent data of the assembly which is stored together with the assembly to make sure that the applications in the projects can use the intelligent data.

Scenarios 1 and 2 are used when there is no existing assembly for a project available. When the new assembly is modelled from existing parts the interface needs to create an empty assembly and make the connection between project and intelligent data part by part up until the assembly is finished. The interface does not only create an assembly by managing the information about where the intelligent data is stored in the CAD vault but an assembly is a complete package containing all the information necessary. Is also the part of an assembly are created new as scenario 1 the intelligent data is create in the local workspace of the location. The locally created parts of a new assembly will be relocated to the CAD vault and the assembly located in the PDM vault. As described in Figure 20 the interface establishes connections between project and the right data, also manages that the data and the data packages (assemblies) are in the correct location.

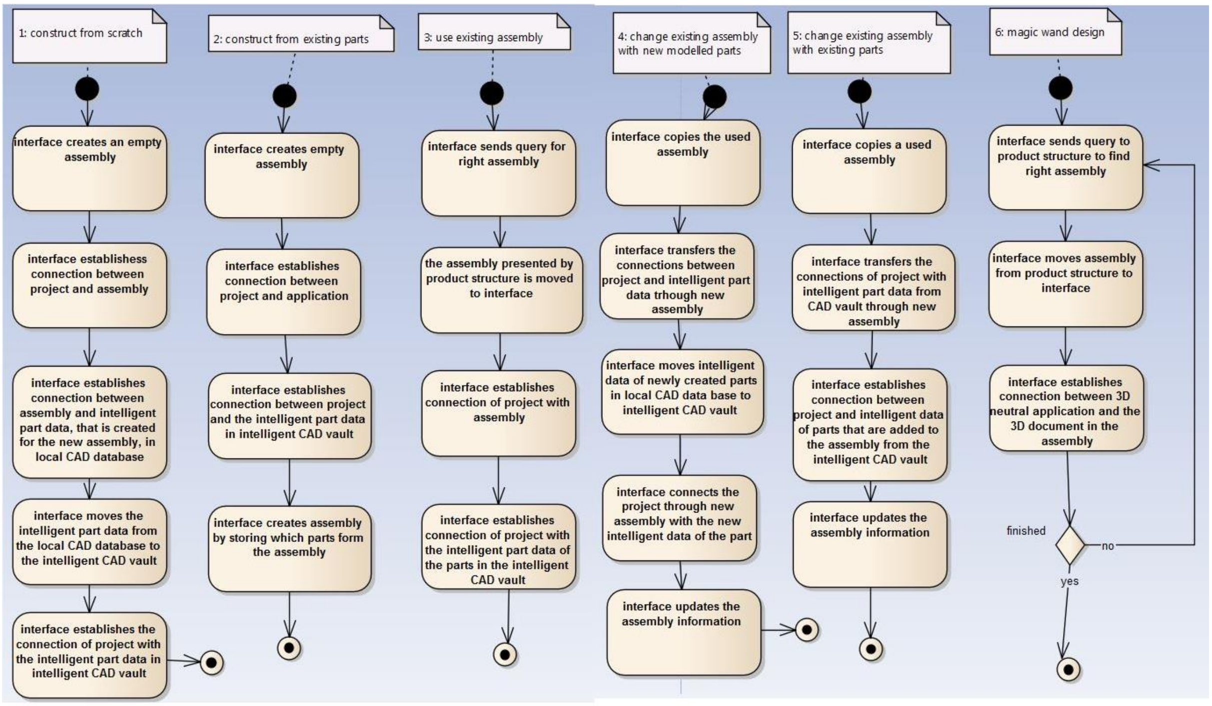



Figure 20, Activity diagram that describes the PDM use scenarios

### 6.2.2 Revision management

As described in the previous section different scenarios can be executed when an engineer executes the construction task to model a certain piece of the ship. Depending on which scenario from Figure 20 is executed the decision whether an assembly will be a completely new assembly or a new version of that assembly is made by revision management of which the activity diagram is depicted in Figure 21. The scenarios 3 and 6 do not trigger the revision management, only the scenarios that create or change an assembly are assessed by revision management. The first diamond distinguishes between the fact if an assembly that a project needs is already in use by another project. If the assembly is already in use by another project it automatically means that that the assembly will be changed according to either scenario 4 or 5. After the change of the assembly the choice needs to be made if the assembly become a completely new assembly or a new revision of the assembly:

- **New revision:** When a changed assembly becomes a new revision of the old assembly it means that they have a lot of similarities. Thus when the to be changed assembly has name A, the changes assembly will be A1.
- **New assembly:** This is the case when the changes to an assembly are big. Thus when the to be changed assembly has name A, the changes assembly will B.

When an existing but unused assembly will be needed for the project the first diamond will give a "no" . As described before, this use case will only be executed when an existing assembly is changed. This also means that this use case maintains some kind of shelf life for the assemblies. When an existing assembly is considered too old it needs to be revised to an up to date assembly, and the old assembly is not kept. Getting to the second diamond "old assembly is kept" when it is kept the same workflow as above described is executed. When the old assembly is certainly not kept thus an assembly is being updated the construct scenarios 1 or 2 ned to be executed. The only difference is that no new assembly is created because the old existing one remains but it has been revised.

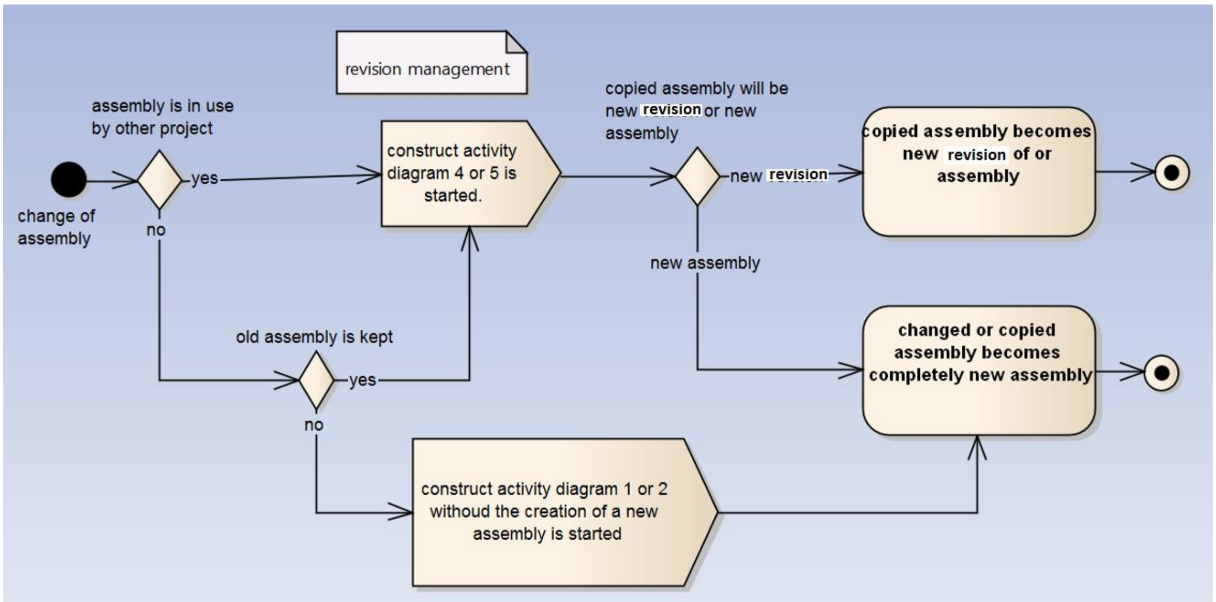



Figure 21, revision management

### 6.2.3 Status handling

The engineering system is build-up of use cases. These use cases are work packages and all the use cases combined form a complete project. To finish a project all the use cases need to be executed and finished. The use cases that build up a project are not sequential but it is unavoidable that a lot use cases are predecessors to other use cases. When a certain building strategy is applied to the process the use cases need to be executed in some kind of order. To somehow make visible what the current status of progress is or if some predecessors of use cases have been finished. The use cases are awarded a status. The statuses that can be awarded to the use cases are:

- **Un-started:** At the beginning of a project all the use cases have an un-started status.
- **In progress:** When an engineer starts a use case it receives the in progress status.
- **Engineer finished:** Because the hull construction and outfitting system are separate systems the use cases receive different statuses. When an engineer is working on a use case it should have the in progress status. When the engineer finishes it first receives a normal finished status. This status precedes the specific hull of outfitting finished status. When the executed tasks have been checked and approved the specific statuses are awarded.
    - Hull finished
    - Outfitting finished
- **Ready for production:** When a use case is completely finished and it concerns some part of the ship that needs to be manufactured or needs some kind of work it receives the status ready for production
- **Disapproved:** For a use case to be awarded the hull finished, outfitting finished, or ready for production status it may happen that the executed tasks are not approved. This means that the use case receives the disapproved status and revision needs to be done to the use case.

Figure 22 shows the activity diagrams that described the workflow of the status handling where the status changes are explained. The status handling use case is triggered when an use case from the engineering is started. The first activity refers to the program management which will be explained in Chapter 6.2.4, this program management assigns the use cases to the engineers according to their status. Program management will also handle how most of the hull construction use cases are predecessors for the outfitting use cases.

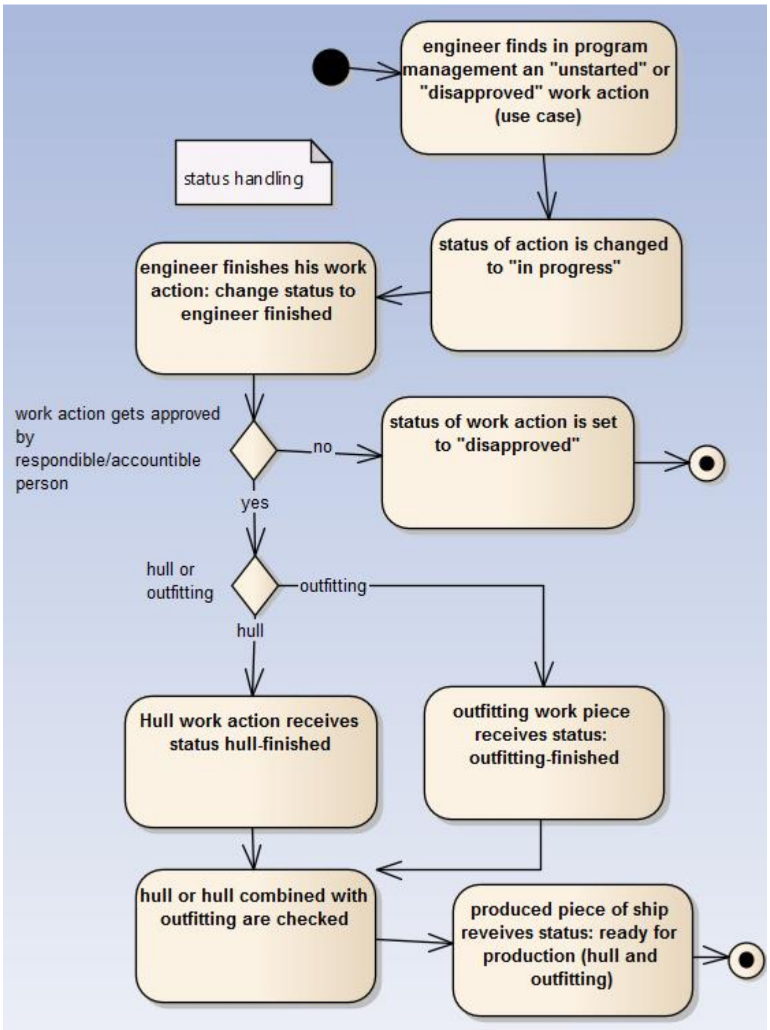



Figure 22, status handling

### 6.2.4 Program management

As shortly described above, the program management function is there to assign the use cases to engineers or make visible which use cases can be started by the engineers. Together with a work breakdown structure of a project which in the case of this research would be a use case breakdown structure, a priority list is made of the use cases that need to be executed. Evidently the top priority use cases will be assigned first. Besides the use cases that precede each other and the basic order of use cases the priority of an use case would in a real situation be very much dependant on the specific situation at that shipyard and in that project. Figure 23 shows the activity diagram of the program management function. The first two activities in the activity diagram describe the collection of the status information but also planning and resource information referred above as the yard and project specific influences for the priorities that may be given to the use cases.

Another specific situation is whether engineers at a yard work on one or more projects at the same time. If all engineers or some engineers work on multiple projects at the same time it is necessary for the program management function to manage a priority list of the use cases that are contained in all the projects. This priority list is not a static list because it is meant that the priority is also based on availability of the use case. Which means that they are available for an engineer to be executed.

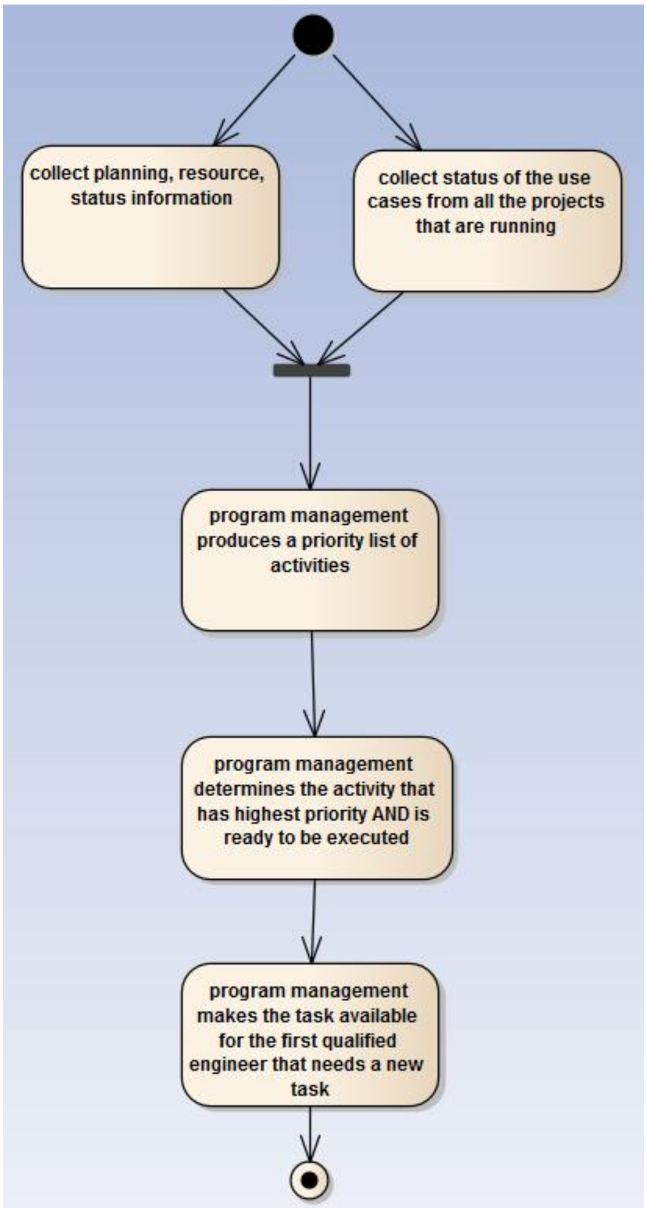



Figure 23, program management

## 6.3 Data vaulting

The storage of data is done in two vaults. The PDM vault and the CAD vault. The CAD vault contains the intelligent data of single parts/objects used by the CAD software. The PDM vault contains static information about the parts that are stored in the CAD vault. This static data is stored in assembly packages. An assembly is a piece of a ship that is made up of multiple parts/objects. The engineering system is described by the hull construction and outfitting system. These both have their own application. This means that an assembly is able to contain parts/objects of two applications.

### 6.3.1 PDM vault

The PDM vault stores information packages that are called assemblies. These packages are called assemblies because the main information that is contained in that package is the information that describes the assembly. The data in the PDM vault is static data, it contains no embedded information that is useful for the functionality of an application. This means that the intelligent CAD data, the data that contains the embedded information which makes the functions of the CAD applications possible, is not in this vault. The information that is contained in an assembly:

- **The containing parts/objects:** The assembly is a composition of parts/objects, all those parts/objects have their own size and shape. Within the assembly package is described how these parts/objects are oriented relative to each other and what the interrelations are between the parts/objects, thus describing the shape and size of the complete assembly.
- **Location of the parts/objects:** The assembly is a combination of parts/objects, but the PDM vault does not contain the intelligent data of those parts/objects. This intelligent data is stored in the CAD vault and the assembly package has information about the location of the intelligent data. How the intelligent data is located will be discussed in Chapter 7.2.2.
- **Production supporting information:** The assembly needs to be manufactured and production information is needed for this process. The production information is generated by the application and the assembly may receive some ship dependant features, therefore the production information is always generated by the application. But for every assembly there will be assembly specific production information which is off course stored together with the assembly.
- **BOM:** An assembly is made up of parts/objects some need to be bought or are in storage others are built on the yard. All these parts/objects appear on the Bill Of Materials that is stored in the assembly package.
- **Meta data:** Besides the assemblies there is also data that describes the characteristics of the static data of the assemblies. This data is used make the assembly recognisable in the vault so it can be found.
- **3D neutral format data:** Data in a neutral format that is of smaller digital size because it does not contain the specific intelligent data that is necessary for certain applications. This means that the 3D models in this neutral format data can be used by multiple applications. For the use of “magic wand design” as described in Chapter 8.7 the assembly needs to contain the 3D neutral format model of the assembly.

The product structure management activity diagram as shown in Figure 24 described the interaction between the interface and the PDM vault. From top to bottom the different workflows of the interaction between interface and PDM vault are described. The assembly does not leave that vault during this interaction. The information is only made available. The workflows are:

1. Every time when a change is made to and assembly or when a new assembly is added a check will be done to make sure there are no double or very much equal assemblies in the vault.
2. When a project needs an assembly the interface does a query to find the correct assembly. The information of the assembly is made available and the assembly is locked. When an assembly is locket it means that is in use by a project. Other projects can still use that assembly as well.
3. When a project is finished or the project just doesn’t need a certain assembly anymore, it needs to unlocked or the connection with the project need to be broken.
4. When a project wants to change an assembly the assembly needs to be copied and work with the copied assembly.

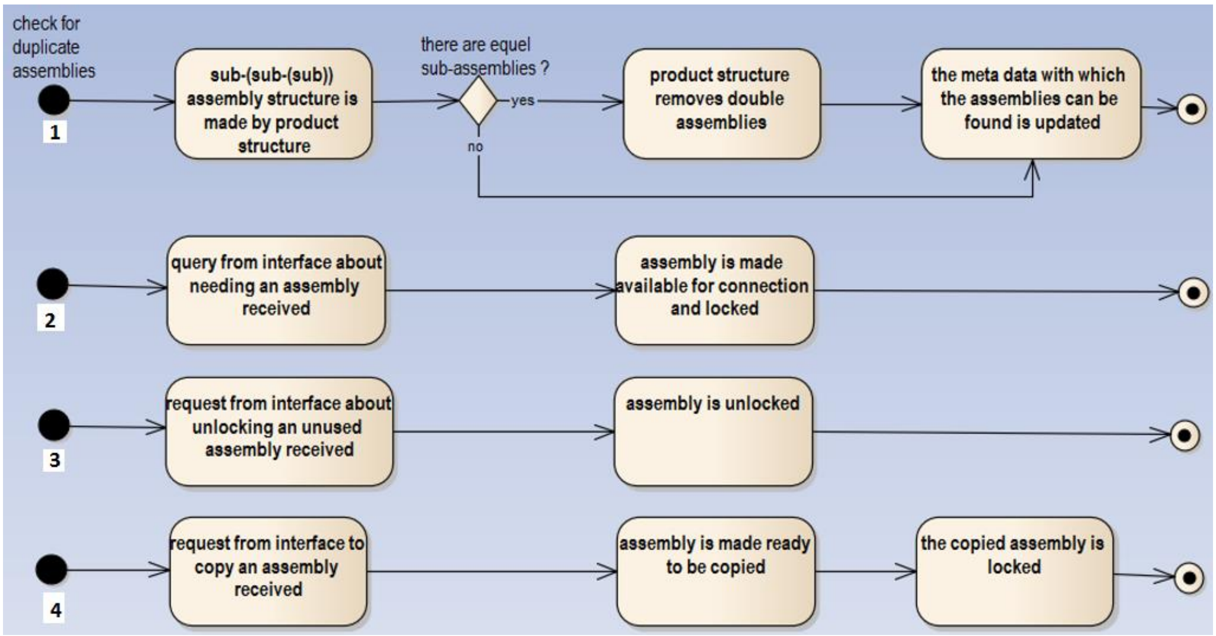



Figure 24, product structure management

### 6.3.2 CAD vault

All the intelligent CAD data is stored in the CAD vault which functions as one central location for the intelligent data of all the parts/objects that may be used by a CAD application. How the CAD vault functions is described by parts management. Within this use case an activity diagram describes how the vault handles the intelligent information. Figure 25 depicts the activity diagram of parts management.

Different activity paths can be executed by the parts management, depending on the task that the engineer is executing. Just as the product structure management the workflows of the parts management describe the interaction between CAD vault and interface. Furthermore the intelligent data does not leave the CAD vault, it is used by the applications on this central data storage location. Starting from the top the four activity paths are described.

1. When a project uses an assembly, the information about the location of the intelligent data is made available and the interface does a query to the CAD vault. The CAD vault makes the intelligent data available for the application of the project to connect to.
2. When a project is not using an assembly any more the connection with the assembly is broken and the interface will thus break the connection with the intelligent data too.
3. If an assembly is changed and there is a newly modelled part/object added, that object needs to be transferred to the CAD vault. New parts/objects are modelled in the local workspace of an application.
4. When an assembly is changed not by adding new parts but changing an existing part/object of the assembly it needs to be copied to the local workspace. After it is copied and changes workflow 3 needs to be executed.

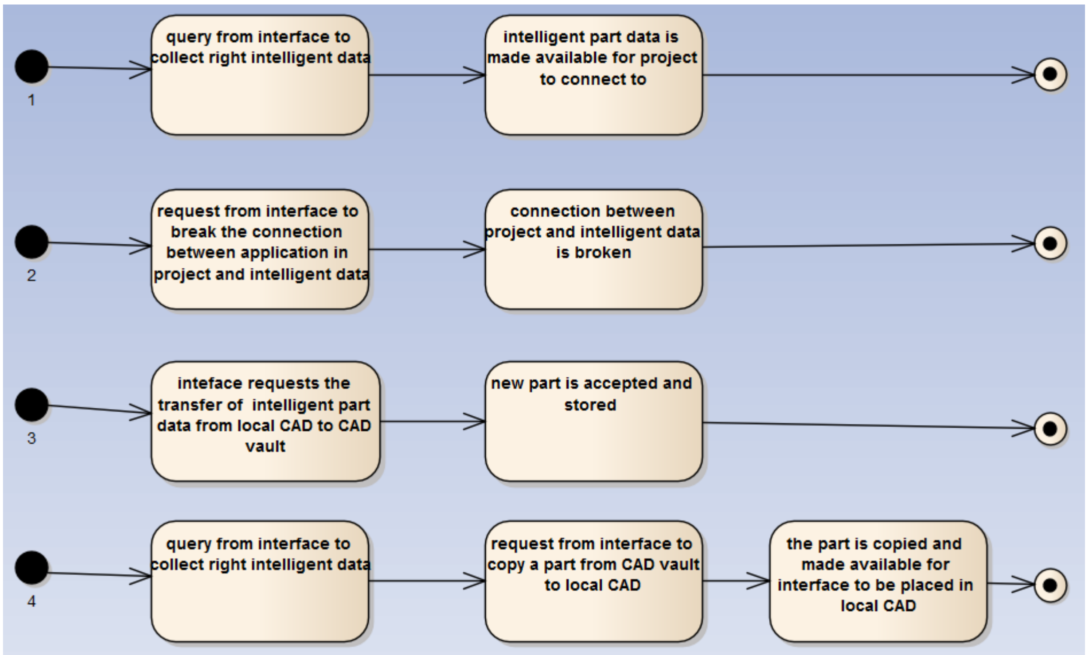



Figure 25, parts management activity diagram

# 7 Architecture

The conceived PDM interface solution as described in the use case model also needs to take shape in the form of a system. An architecture that represents the components of the system that facilitate the PDM interface is described in this chapter. Moreover, this chapter explains the use of the system in the form of 6 scenarios that were introduced in Chapter 7.1.

## 7.1 architecture components

The use case diagrams of the PDM solution as described in Chapter 7 already has the same shape as the system architecture explained in this chapter. The architecture depicts the PDM solution using real system components to get a grasp of how the PDM solution may be deployed.

### 7.1.1 Assemblies and parts

The first elements of the architecture that are explained are the assemblies and the parts. When the ship is modelled it is build up from assemblies that are build up from sub-assemblies that are build up from sub-sub-assemblies etc. Eventually the sub-sub-...-assemblies are build-up of parts.

Assembly and part elements are depicted with a small document sign and are shown mostly on the right side in Figure 29 to Figure 34. Within these representations the document symbols with one capital letter represent a part and the document symbols with a combination of capital letters represent an assembly. Thus within these figures there are only assemblies that consist of parts, i.e. bottom level assemblies.

The goal of forming assemblies is to shorten engineering time by using pre-engineered pieces of a ship. Depicted by Figure 26, Figure 27, and Figure 28 is an example of how an assembly can be engineered. The assembly could represent a cooling water system consisting of three components that are connected by a series of pipes and valves. The three components that are in this assembly are highlighted in the figures and also represent one part. The valves and pipe connections are parts. Depending on how the assembly is modelled the pipe parts are arranged in this assembly, pipe sections can be one part or more. Unfortunately the foundation of the components is missing but the founding connections to the hull construction are shown in the figures. These foundation parts are also parts of the assembly.

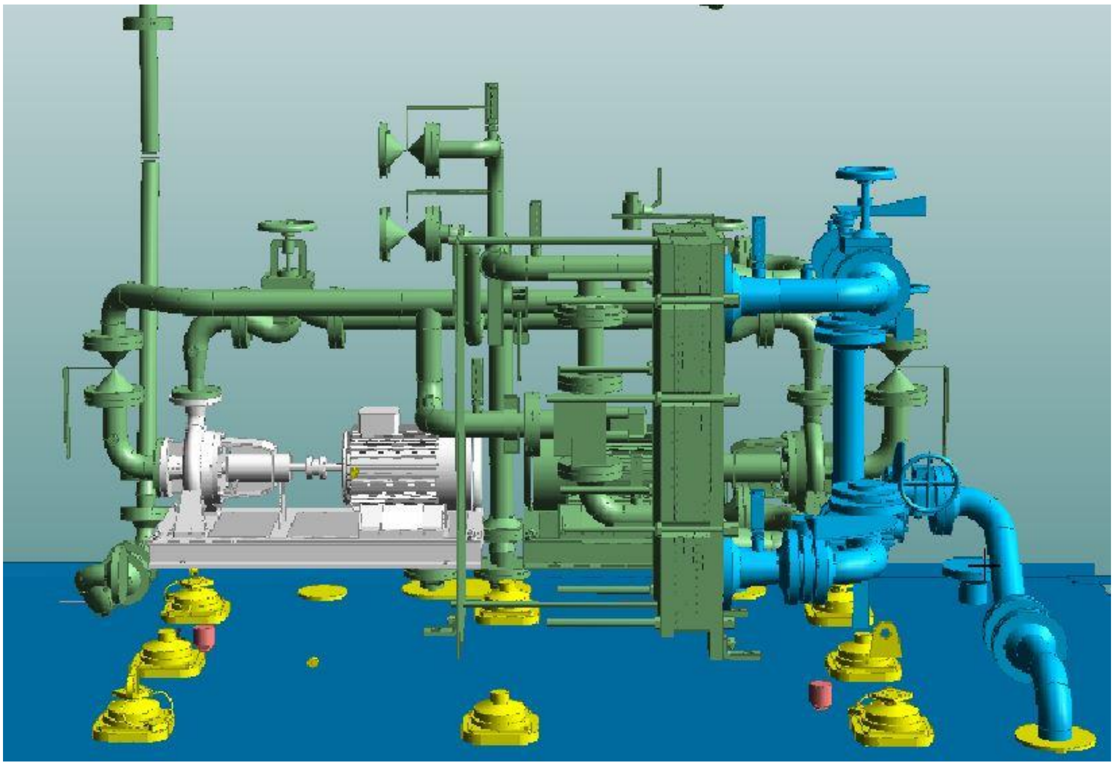


Figure 26, assembly cooling water system pump 1

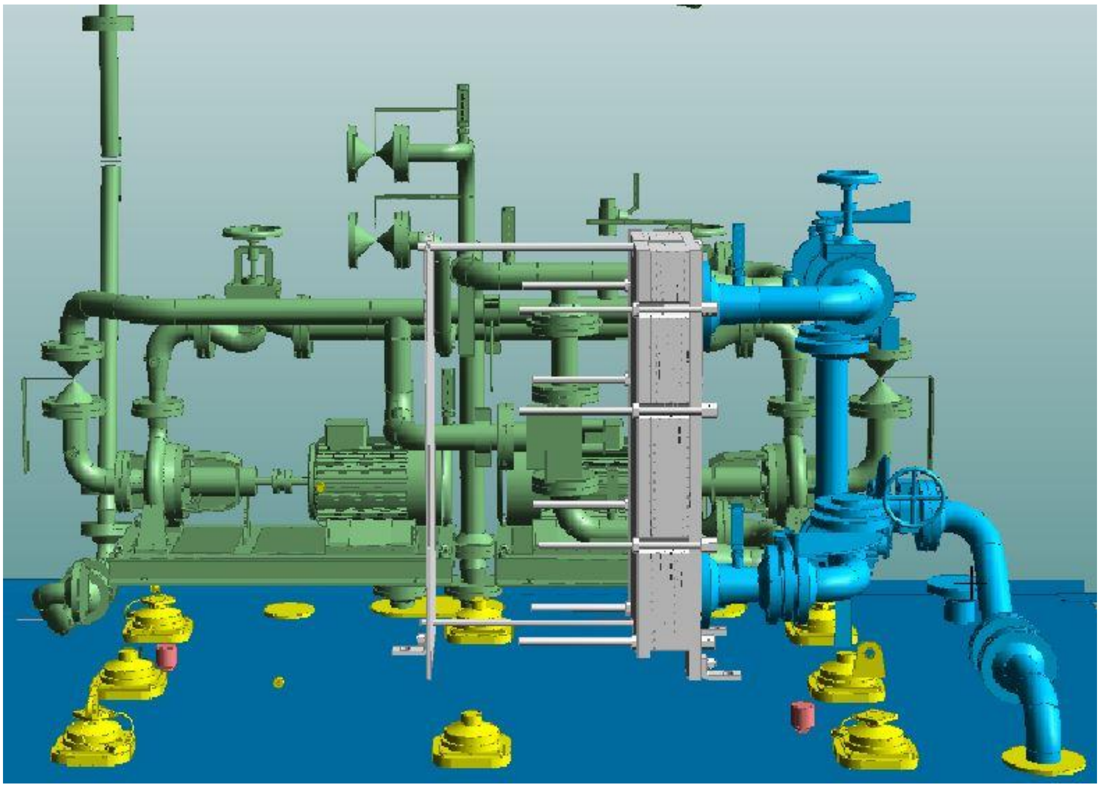


Figure 27, assembly cooling water system heat exchanger

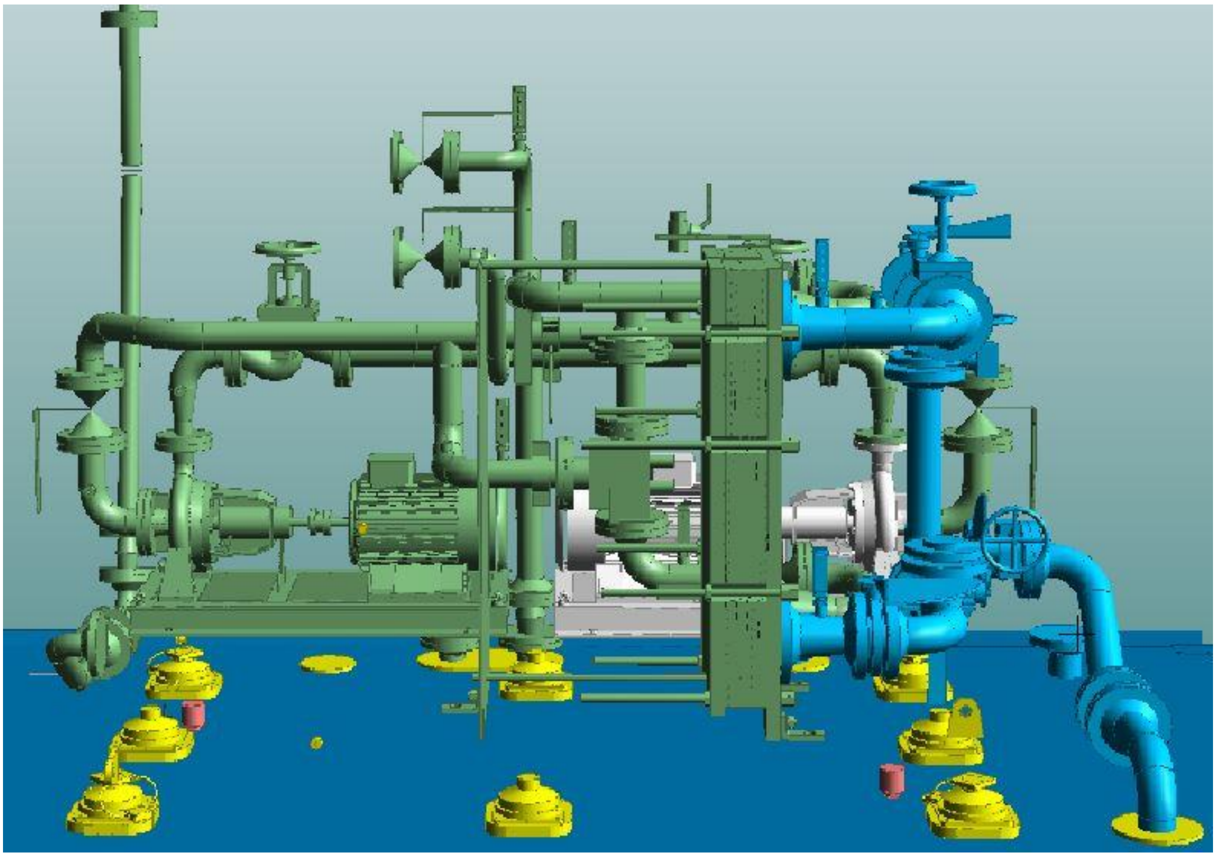


Figure 28, assembly cooling water system pump 2

### 7.1.2 Applications

An application represents a piece of software that is used in the engineering process. Within the architecture figure the application executes the “construct” actions that are in the activity diagrams of the engineering system, not to be confused with the “construct” use case in the interface. The applications represent the tools to model the ships design, which are divided in three categories for this system hull construction, outfitting, and one that combines the models in one 3D model.

The three symbols with a computer sign on the left represent the applications that are used within the architecture. The applications make use of the existing assemblies to form a model of a ship or an internal system in the ship. These applications are also capable of creating new parts that can form new assemblies.

### 7.1.3 Local CAD

When applications create new parts, it will be done in connection with the Local CAD database. The local CAD database can also store intelligent data but is located in the project and not in the CAD vault of the PDM system. The applications create the parts and the containing intelligent data in this working directory. It is only when the project is coupled to a PDM system that this data is organized in assemblies and parts at the level of the PDM interface.

### 7.1.4 Project A

The above explained elements represent one project. A project contains all these elements to fulfil the function of executing the engineering process. There can be multiple projects which all contain the applications and local CAD databases.

### 7.1.5 Interface

The middle part is the interface which manages the PDM functions. A clear explanation of the functions that are in the interface are given in Chapter 7.1. The interface functions that explain the different PDM use scenarios act as a doorway for the information exchange between the project with its accessories, and the PDM and CAD vault, whose functions have been explained in Chapter 7.2. This connection between project and the necessary data through the interface makes the modelling of a construction and outfitting system possible by the use of existing assemblies or creation of new parts.

### 7.1.6 PDM vault

The PDM vault contains the assemblies. These assemblies never leave the vault although Figure 29 to Figure 33 suggests otherwise. The assemblies are only depicted in the interface to make clear how the use of the assemblies is managed by the interface. The PDM vault only contains the static information of the assemblies the information that an assembly carries is explained in Chapter 7.2.1. The information that the assembly carries is essential for the use of PDM when modelling a hull construction or outfitting system.

The structure of how the assemblies are formed is not predefined, every use of this architecture could specify its own size and complexity of the assemblies. As mentioned before the ship is divided into assemblies, sub-...-sub-assemblies, and parts. It will be up to the user of the system to arrange the amount and size of the assemblies and sub-...-sub-assemblies but there will always be a top level assembly which is made up of sub-sub-....-sub-assemblies and finally single parts.

### 7.1.7 CAD vault

The remaining object in the architecture is symbolised by the CAD vault. Within this vault all the separate parts and the associated intelligent data of the parts that enables the features of the applications that make use of the parts are stored. The intelligent data of the single parts never leaves the CAD vault, the applications make use of this database when they have received the information about the location of the parts when the interface has established the connection between the application and the intelligent data. To make this connection, the interface uses the location data that is contained in the static data of the assembly.

***Note:*** *The figures in this chapter have assemblies depicted in the interface, this is just to make clear how the interface manages the use of the assemblies and in this way the arrows are more clearly visible. The architecture is meant to function with the assemblies in the PDM vault at all times.*

## 7.2 Construct assembly from scratch

The first scenario explains when the correct assembly does not exist and thus a new assembly needs to be constructed. In Figure 29 the actions that need to be performed to create a new assembly of new parts within the existing architecture are described. This scenario and the scenarios described after that are also shown in Figure 20.

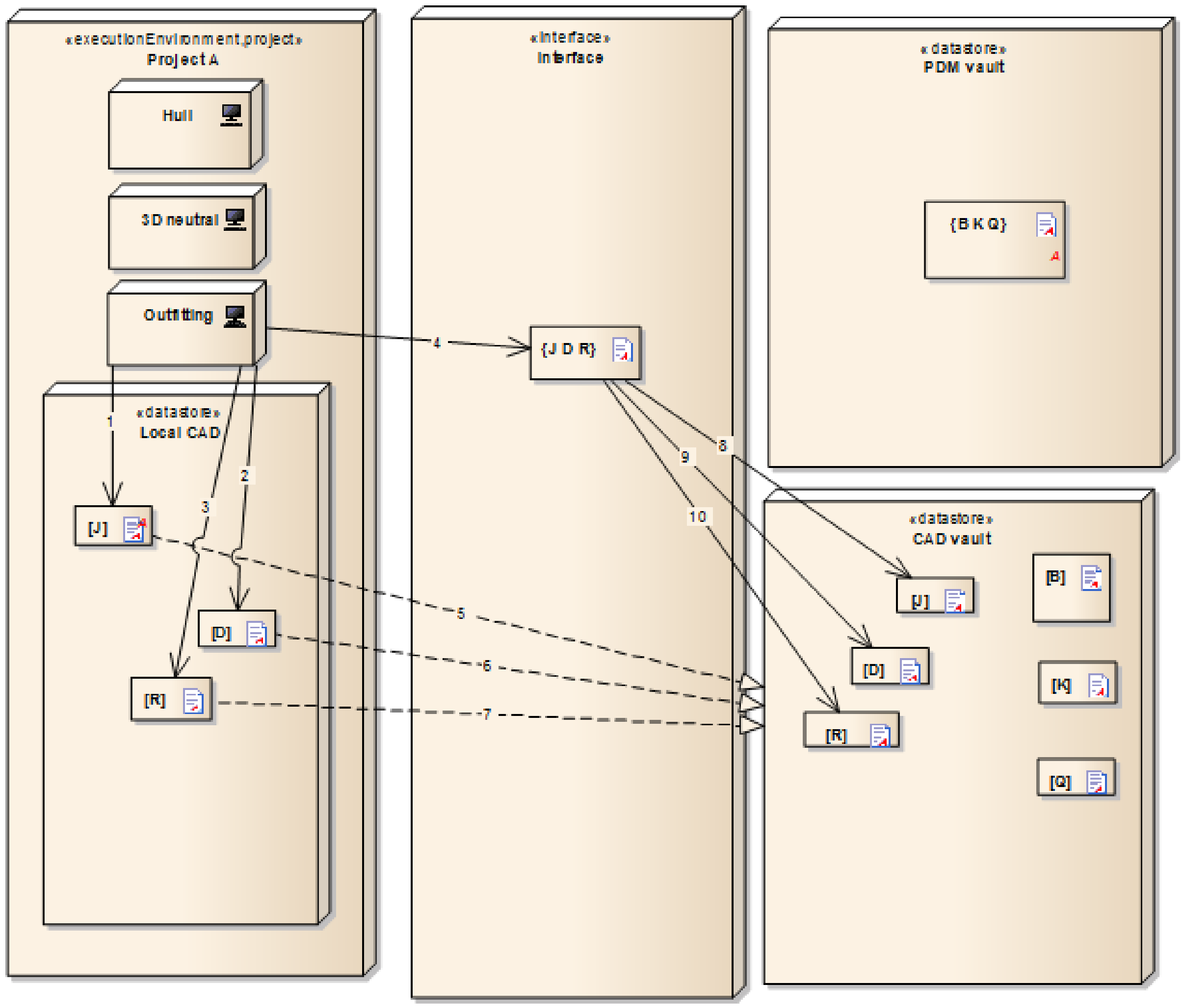



Figure 29, construct assembly from scratch

1. The first step in this scenario is the creation of the new parts as indicated by arrow 1, 2, and 3. In this case the outfitting application has modelled a new assembly containing three parts. The application has created the intelligent information of those parts in the local CAD database.
2. The new assembly is modelled and the created intelligent information of the parts is in the local CAD database. Now the assembly package is created by the interface and a connection is made from project to the assembly as shown by arrow 4. The information about how the parts are linked together thus form an assembly is stored in the assembly.
3. When the assembly is placed in the PDM vault the intelligent data of the parts is transferred from the local CAD database to the CAD vault. This is described by the arrows 5, 6, and 7.
4. To finalize this scenario the interface redirects the connection of the outfitting application with the intelligent data through the assembly. The resulting is that the connection goes through arrow 4 and then 8,9, and 10

## 7.3 Construct from existing parts

This scenario creates a new assembly but the single parts used are already stored in the CAD vault. Some arrows have the same number this means that it is the same connection but a new situation in the creation of the assembly.

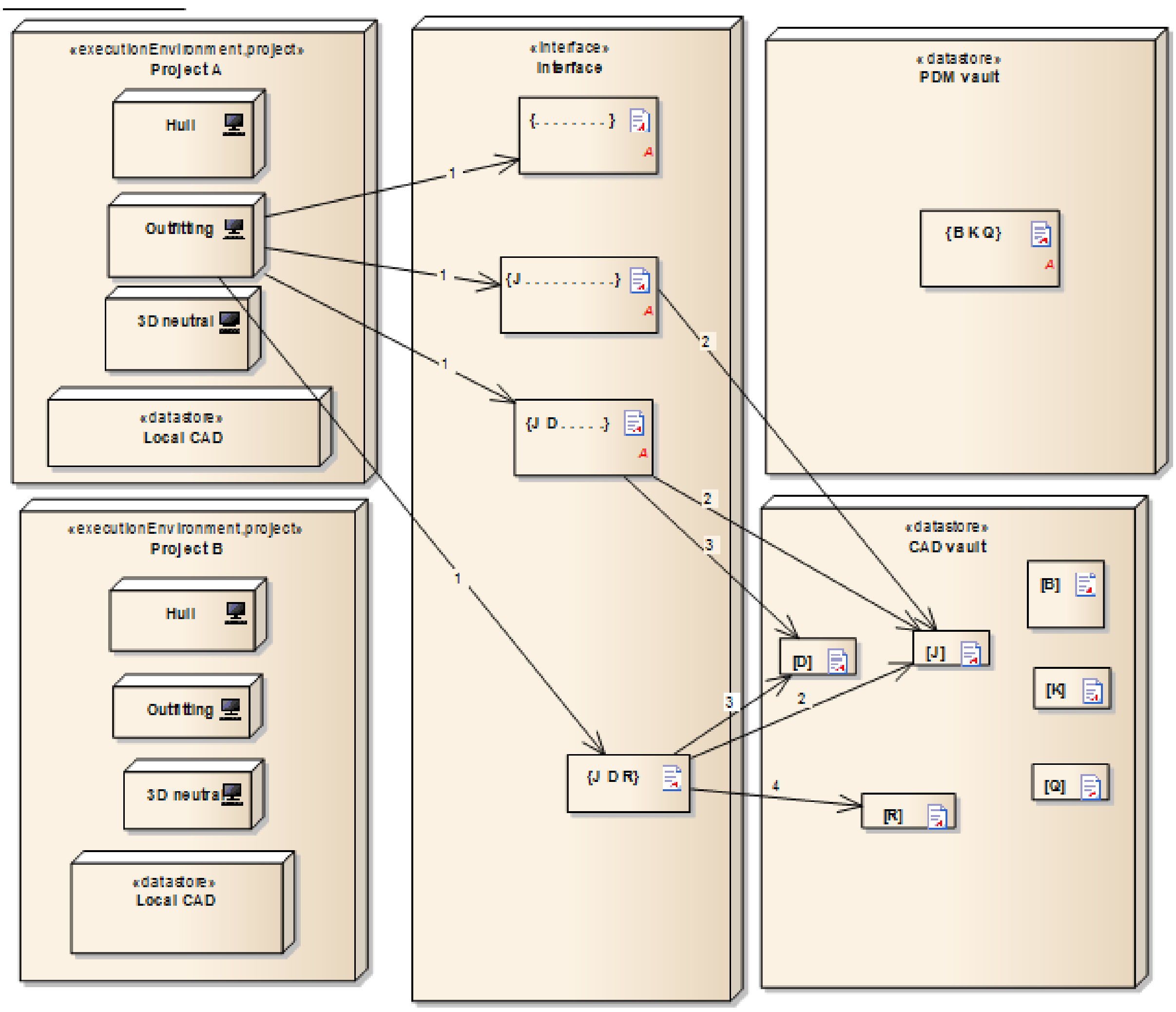



Figure 30, construct from existing parts

1. The interface creates an empty assembly. This assembly will be filled with the information about which parts form the assembly. The interface establishes a connection with the application represented by arrow 1
2. As new parts are used in the assembly the interface adds the necessary information of the part in the assembly. Also the connection from assembly to the intelligent data in the CAD vault is made, as can be seen in Figure 30. The first added part is represented by arrow 2, the second arrow 3, and finally to finish the assembly arrow 4. As shown in Figure 30, the application is connected to the intelligent data through the assembly.

## 7.4 Use existing assembly

The optimal use of the PDM system is when existing assemblies are used to model the ships construction and the outfitting systems. Figure 31 explains this optimal PDM use by means of one assembly.

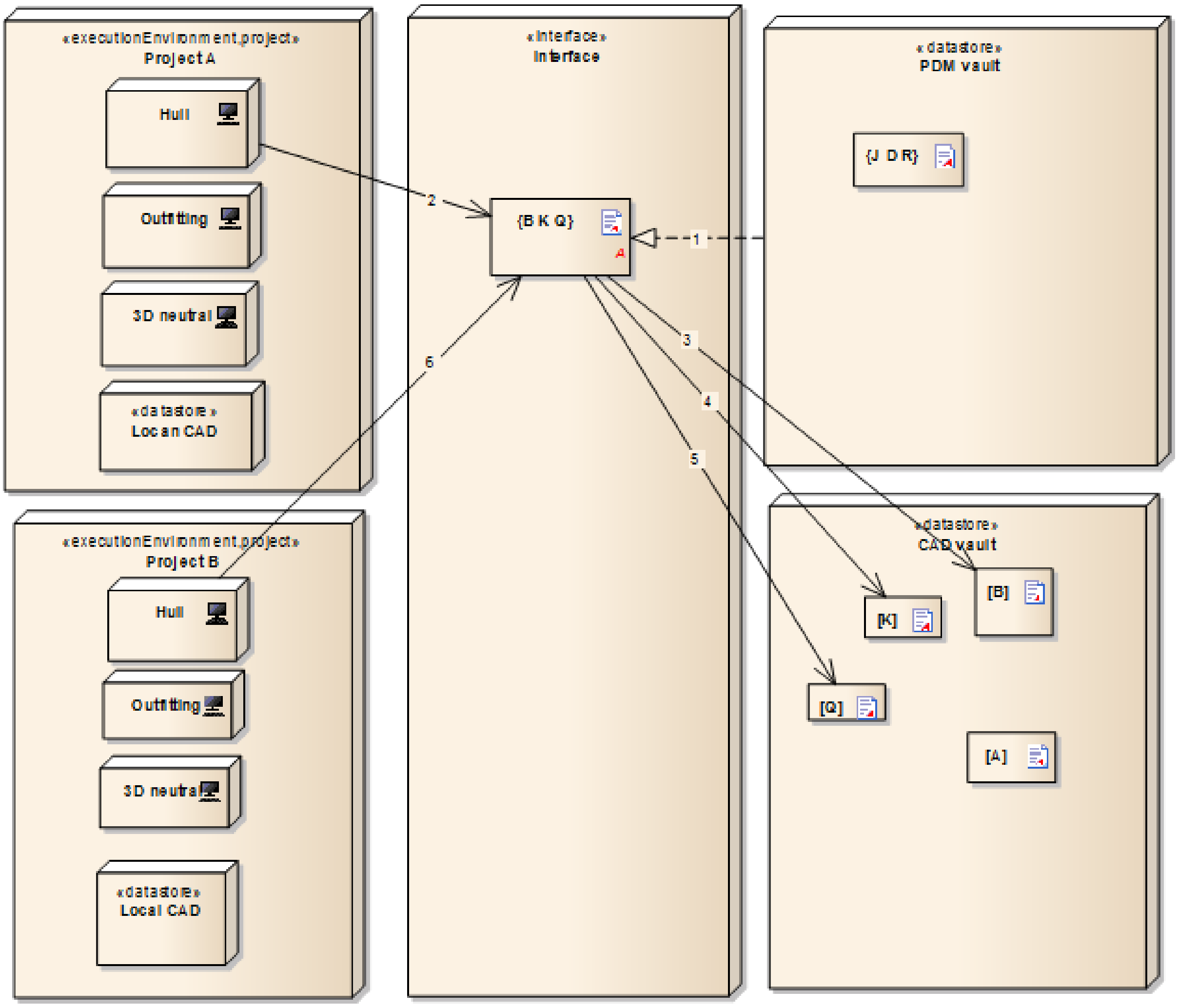



Figure 31, use existing assembly

1. The assemblies are stored in the PDM vault. The interface does a query for the right assembly at the PDM vault. The PDM vault presents the correct assembly to the interface so the information is available shown by arrow 1. The interface then connects the application with the assembly shown by arrow 2.
2. Within the assembly the location of the intelligent is stored. The application can now connect to the intelligent data as shown by arrows 3, 4, and 5.
3. When the application in project A uses an existing assembly it is also possible for another project to use that assembly. The interface establishes the connection of hull application in project B to the assembly where the location of the intelligent data is stored.

## 7.5 Change existing assembly with new modelled parts

While there are a lot of useable existing assemblies in the PDM vault it can happen that an assembly still needs some change. This situation is explained in Figure 32. Again, the arrows that have the same number which means that they represent the same connection but in another situation. In this specific situation the information of one part is removed from the assembly and also its connection. This is shown in the sub-situation with arrows 5 and 3-4. This scenario would also apply if for example a new modelled part is to be added to an assembly to form a new one.

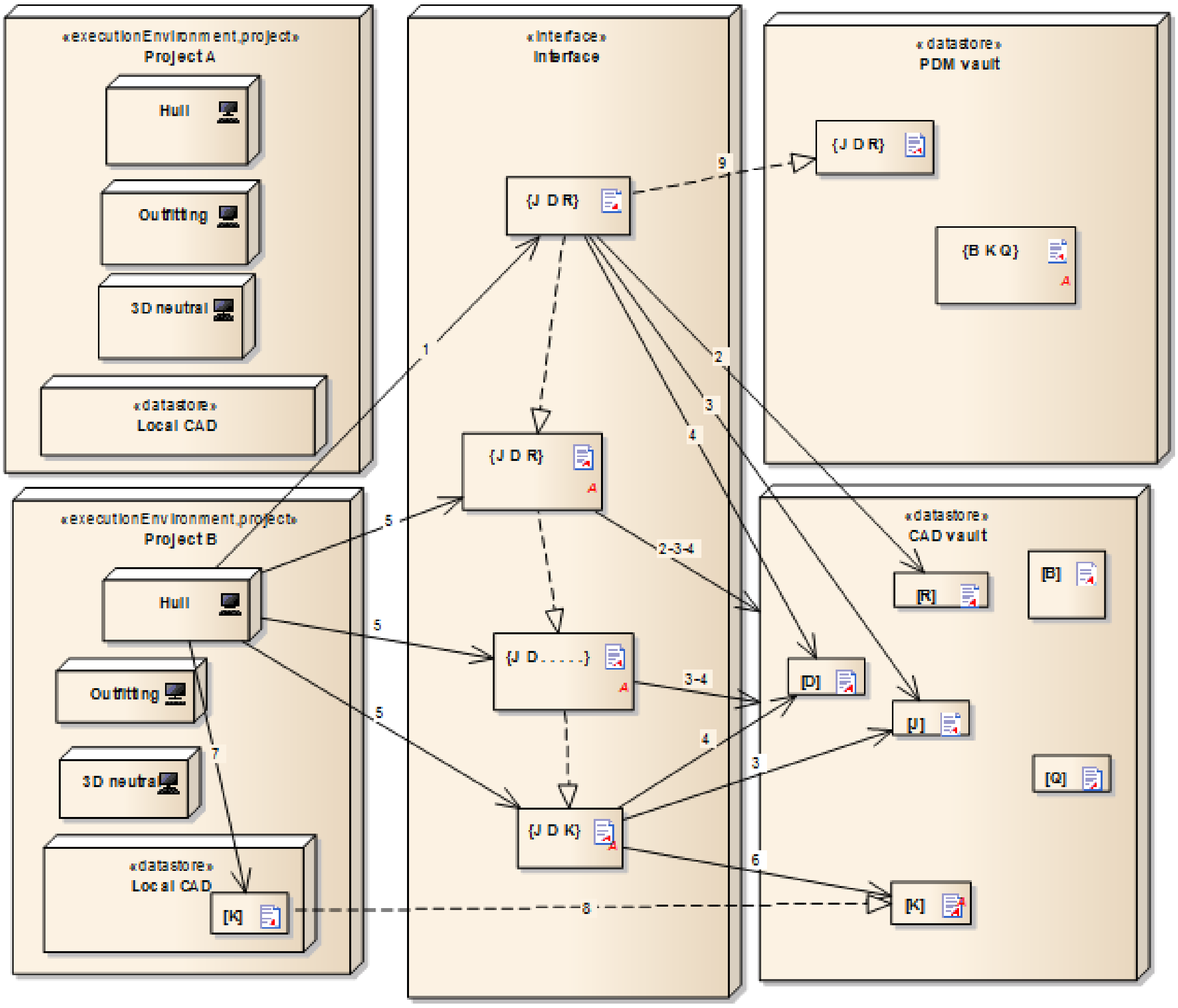



Figure 32, change existing assembly with new modelled parts

1. This scenario can only start if the use existing assembly or create new assembly first has been executed. This situation is depicted with arrows 1 to 4. The information about the intelligent data of part R is deleted entering into situation shown by arrows 5 and 3-4.
2. The hull application creates in the local CAD database the intelligent data of a new part.
3. The interface adds the data of that part to the assembly and transfers it to the CAD vault shown by arrow 8. Also establishing the connection of the application with the new part through the assembly. The new situation shown by arrows 5, 3, 4, and 6.
4. Finally, provided that the assembly is not used by other projects, the connection with assembly {J D R} is broken displayed by arrow 9.

## 7.6 Change existing assembly with existing parts

Just as the scenario from Figure 32 an assembly can be changed but the change is done with an existing part. Figure 33 depicts this scenario, as can be seen from Figure 33 there are no new parts modelled in the local CAD data base.

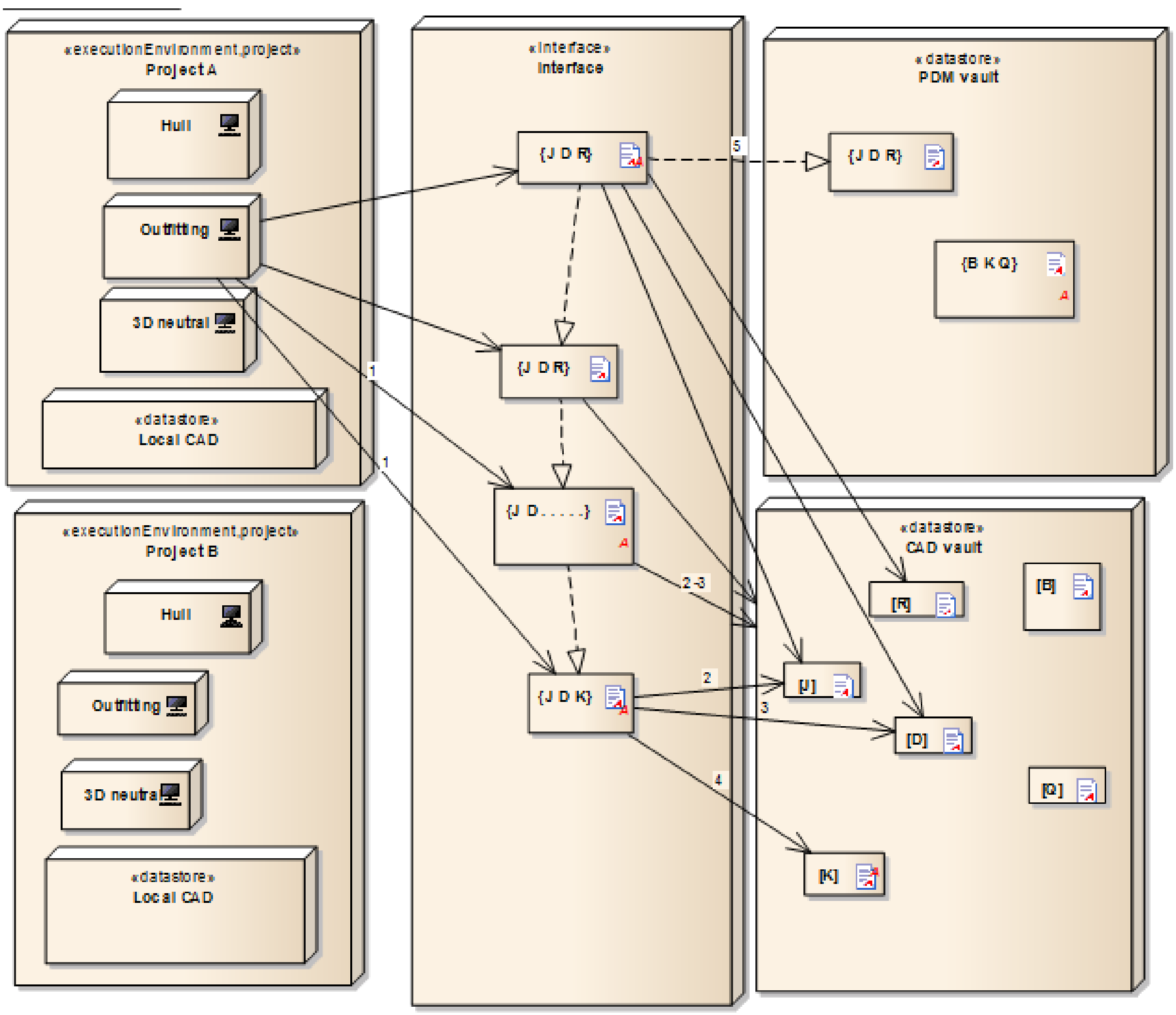



Figure 33, change existing assembly with existing part

1. Up until step 3 from the scenario “change existing assembly with new modelled parts” this scenario is the same. The explanation will start from the situation shown by arrows 1 and 2-3
2. Instead of creating a new part in the local CAD database, a part that is already in the CAD vault will be used. The interface adds the information about the new part to the assembly in the interface and establishes the connection between the application and the new part through the assembly, resulting in the situation shown by arrows 1, 2, 3, and 4.
3. Provided that the assembly is not used by other projects the connection with the assembly {J D R} is broken, displayed by arrow 5.

## 7.7 “magic wand design”

As shortly explained before, the “magic wand design” is used as a tool to pull a lot of the engineering work forward to the early design. The way to do that is with the use of “magic wand design” , the execution of this way of working is supported by the 3D neutral application, expressed by the bottom cube within the project A cube containing a computer symbol shown in Figure 34.

The 3D neutral application is normally used to combine the hull and outfitting application models and depict them by means of an easy to use small file size, a so called neutral format.

This 3D depiction of the complete ship through a small sized file increases the accessibility and visibility of the assemblies and sub-assemblies that have been finished and helps compose a new project from the predesigned parts.

The idea of the “magic wand design” is to reverse the normal way of working from engineering the assemblies and create its intelligent data without going through the complete design phases, and to assemble the ship using the 3D neutral applications and the neutral format files. Basically the ship is assembled using only the 3D depictions of the assemblies. When this 3D model of neutral formats is created the necessary intelligent data will be connected to the project to make sure that the engineering can be finalized and the project contains all the information. This way of working can only be sustained by a database of existing assemblies of which the intelligent information is also available. Figure 34 shows a small representations of how the magic wand design works.

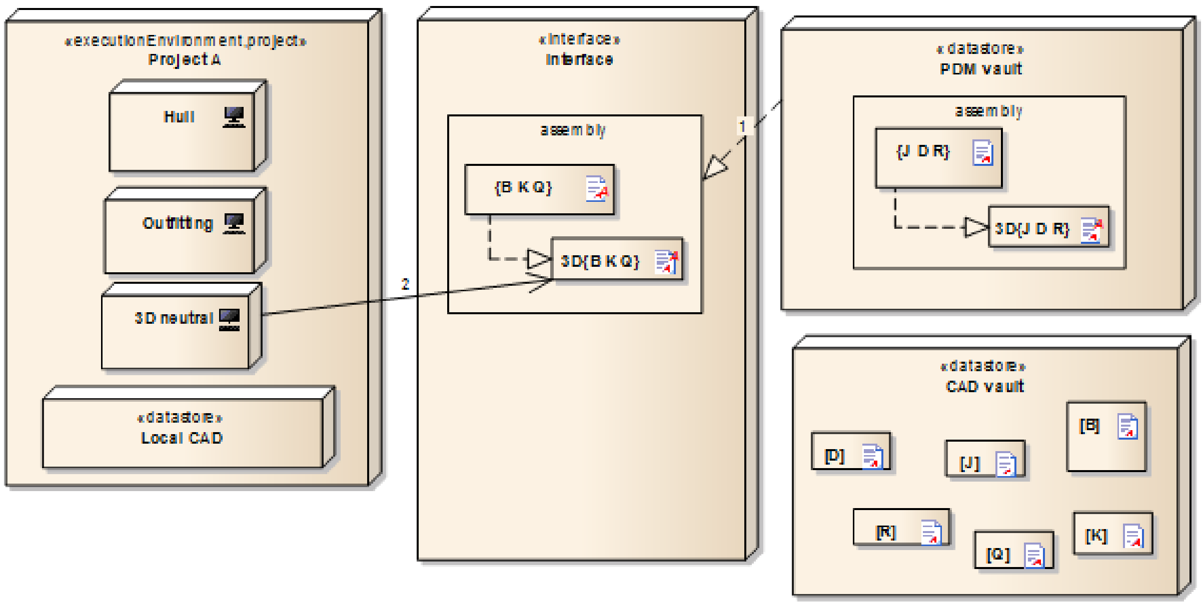



Figure 34, "magic wand design"

The PDM vault contains all the assemblies, which in turn contains the neutral format 3D model of that assembly. When a project starts with the early design and uses magic wand design the assemblies are found in the PDM vault. To use the neutral formal 3D model the complete assembly needs to be moved to the interface as shown by the dotted arrow in Figure 34. When the assembly is moved to the interface, the interface will establish the connection between the application and the neutral format file as shown by the straight arrow.

When the magic wand design is finished the assemblies will be used in the same way as the scenario: *use existing assembly*, which is described above. The user will be able to complete the design and production of this project in the usual way.

The execution of this “magic wand design” needs to be done in early design thus the use case that describes the execution of the magic wand design is located in the global ship information system as described in Figure 8 of Chapter 6. The next section describes the magic wand design use case.

### 7.7.1 "Magic wand design"

Preliminary or early design is the first global activity where the ship starts to get a shape within the engineering process of shipbuilding. The growing use of PDM in shipbuilding is not only powered by the need to make the engineering process as it is more efficient but also to transfer some engineering work to early design. Within the global ship information system, in the use case model this engineering work in early design is represented by the use case "Magic wand design". Figure 35 shows the activity diagram of this use case.

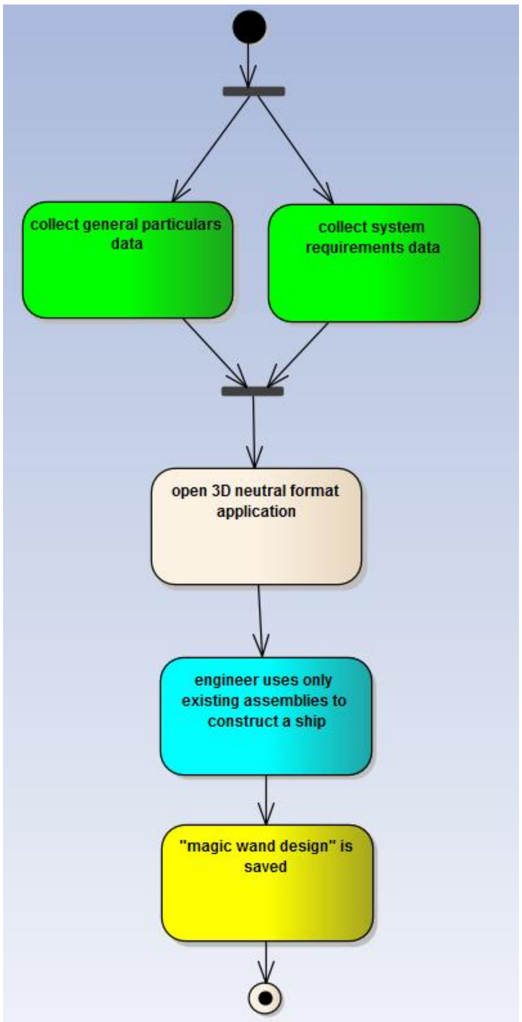



Figure 35, "magic wand design"

The "magic wand design" use case is introduces to make it possible to design the ship in a "Lego" fashioned way using stored assemblies that are made available by the PDM system. This "Lego" fashioned modelling of a ship is done in a 3D neutral format application as can be seen in the third activity from Figure 35. the "magic wand design" is a beneficial tool that arises from a proper use of a PDM strategy and system as explained later in this chapter.

## 7.8 Standard module intermezzo

As mentioned earlier, the goal of the implementation of PDM is to reduce the engineering time. When as in scenario three assemblies are re-used the engineering time should be reduced at least of those single assemblies. Other scenarios describe that existing assemblies can be changed so that the assembly fits in the ship or an assembly is change because it is no longer up to date or maybe the research and development department has found a way to make an assembly better. This chapter focuses on the assemblies that are used in the outfitting department. Just as the hull assemblies the outfitting assemblies vary in size. Very often it occurs that a system containing multiple small and/or big components form one assembly. When one of those components is out of date or there is a better version available, it needs to be changed. In a normal situation scenario four or five will be executed to make sure a more up to date assembly is in the system. The problem is that now the most time reducing scenario is not executed.

The use of assemblies also means that they need to be updated. But then the time reduction will be less. How can the assemblies be updated in another way do that the use of PDM will always go through scenario three. The following described functionality is not necessary for the use of assemblies but can be an extra feature of the PDM solution.

Normally the assemblies as they are stored in the PDM system have their actual shape. The reason for this is that it is just very logical and when assemblies have their actual shape it can be advantageous when routing cables, piping, or ducting as they sometimes almost go through the assemblies. When an assembly and its containing components are shaped like the smallest beam or cube that can fit the components of the assembly it would be possible to replace a component for a better one before the assembly is used in a ships design. The modelling of a ship would be an extreme form of a "lego way of engineering" . this makes it possible for a research and design department to improve the assemblies continuously as long as they stay within the determined beam or cube boundaries. Because the structure of how the assemblies should be stored is not predefined for the system to work, this "lego way of engineering" is a good possibility for shipyards to implement. Figure 36 shows the before discussed cooling water system assembly boxed in a module.

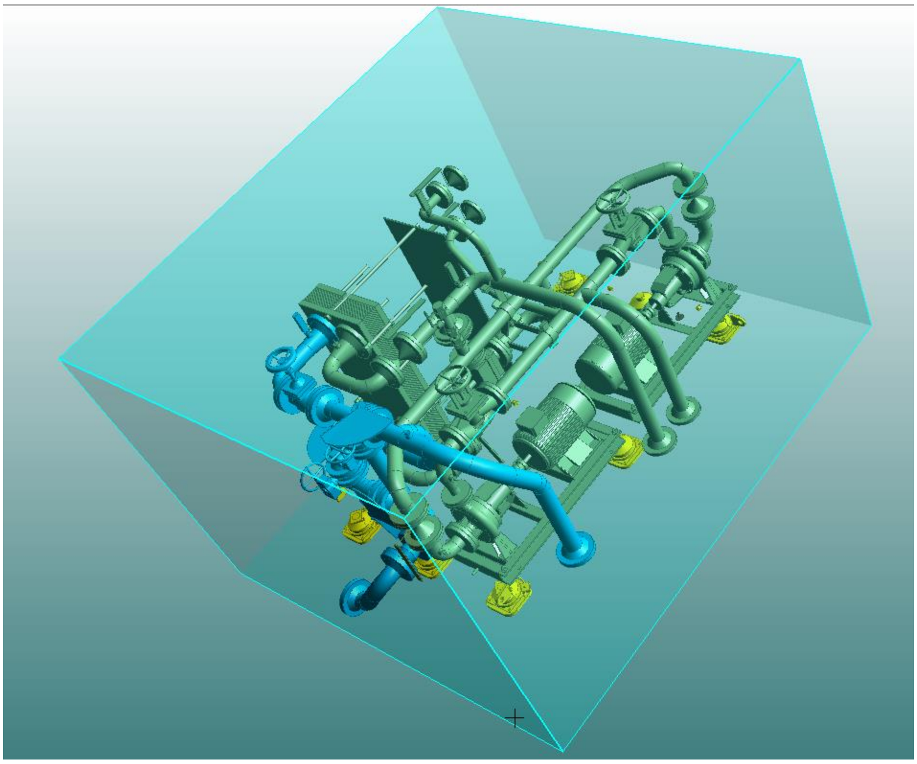


**Figure 36, assembly module**

# 8 Qualitative measurement of PDM improvements to the engineering process

The use of PDM together with CAD/CAM software can pose difficulties which have an effect on the devised solution of the CAD/CAM PDM interface. Overcoming these difficulties is however not the only goal of this research. The engineering process of shipbuilding must benefit from the use of PDM and those benefits need to be indicated. The PDM interface solution is at this stage a theoretical concept design, it is hard to do quantitative measurement on its added value and its performance.

The engineering process as described in Chapter 6 has a focus on the actual engineering tasks thus the modelling and calculation of the ships design. The use cases where the hull construction and outfitting systems are created, almost all contain the activities collect, construct, and save. In both situations, with and without using PDM for the engineering process, these activities need to be executed. When PDM is integrated in the engineering process, these actions make use of the PDM system. The goal of this section is to estimate roughly the potential benefits in terms of time saving by making use of the proposed CAD/PDM architecture when executing the actions collect, construct, and save.

The engineering activities described in detail in previous chapters are divided in two categories: hull construction and outfitting system. Within these categories collect, construct, and save actions are counted. Table 4 shows the result of the count of these actions for each engineering category.

| | | | collect | construct | save |
|---|---|---|---|---|---|
| hull construction | | make production information | 1 | | 1 |
| | | make construction drawings | | | |
| | | model hull construction | 1 | 1 | 1 |
| | | model big construction elements | 1 | 1 | 1 |
| | | model stiffening members | 1 | 1 | 1 |
| | | scantlings determination class check | 1 | | 1 |
| | | make 3Dmodel of hull | 1 | 1 | 1 |
| | | strength calculation | | | 1 |
| | | buiding plan (panel, block, hull erection) | 1 | | 1 |
| | | react to change requiest | | | |
| | | | 7 | 4 | 8 |
| outfitting system | | dependancy between system | | | |
| | | engineer small outfitting | 1 | 1 | 1 |
| | | enigneer electricity system | | | |
| | | make power plan | 1 | | 1 |
| | | place components(switchboard, distribution gear) | 1 | 1 | 1 |
| | | route cabling (cable trays) | 1 | 1 | 1 |
| | | engineer airco unit | | | |
| | | route ducting | 1 | 1 | 1 |
| | | find and place airco unit | 1 | 1 | 1 |
| | | engineer foundation of airco unit | 1 | 1 | 1 |
| | | airco system calculation | 1 | | 1 |
| | | find/create schematics airco system | 1 | | 1 |
| | | engineer cooling water system | | | |
| | | route piping | 1 | 1 | 1 |
| | | find and place pump | 1 | 1 | 1 |
| | | engineer foundation of pump | 1 | 1 | 1 |
| | | cooling water system calculation | 1 | | 1 |
| | | find/create schematics cooling water system | 1 | | 1 |
| | | | 14 | 9 | 14 |
| | | | 21 | 13 | 22 |

Table 4, count of PDM linked activities in use case diagram

The Table 4 shows a list of all use cases of the hull construction system and outfitting system and their corresponding activities. However, most of the use cases need to be executed multiple times for a design to be finished and not all contain the three before mentioned actions. This means that for a certain project a multiplication needs to be made to the 21+13+22=46 actions to derive the correct amount of use cases that belong to a project containing a complete ship to measure the correct engineering time. The use cases that are listed in table 1 are the different use cases that are in the hull construction and outfitting system of the virtual bock described in Chapter 5.1. the collect, construct, and save actions are the actions that overcome a change when PDM is connected to the engineering system. This is because these actions make use of the PDM functionality and thus should benefit from the improvements that are described in Chapter 2, and vary between reduction of time, error reduction for the improvement of quality.

## 8.1 Use Case Breakdown Structure

To develop the PDM solution a virtual block was used as guidance for the description of the engineering model that is described and used in this research. To give validation that the situation as described for this research is representative for a complete ship a Use Case Breakdown Structure (UCBS) of two ships is made. This UCBS is a description of all the use cases that are necessary in the engineering process of a complete ship. These use cases have than been divided over eight work specialisms. The use case arrangement shows that the work specialisms that contain the biggest amount of work, five and eight, also are the works specialisms that benefit the most from PDM.

Two UCBSs have been created by the use of two existing ships of which the data is collected from significant ships 2007 and 2010. The used ships are: DEO VOLENTE a multipurpose/heavy lift ship and the PACIFIC ORCA a wind farm installation vessel. The significant ship pages that contains the data of these ships can be found in appendix B. The two complete UCBSs which are an expansion of the engineering model described in Chapter 6 are shown in Appendix C. Table 5 shows by percentage of the total amount of use cases how many use cases are assigned to every specialism of the engineering process. The engineering man hours have been determined through ‘expert estimation’, ship type not revealed for reasons of confidentiality. Which means data of other ships have been used for this estimation.

| DEO VOLENTE | | | | | PACIFIC ORCA | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | specialisms | amount | % | manhours | | specialisms | amount | % | manhours |
| 1 | main engine, gearboxes, thrusters | 45 | 9% | 1209 | 1 | main engine, gearboxes, thrusters | 98 | 13% | 2461 |
| 2 | accommodation | 43 | 9% | 1036 | 2 | accommodation | 94 | 12% | 2102 |
| 3 | automation | 34 | 7% | 930 | 3 | automation | 71 | 9% | 1861 |
| 4 | hydraulics | 38 | 8% | 1054 | 4 | hydraulics | 55 | 7% | 1861 |
| 5 | hull/structure | 127 | 27% | 3543 | 5 | hull/structure | 220 | 28% | 6632 |
| 6 | hydrostatics | 16 | 3% | 414 | 6 | hydrostatics | 12 | 2% | 610 |
| 7 | secondary steel, hatches, windows, floors, walls | 51 | 11% | 1538 | 7 | secondary steel, hatches, windows, floors, walls | 63 | 8% | 2473 |
| 8 | mechanical engineering systems | 121 | 25% | 3317 | 8 | mechanical engineering systems | 161 | 21% | 5706 |
| | | | | | | | | | |
| | | 475 | 100% | 13041 | | | 774 | 100% | 23707 |

**Table 5, use case distribution of two existing ships over specialisms**

The ships are divided into blocks for the estimation the amount of use cases that contribute to the accommodation and hull/structure mechanisms. The amount of blocks determines how often the hull construction use cases should be executed. The amount of outfitting use cases that are divided over the other specialisms, are determined by the amount of separate systems and over how many blocks they spread, e.g. the ballast water system containing two pumps gets appointed one “ballast water system calculation” use case and two “find and place pump” use cases. The amount of “route piping” use cases is based on the amount of blocks the pipes cross.

Table 5 shows that for both ships the hull/structure and mechanical engineering system specialisms have the biggest contribution to the amount of use cases. The systems of which the use cases have been assigned to these groups contain the most parts which can form standardized assemblies. This means that USBC shows that the amount of use cases that benefit from the use of PDM is significant. These two ships are compared because the DEO VOLENTE is a relative simple vessel and the PACIFIC ORCA a complex one. Not only is the PACIFIC ORCA much bigger which causes the higher amount of use cases also the duration per use case of the PACIFIC ORCA in specialisms five and eight is longer. This means that in the engineering of a complex ship has much more man hours in the specialisms that benefit the most from PDM. The average duration of the use cases per specialism is shown in Table 6.

| | DEO VOLENTE | PACIFIC ORCA |
|---|---|---|
| 1 | 26.9 | 25.1 |
| 2 | 24.1 | 22.4 |
| 3 | 27.4 | 26.2 |
| 4 | 27.7 | 33.8 |
| 5 | 27.9 | 30.1 |
| 6 | 25.9 | 50.9 |
| 7 | 30.2 | 39.3 |
| 8 | 27.4 | 35.4 |

Table 6, hours per use case

## 8.2 Performance indicators

In the above introduction it is mentioned that an engineering time should be measured. Engineering time depends on multiple factors and can be measured according to certain indicators. These so called performance indicators can quantify the performance of a process. In order to obtain the correct quantified performance the indicators need to be chosen very thoroughly. When a measurement needs to indicate an improvement or deterioration of a process induced by a change in the process there need to be two clear situations on which measurements should take place. There needs to be a clear “before situation” representing a baseline for all further measurements; within the scope of this research, the before situation is the described engineering process without the PDM improvements. The after situation contains the PDM improvements. Within these two situations there need to be equal indicators that can be compared to receive measurements that give a fair result on the changes to a process. Both in the before and after situation the engineering process is executed as it is described, by introduction of the PDM system there will be no changes to this engineering description. Not only are the PDM improvements directed towards engineering time reduction, it should improve the quality of the process.

For the measurement of quality an error needs to be defined. Figure 37 shows schematically how an error can be considered. Every time a use case needs to be done again whatever the reason it is counted as an error.

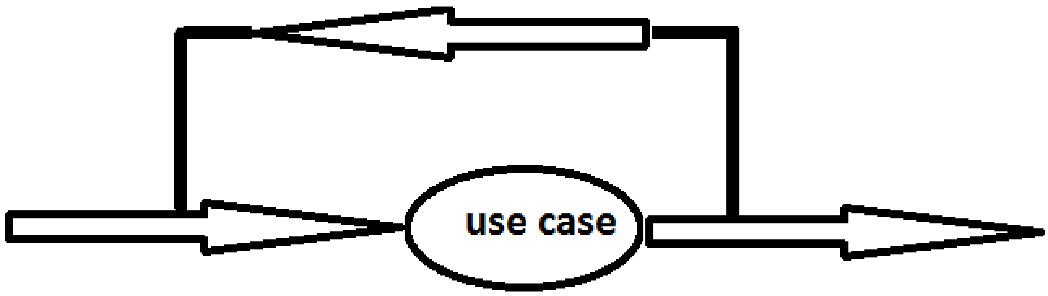



Figure 37, error

The improvement of quality is measured by counting errors that occur during the process. For every project there will be a Work Breakdown Structure (WBS) the advantage of the engineering description being in well described work packages (use cases) is that a use case breakdown structure can be made similar as the use case breakdown structure shown in Appendix C. This gives a clear description of the work that is in a project and thus a good way to compare two situations with the same use case breakdown structure, one without PDM improvements and the other with PDM improvements.

Two major indicators can be defined to measure the results of the change in process. The PDM improvements lead to a shorter engineering time and better quality of the engineering as described in Chapter 2. The quality of the engineering can be measured by Equation 1:

$$quality = \frac{first\ time\ approved\ executed\ use\ cases}{total\ executed\ use\ cases}, \qquad [1]$$

Equation 1, which divides the amount of use cases of a project that have been finished with immediate approval and do not have to be redone or re-evaluated, between the total amount of use cases that need to be executed for the project to finish. It is important when this formula is used that the use cases that are used in the compared original and changed process are equal. Equation 1 can be used to measure the quality of different sections of the engineering process or the complete process. The following enumeration describes the different uses of Equation 1:

- **Complete process:** When the quality of the complete engineering process needs to be measured all the use cases of the hull construction and outfitting system are used for this measurement. This gives the quality of the whole engineering process.
- **Partial process:** Within the engineering process, there is a clear distinction between the engineering of the hull construction and outfitting systems. These activities are usually performed by different departments. It is therefore important that the quality metrics that are measured at the macro level are also measured for the sub-systems of the process. In this way very specific parts of the engineering can be measured, under the condition that in both the measurements, old and new situation, the measured use cases are the same. Also a difference in quality between parts of the same project can be measured and thus a more concentrated

assessment of problems in the engineering process can be made. Sub-processes for which it can be beneficial to measure the quality indicators are:

- Calculation use cases; scantlings calculation, make power plan, system calculation, find/create schematics.
- Model use cases; model hull construction, model big construction elements, model stiffening members, model outfitting component, model outfitting foundation, route pipes, cables, or ducting.

Even though a reduction in errors probably reduces the engineering time the main contribution of the reduction in errors is to improve the quality of the process. The improvements of the PDM are also meant to support performing engineering tasks quicker and thus reducing the engineering time. Equation 2 describes the performance indicator that measures the improvements or deteriorations of the engineering time:

$$engineering\ time = \frac{\text{budgetted use case work breakdown time}}{\text{actual engineering time}}, \qquad [2]$$

The engineering time performance indicator, which shows whether the process is done efficiently. It gives an indication of the amount of time losses that occur during the execution of the use cases. This means that the engineering time performance indicator is not a comparable measure of the duration of the engineering time of two projects because the budgeted (planned) use case work breakdown time could change when PDM is implemented. The engineering time indicator is a quality measure of the engineering time and the planning. It says something about the difference between the actual engineering time and the time it should have taken. When budgeted use case work breakdown time is kept constant between two equal processes it says something about duration but this would be no different than simple time measuring, which will be explained in the next section. PDM does not only shorten the engineering time but it improves the visibility and availability of information within the process thus resulting in less delays in the process which means no or smaller increase in the actual engineering time. These kind of improvements can be measured using the cycle time indicator. The shortening of engineering time which is a result of PDM implementation does not need a performance indicator. The engineering times of a project with and without PDM should be compared and investigated whether there are differences. This basic engineering time difference will be explained later in this chapter. Just as for the quality performance indicator the cycle time indicator can be applied to either the complete process or parts of the process as explained above in this chapter.

## 8.3 Reduction of time

By reduction of time is meant that the use of the PDM system causes the actions that are counted in Table 4 take less time when the PDM system assists the engineering process. The use cases that are counted in table one are the different use cases that are present in the engineering system. Most of the use cases will be present multiple times in the use case breakdown structure of a project. This also means that the time reduction, by previous mentioned advantages, within a use case that is present a lot of times in a use case breakdown structure will result in a better overall time reduction.

The reduction of engineering time can be the result of different advantages of the PDM system. The improvement of information availability for engineers working on the same project or even engineers working on different working sites, results in engineering time reduction. When engineers take less time to find the information thus shortening the time to execute the use case, time is reduced. Increased availability of information is made possible by the PDM system. The collect activities within the use cases will therefore take less time because information is found quicker. Just as for collect, there is another activity within the use cases that will have a shortened execution time by the use of PDM. The construct activity is the activity where the engineer does the modelling of a certain part of the design and it should also benefit from the PDM implementation. When product data is stored it is evident that the data will be reused, especially when it is CAD/CAM data. When previous modelled assemblies or parts are re-used in a design it will take a lot less engineering time than when an engineer has to model the assembly or part from scratch. Re-use instead of new modelling results in a significant engineering time reduction. The final activity that connects the engineering process to the PDM system is the save activity. This activity finalizes all the use cases as can be seen from Table 4. The save action does not really store the modelling data but it finalizes a use case by triggering the status handling, thus it makes clear in the system that a use case is finished. The storing of information is done during the construct action. As mentioned before there are use cases that contain preparation activities like calculations or making schematics and the use cases that contain the actual modelling activities in this arrangement the time reduction of engineering process with PDM will be explained

As described in Chapter 2, when introducing the improvements of the use of PDM, an important aim of the PDM system is to be able to re-use earlier engineered assemblies. These assemblies can consist of hull construction components, outfitting components or systems, and even assemblies that consist of both hull and outfitting. It is evident that when an assembly is re-used it will take less time than when a new assembly is engineered. Re-use can also mean that an assembly is re-used and then slightly modified to meet the requirements of a specific project. The time reduction is presumed to be an outcome of the re-use of assemblies made possible by this PDM system. Time reduction is also presumed by the process to have less errors and conflicts as can be assessed by Equation 1.

## 8.4 Measurement

The section of the performance indicators discusses how the factors influencing the indicators are determined by the use cases, either combined in subgroups or taken as a whole, considering all the use cases of the process. The engineering description is built up of use cases which basically are work packages. These use cases make it possible to have an assessment of the process one level higher than the actual actions of the engineer as described in Table 7. The use case arrangement has divided all these actual action of the engineer to physical parts of a ship. It is now possible to have a very detailed work breakdown structure (use case breakdown structure) but no chaos is created by the high level of detail, because this detail is enclosed in the use cases.

| level | description |
|---|---|
| Use case | Group of activities that form a product based work package |
| Activity in activity diagram | Work task: make a 3D model, collect information, do calculation |
| Actual engineer actions | Open files, do modelling work, walk to printer |

Table 7, level of work

The before mentioned Use Case Breakdown Structure makes it possible to compare the engineering of different projects. In shipbuilding, it is very rare that two of the same ships are built, thus making it hard to make a good comparison between two projects. On the other hand, different ships may, to a great extent contain similar systems and compartments/blocks/sections. The time it takes for use cases to execute these similar systems and compartments/blocks/sections can be measured. Because the activities in these use cases are practically the same they may very well be compared and conclusions can be drawn on the efficiency of working in two different projects.

To make representative measurements the time it takes to execute the selected use cases should be measured. It is convenient that all the use cases start with a collect action and end with a save action. These actions should function as measurement points when the process is executed. When the engineering time of two certain groups of use cases is measured, differences between the process with PDM and without should become visible. This result will be a pure difference in duration of these certain use case groups. It will be assumed that the use of the PDM interface will speed up the process but there can still be other possibilities for time reduction. Also it might be the case that the duration of the measured group of use cases that made use of the PDM interface is not decreased. To make sure that a fair comparison between the measured durations can be made the before explained performance indicators need to be the same for both measured group of use cases.

# 9 General feasibility

Before the feasibility of the proposed architecture for NUPAS-CADMATIC will be evaluated, some observations on the general feasibility of applying PDM in ship Design and Engineering should be made.

During this research towards the use of PDM in shipbuilding, it became clear that not every aspect of PDM carries the same amount of importance. Even though there exist PDM software packages that claim to provide complete PDM, it is not certain that the right aspects of PDM have been provided for the shipbuilding process when taking into account that not every shipyard has the same requirements for a PDM system. During this research it became more clear over time that the M (management) in PDM has gotten an important role in the implementation of a PDM way of working and its success. When looking at the most important functions of PDM described in Chapter 7, most of them have to do with management, which is closely coupled with certain choices for change, user rights, or the shape and size of building blocks/assemblies of a ship. Of course, the CAD/CAM-PDM system used needs to be able to facilitate and execute those managerial PDM functions but it is infeasible for the system alone to carry out all the managerial functions of PDM. The engineers and project manager may also have to apply changes to their way of working to make such an implementation a success. An important decision has to be made concerning how much of the PDM will be captured on a system level and how much on human level.

Within the shipbuilding industry, the engineering process is mostly shaped to handle an engineering to order product development. This way of doing engineering originates from the fact that the level of series or continuous production is extremely low in shipbuilding; one might even say that every built ship is unique. This way of working means that the shipbuilding industry needs to be extremely flexible. Figure 38 shows how the shipbuilding industry is located in the “flexible production universe” compared to some other industries. The flexible production universe can be mapped by making use of two dimensions: standardization and amount of equal products produced. Standardization, that is on the y-axis, is taken as a measure because it gives a clear distinction between shipbuilding and the other industries. Standardization is essential for implementing re-use of parts/assemblies which is an important function of PDM and also a function which can be implemented on a system level.

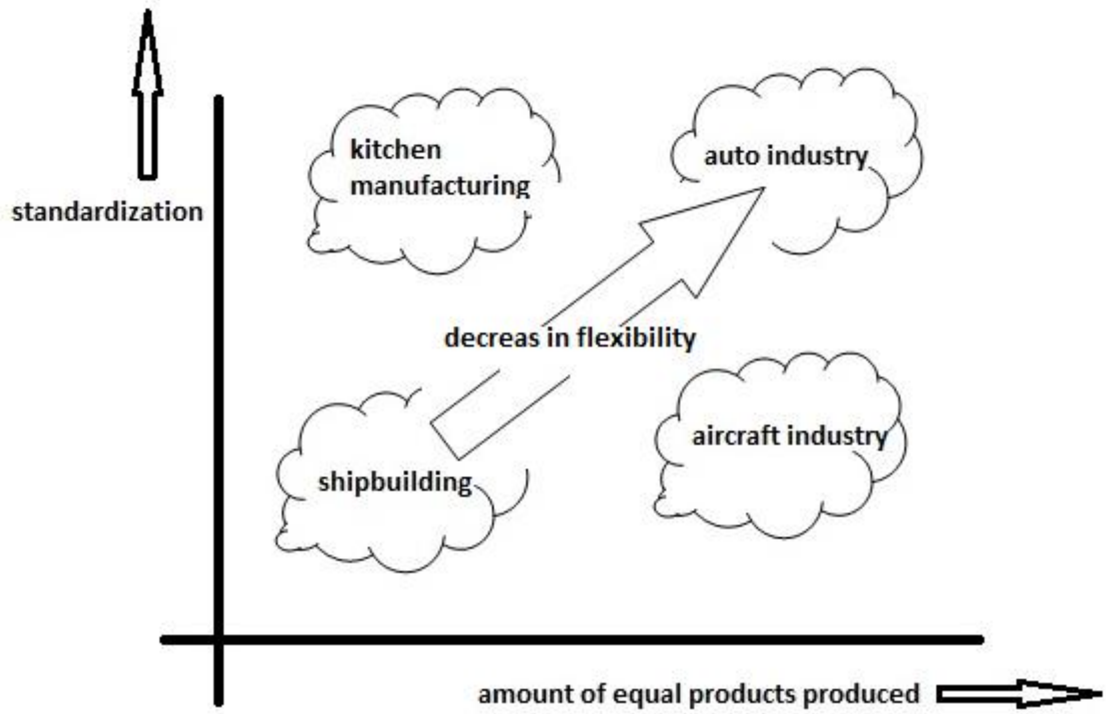



Figure 38, "flexible production universe"

The point of Figure 38 is to show what happens when either the amount of equal products produced, standardization, or both increases. The flexibility of the production of products is decreased. This may result in a shipbuilder not being able to comply with its customers' demands as the shipbuilding industry is doing at this moment. Looking at Figure 38 it becomes clear that when the production process contains more standardization and more of the same products are produced, the implementation of PDM might be easier looking at the other industries that have implemented PDM to a greater extent. By introducing standardization or producing more of the same products, the position of the shipbuilding industry in the "product manufacturing universe" will change.

# 10 NUPAS-CADMATIC feasibility

The solution present in this research for the use of PDM is not based on a certain CAD/CAM application or PDM software or even some kind of CAD/CAM-PDM integrated combination. The solution presented here is based on general PDM theory and its applicability in the shipbuilding industry. Because a single theoretical CAD/CAM-PDM integration solution cannot have much practical value, the feasibility of this solution will be tested on the NUPAS-CADMATIC hull and outfitting packages. These packages are commonly used in the shipbuilding industry and feel the need to explore the possibilities of PDM. These packages form a collaboration to supply CAD/CAM software for the modelling of hull construction (NUPAS) and outfitting systems (CADMATIC) of a ship. Figure 39 shows a very global overview of how the two systems collaborate in the current situation. The three components in Figure 39 are the two applications NUPAS and CADMATIC both of these application have some kind of connection to the database Common Object Storage (COS). It can be seen that there is a double arrow between NUPAS and COS and a line between CADMATIC and COS. The double arrow represents a coupling and the line an active connection. This coupling means that there is no active link but there needs to be a manual transfer of a modelled hull construction piece from NUPAS to the COS database. When this transfer is done the outfitting application CADMATIC can use this workplace to model outfitting objects into the in NUPAS modelled workspace. Of course, the situation described in Figure 39 is the most simple description of the system. In the next section the applications, database and their collaborative model will be explained.

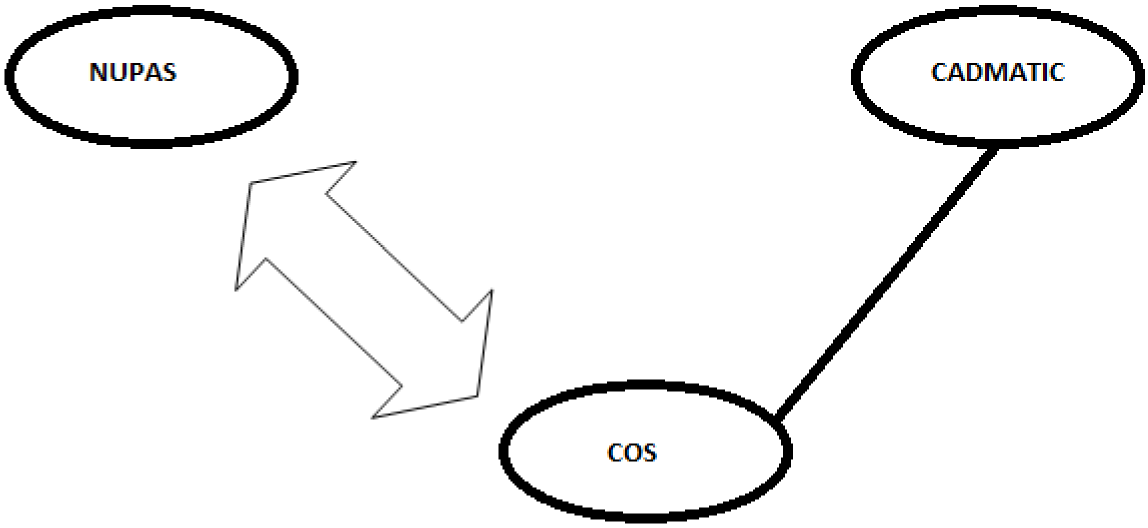



Figure 39, NUPAS/CADMATIC collaboration situation

## 10.1 NUPAS-CADMATIC hull

The hull construction application is NUPAS-CADMATIC hull, the software developed at NCG. With this application, the basic and detailed design of the hull construction can be done. For a typical shipbuilding project, the creation of a hull construction starts with a hull shape which can be imported using the Hull Shape Import Manager as shown in Figure 40.

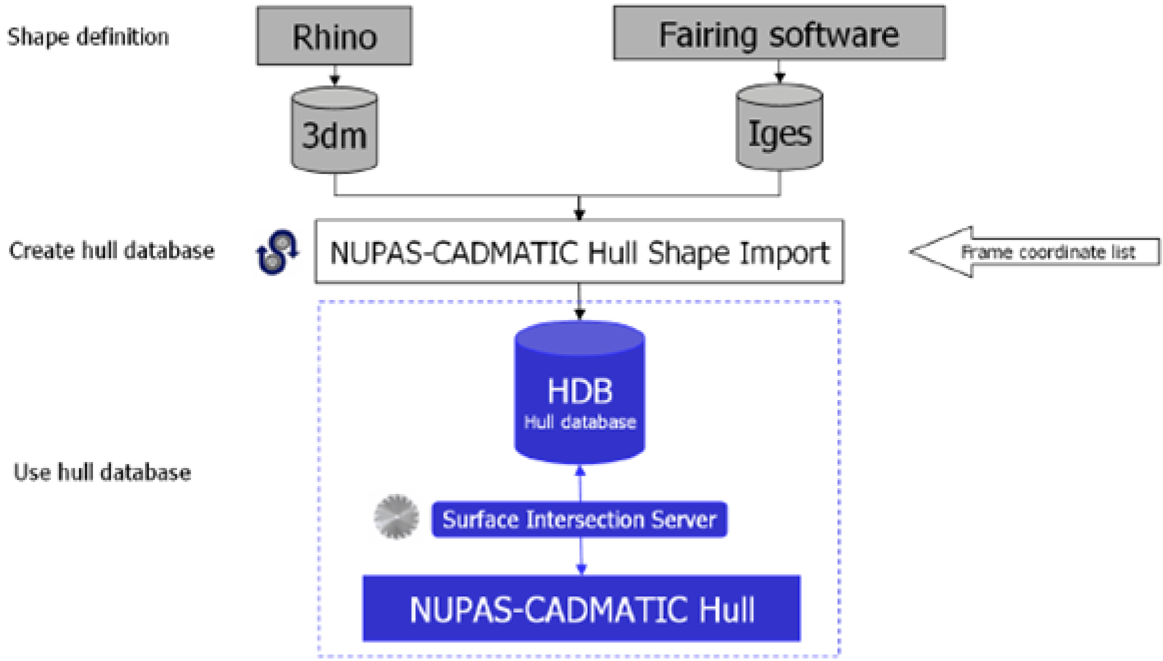



**Figure 40, Hull Shape Import Manager (The formats 3dm and iges are just example, a single file in another format could also suffice)**

The file formats that can be imported are:

- IGES (.igs, *.iges)
- Rhino 3dm (up to version 4) (*.3dm)
- NAPA (*.db)
- ACIS (*.sat)
- AutoCAD (*.dxf) (polygons only)
  extra information such as group numbers, curve types, and curve numbers can be imported from IGES, Rhino, and NAPA files.

The imported hull shape is subsequently filled with hull construction. The 3D hull engineering is the core of the NUPAS-CADMATIC hull system. The application contains of a number of sub-modules to provide a solution as complete as possible for the modelling of the hull construction of ships. The sub-modules consist of 2D drafting functions, 3D modelling functions, numbering, list generation, shell views, lay out views, nesting, and seams and butts functions. These functions can be considered as basic modelling functions of CAD software.

The NUPAS-CADMATIC hull system uses a high level intelligent built-in structural topology. The meaning of topology is: *Topology is the mathematical study of the properties that are preserved through deformations, twistings, and stretchings of objects. Tearing, however, is not allowed.* (Topology, 2015)

For the NUPAS-CADMATIC hull system, it means that the modelled construction elements e.g. stiffeners, plates, bulkheads etc. are related in such a way that a dependency is created between the construction items. All these related items and their dependencies make sure that when a change is applied to a construction item, it will be possible to recalculate all other depending construction items automatically. This functionality is of great value for the NC hull application because a complete construction can easily be recalculated when a change is made to a construction detail, but it may pose some problems when applying the PDM architecture because of the high level of dependency of the construction elements in the complete construction.

Within the NUPAS-CADMATIC hull system, the sub-modules make use of so-called type files. These files are used to define profile end types, cut-outs, and (corner) holes. The type file data which makes up the norms on which the construction is modelled is of great importance to the construction model of a project because it configures the rules for the production data. Figure 41 gives a global overview of how the type file data is entered. The hull database provides the parts that need to be in the hull construction. NUPAS-CADMATIC hull uses the information contained in the type file and forms the construction elements. The application uses topology to manage the relations of the parts and its intrinsic intelligence is able to recalculate the depending construction parts when changes have been made. The output of the hull program is construction drawings and production data.

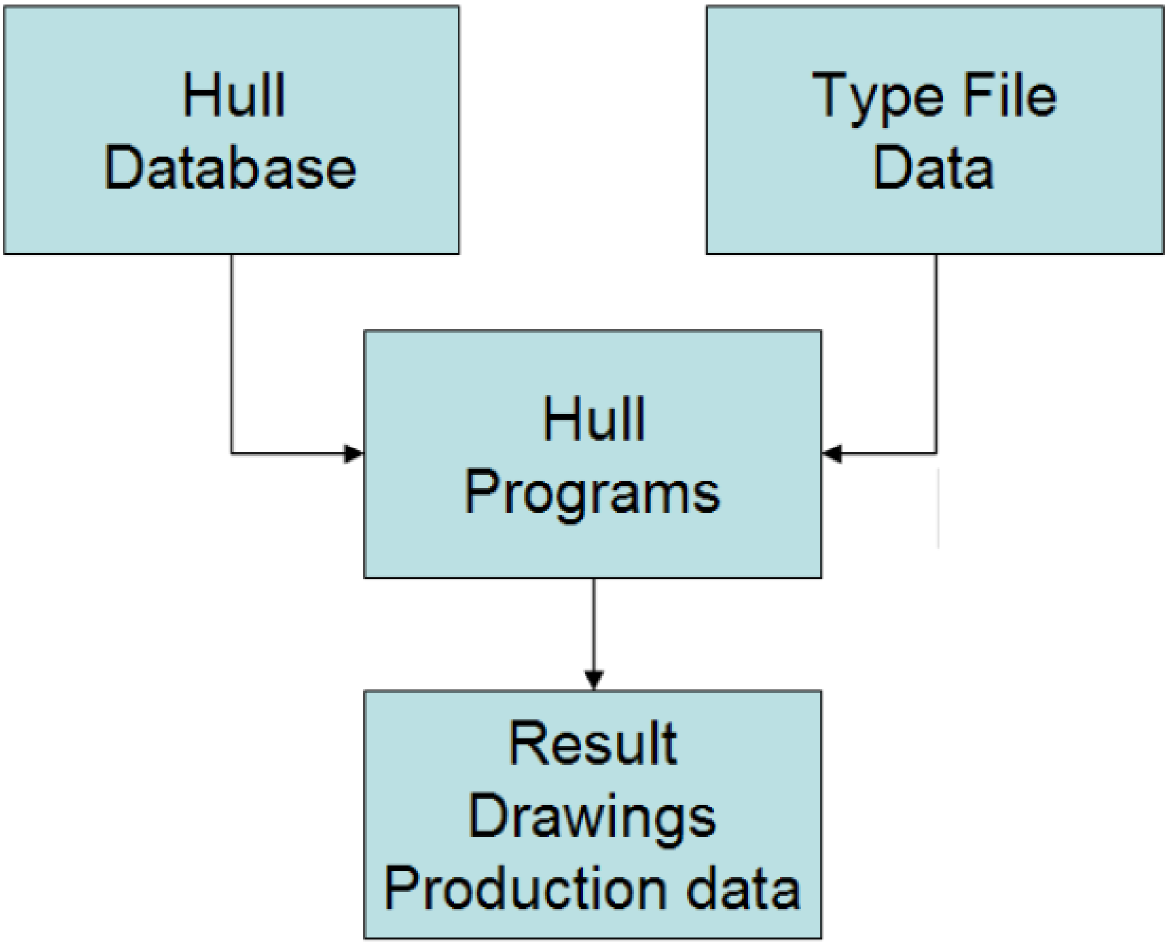



Figure 41, use of type files

The NUPAS-CADMATIC hull application stores its project data in a local workspace accessible to workstation that are on the same network as the project data is stored. To exchange hull data between offices or companies the Hull data exchange (Hdx) is used. Figure 42 shows how the environment using Hdx looks like.

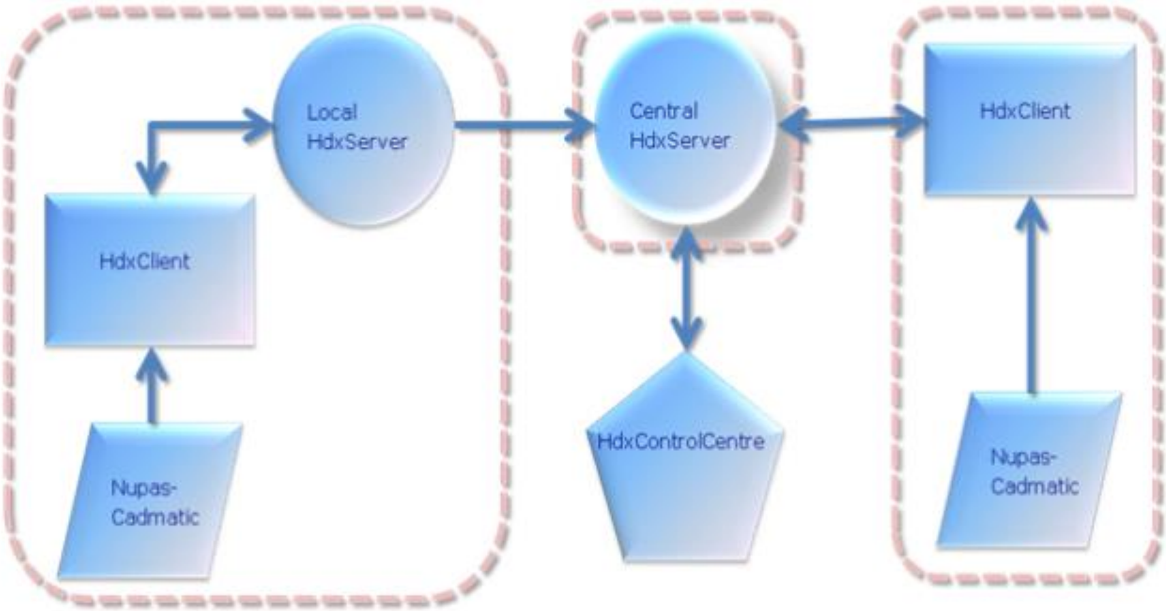



Figure 42, Hdx building blocks

HdxSystem is a software layer which runs alongside Nupas-Cadmatic Hull (NC Hull). NC Hull interacts with HdxClient, which runs on the same workstation as NC itself. In turn, HdxClient interacts with HdxServer. HdxServers are responsible for managing the project data, and the HdxServer program is generally placed on the same machine which stores the data. The HdxControlCentre is the user interface to the HdxServer and can operate from anywhere. Functions of the symbols of Figure 42 are:

**HdxServer**

- Packs, sends, receives, and unpacks project data.
- Verifies permissions for requests from HdxClient and other HdxServers.
- Monitors and notifies the Hdx environment for changes.
- Nupas-Cadmatic data is stored in the file system exactly as before.

Each HdxServer can be used in one of two configurations: Local HdxServer (Optional), and Central HdxServer (Required).

**HdxControlCentre**

- Forms the user interface to HdxServers.
- Has interfaces for project management, user management, authorization, and monitoring.
- Used to configure the communication parameters with HdxServers at other locations.

**HdxClient**

- Forms the connection between Nupas-Cadmatic and the HdxServer.
- Regulates the behavior of Nupas-Cadmatic on the basis of the project.
- permissions information provided by the HdxServer.
- Provides continued authorization services if HdxServer is temporarily unavailable due to network outage.

The Hdx is developed to make it possible to work in a distributed project environments where data exchange between participants in such an environment is necessary. A schematic example of such an environment is shown in Figure 43.

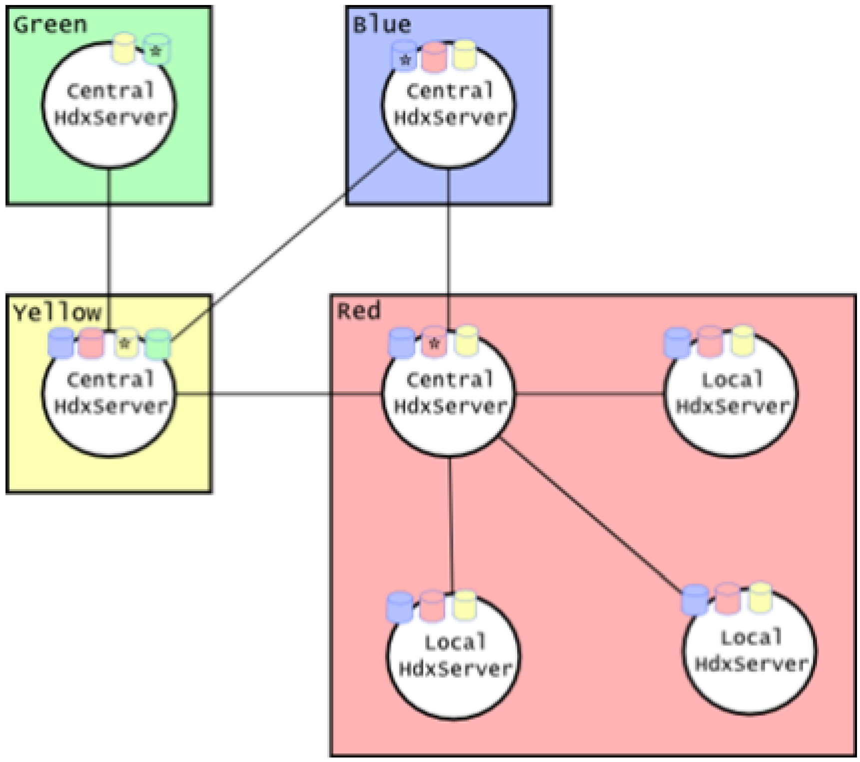



Figure 43, Distributed Project Environment

There are four participants in our scenario – participants Green, Blue, Yellow, and Red. Each participant owns a bunch of projects, stored on their hard drives – participant Green has green projects, participant Yellow has yellow projects, and so on. These are the little cylinder icons with *-symbol in the diagram.

Participant Green has linked with participant Yellow and exchanged some project data. The little green cylinder in participant Yellow indicates this. Yellow is also sharing some project data with Green, as indicated by the yellow cylinder in participant Green.

The diagram indicates that in this scenario, Yellow has linked with the other three participants, Green is linked only with Yellow, Blue is linked with Yellow and Red, and Red is linked with both Blue and Yellow. They have exchanged all projects as far as their links permit. The network connections needed for collaboration and their consequences are as follows:

- A participant always retains ownership (and thus control) over their own projects.
- A project can only be shared with other participants if you are the owner.
- Local HdxServers hold copies of projects which are shared with the participant via the Central HdxServer.
- Local HdxServers are not project owners.
- HdxSystem project sharing philosophy is known as a “star configuration”
    - The centre of the star is a project owner (such as Red).

- The points of the star are the possible destinations (Yellow and Blue).
- Green is not a possible destination because it is not directly linked to Red.

The above explained way of working is the current situation where project environments are distributed. As depicted in Figure 39 the future situation will be that the project distribution will go through the COS server and uses a Hull to COS Agent (HCA) instead of Hdx. Figure 44 shows the future situation which will execute the project distribution. By delegating the project distribution to COS, the typical COS server replication structure as described in the Chapter 11.2.1 applies. The HCA converts the models created by NC hull to a format that is compatible with the storage method of the COS database. As shown in Figure 44 the main and every satellite location needs a HCA to do the data conversion. The HCA functions as hub between the NC hull application and the COS database for every project at a certain location.

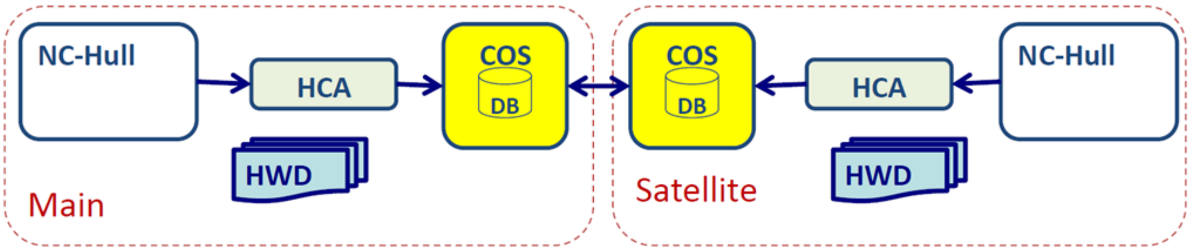



Figure 44, new hull project distribution

## 10.2 NUPAS-CADMATIC outfitting and COS server

Together with hull construction a ship consists of outfitting objects. Just as the hull construction, NC outfitting does the design of the outfitting objects needed to be modelled. Both the NC hull and NC outfitting make use of a local workspace but the shared database that the packages use to create a complete model of a ship is the COS Database as can be seen from Figure 39. The NUPAS-CADMATIC outfitting system can be used to model all the outfitting components in a ship. The complete outfitting describes the parts and components of the systems and interior of the ship. This interior contains everything that can be thought of when designing/engineering a ship: system outfitting, such as big and small components, piping, ducting, cable trays, small outfitting (e.g. ladders, hatches, railings, etc.) Interior outfitting (e.g. tables, chairs, etc.)

Together with the NUPAS-CADMATIC outfitting system, the COS server manages the libraries that the projects use. These libraries that are used by the outfitting system contain the 3D objects which are used to model the outfitting in the ship. As can be seen in Figure 39, the connection of the NC outfitting system with the COS is active and thus the connection with the libraries is active. The data of the 3D objects is merged from two levels, the library and the 3D model object. In the library the object properties such as components and materials are defined. The library also contains dimensions needed to make the geometric 3D model of the component and of course material information: description, manufacturer, type, size , mass, order number, etc. The 3D model object, which is the instance of library component in the global ship model, contains a reference to a library component and additional individual attributes (e.g. a unique position id). This 3D model object visualizes the 3D object using the referenced component model and the location and orientation in the global 3D model. Other than the information that refers to the actual objects the project contains settings and rules: different kind of configuration (e.g. pipe connections and face definitions) and different kinds of design rule definitions, such as compatibility rules of objects to be connected in the model, piping specifications like lists of allowed parts for piping depending on purpose and environment.

### 10.2.1 Architecture

As described in Figure 39, a collaboration between NC hull, NC outfitting, and COS is a desired way of working in the future where also the connection between NC hull and COS is active. NC outfitting has already an active collaboration with COS, this architecture is shown in Figure 45. The architecture pictures in Figure 45 consists of two levels: the application server containing COS server projects and library and the workstation containing the NC outfitting system (plant modeller). The figure displays a distributed working environment containing two projects. The application is connected to the COS server which manages the connections to the library and project databases. The use of the 3D objects that are stored in the library is done through referencing. The 3D objects are not copied to the project data base but they remain in the library. This is also the reason why project A is distributed together with the library as shown in Figure 45. The library needs to be replicated on a distributed workplace for the referencing structure to work. This also means that the project data base needs to be replicated. Within the project database the outfitting systems made up by the 3D objects from the library are formed. This also contains the connections of pipes, ducting, cable trays, between the components and each other.

The collaboration between NC hull and outfitting is done via the coupling between the hull system and Project database through the COS server. A piece of hull construction model that forms the modelling environment for the outfitting system is transferred to a project database so that the NC outfitting application can use the hull construction and outfitting can be modelled in it. When by modelling outfitting changes to the hull construction are necessary a work request mechanism is started. Common changes to the hull construction created by outfitting are holes or pipe penetrations. This work request mechanism will make sure that a preliminary hole is created in the hull construction.

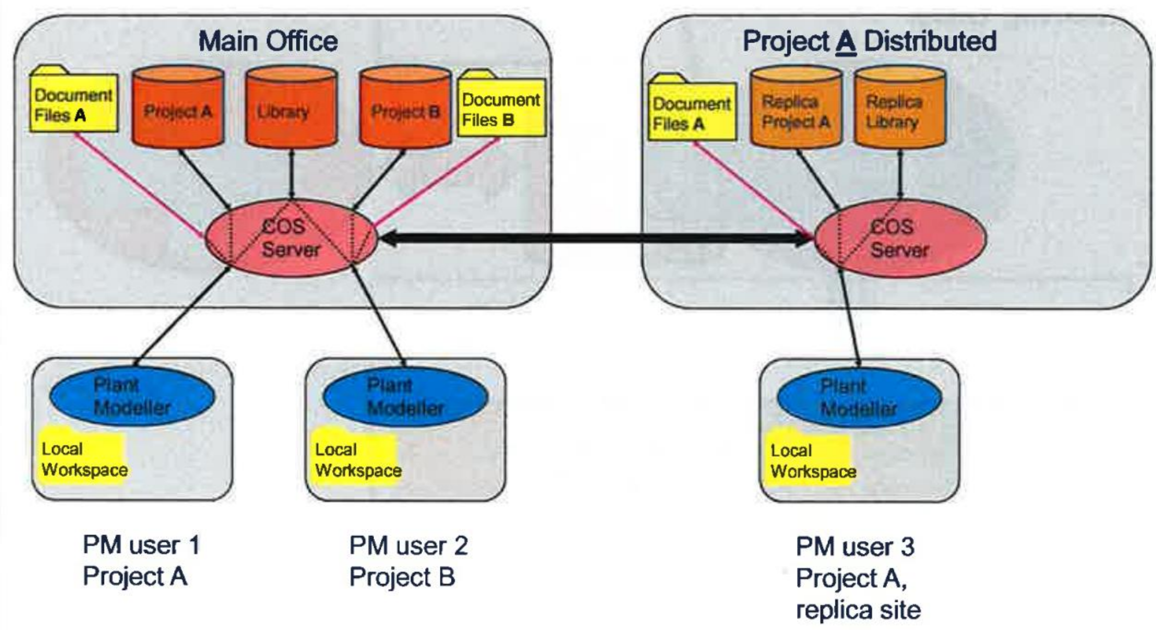



Figure 45, applications vs. COS

## 10.3 Collaboration of Hull and Outfitting

The above described applications both have their own functions and way of working. Nevertheless a ship consists of both a hull construction and outfitting thus the applications have to work together. Figure 46 shows a schematic representation of the hull (NC-H) and outfitting (NC-O) collaboration. The collaboration is handled by the Hull Agent (HA). When the hull application has modelled and finished a construction in the Hull database (Hdb) workspace, the outfitting application needs to model the outfitting components in the model. The HA converts the model and links it to COS such that it can be used by the outfitting application. When the outfitting application makes a hole in the hull construction the HA converts this outfitting hole to a contour of a hole that is visible in the hull application. During this collaboration the data of the hull construction remains in the Hdb.

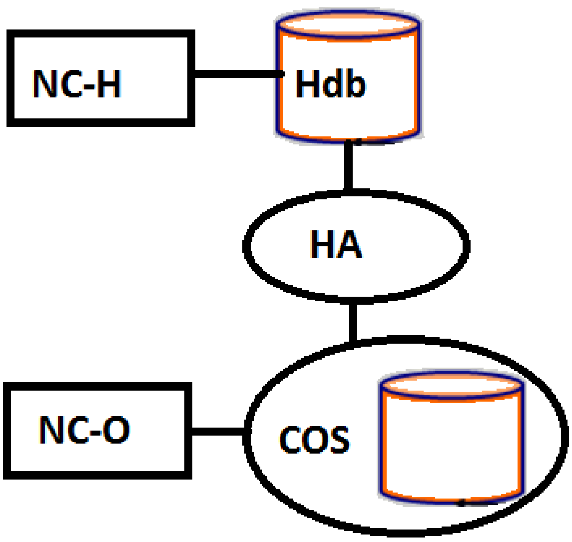



Figure 46, collaboration through Hull Agent

## 10.4 System feasibility

The devised PDM solution is based on a generic shipbuilding engineering system. This generic engineering process that is described in Chapter 6 does not have the origin of an existing engineering process at a shipyard somewhere in the world. The described engineering process is developed from transforming ship manufacturing flowcharts to tasks that an engineer has to execute for that process to be possible. Moreover, basic knowledge is used that assumes that if a ship needs a part it means that someone needs to engineer it. Thus by identifying parts of the ship the tasks of the engineering are also identified. Thus when a PDM solution is based on a generic engineering process the solution will also be generic. The advantage of this generic solution is that limitations of an existing process have not influenced the solution. This advantage may very well be a disadvantage because it may be that the solution is infeasible for some processes in combination with specific software solutions.

This research is conducted in collaboration with Numeriek Centrum Groningen (NCG), which provides hull construction software, and Cadmatic, which provides outfitting software (CAD/CAM software). For this feasibility study the PDM solution is to be tested on those two shipbuilding applications. As provider of shipbuilding CAD/CAM software, NCG has multiple customers each with its own engineering process. The use of PDM within an engineering process is not only the result of the used

CAD/CAM software but also the configuration management of the engineering process, thus a complete PDM solution is unique for every engineering process. Within the system feasibility section, first the feasibility is discussed, after that recommendations to the NUPAS-CADMATIC software and PDM solution have been done to provide feasibility.

### 10.4.1 Assemblies

The PDM solution is based on the possibility to build ships made up of assemblies. These assemblies are best described as 3D objects that are made up of single 3D parts and form building blocks of a ship. Assemblies have been explained in detail in Chapter 7.2.1. In order for the PDM solution to function at its best, a collection of assemblies should already be available. This requires a certain level of standardization to be able to have an effective re-use of assemblies.

- **NUPAS:** The use of assemblies does not fit well with the NC hull application. Within NC hull there are no 3D objects put together to form a structure. Within the hull construction NUPAS environment, the shell of a ship that contains stiffeners is constructed as follows: the shell of a ship's hull is created and forms a shape, subsequently the stiffeners are chosen from the library. The actual shape of that kind of stiffener is determined by relation the stiffener and the shell will have relative to each other. This topological way of working makes it difficult to have pre-defined pieces of hull construction, assemblies, that can be re-used because the geometry is dependent on the specific construction. Another difficulty for the use of assemblies is the norms set that is used when modelling a hull construction. If an assembly is modelled using a certain norms set it is not possible to use that assembly in a construction modelled under another norms set, which make re-use of assemblies with a certain norms set impossible in models with another norms set.
- **CADMATIC:** In NC outfitting, the possibility of defining and working with assemblies exists because the application makes use of 3D objects that are placed within a working environment. It should be possible to combine those 3D objects to assemblies. A problem can occur with ship specific properties. These properties specify for example connections between 3D objects (e.g. pipes that connect two pumps) or the connection of a 3D object to the hull construction model that is only applied in one certain ship. These properties are hard to be implemented in re-useable assemblies.

### 10.4.2 Data vaulting

The main use of the data vault is to have one central location for all the data. Furthermore the data should be protected against concurrency problems. Concurrency problems occur when multiple projects make use of the same data and changes are made simultaneously to the data at different locations. What should be the same data is now not the same anymore. Even though the applications in the projects have local workspaces, the part/object data does not leave the vault. A referencing system makes sure that the application within a project can use the latest version of the data of that a specific version can be used. The ship specific interrelations between parts/objects are stored in the local workspace.

By means of the data being "intelligent" (therefore the term "intelligent data") these interrelations within the model of a ship can be formed. The data vaulting is split into two vaults the intelligent data of the parts/objects is stored in the CAD vault and the assembly data as described in Chapter 7.2.1 is stored in the PDM vault. The storage of parts/objects in one central location should basically be the same as the use of library as is common for practically all existing CAD/CAM software. The difference is that there is the possibility for multiple applications to use that central data storage as library, which means that a CAD/CAM application cannot have its own dedicated library. In order for multiple CAD/CAM applications to use on central data storage they need to be able to either use each other's data or neutral format data needs to be used.

- **NUPAS:** the NUPAS hull construction uses a library that contains all the kinds of parts that can be created to form a hull construction. This library can be merged with the CAD vault but that library does not really contain the intelligent data. The parts in the library are uncreated shapes. When a shape, e.g. a bracket, is chosen the shape is created according to the surroundings of the modelling workspace. Moreover the modelling of a construction thus the creation of the shapes needs to be done in the local workspace of the application. Therefore it is very difficult for the NUPAS hull construction application to use pre-shaped parts through a reference way of working.
- **CADMATIC:** the CADMATIC outfitting application uses pre-shaped 3D objects which are already modelled through a referencing and uses ID numbers for the referencing so that the application using an object can find the intelligent data from the COS server which is the data storage used by the CADMATIC outfitting application. The centralized data storage that is part of the PDM solution does not fit together with how the data storage is done together with CADMATIC. The CADMATIC outfitting application makes use of replicas of a master library and project library which goes against the one location data storage of the PDM solution.
- **Combined NUPAS-CADMATIC use of COS:** Both the hull and outfitting application of NUPAS-CADMATIC have their own way of working concerning the data storage at this moment. Future plans are that both the applications will work with the COS server and from there make use of PDM like the configuration shown in Figure 39. The difference is that NUPAS will have an active connection with the COS server as well. An active connection means that the hull model is accessible in the COS server at any time during the modelling of the hull construction. This section will discuss the feasibility of the PDM solution together with the COS server when the hull and outfitting application use it.

  It has become clear that the PDM solution needs 3D parts/objects stored at a central location and are use through referencing. Where the CADMATIC outfitting application makes use of 3D objects and referencing, does the PDM solution together with the combined NUPAS-CADMATIC use of COS pose much problems for the NUPAS hull application because of the topological way of working. Especially when assemblies contain both hull and outfitting components which is the intention of the ideal PDM solution. A model that has topological references between all the parts in the model. When dividing a hull construction model in assemblies the topological references between the parts need to be cut, when cut the do not automatically arise when an assemblies is used.

  When the feasibility of the PDM solution together with NUPAS-CADMATIC without any changes to either toe PDM solution or the NUPAS-CADMATIC software is assessed, the NUPAS hull software together with the PDM solution is infeasible. The CADMATIC outfitting software together with the PDM solution is feasible. A combination as described in Figure 39 is because of the infeasibility of the NUPAS hull software also infeasible in this situation. In Chapter 11.6 will the recommendations to the applications and the PDM solution discussed

### 10.4.3 Interface functions

In the interface of the PDM solution are use cases that describe PDM functions to provide a smooth collaboration with the engineering system and the PDM solution. The PDM functions that are in the interface shown in Figure 47 are based on the use cases that make up the engineering process. These functions are in this research define in a way so that they fit the generic engineering process. Moreover, these functions are mostly managerial functions that are highly dependent on the specific engineering processes of a shipyard.

- **NUPAS:** the NUPAS hull construction application has a build in work breakdown (wbd) functionality. This wbd functionality makes it possible to make a breakdown structure of the hull in a free to choose arrangement. When this arrangement matches the use case distribution the NUPAS hull application together with its wbd functionality can work very well with the PDM functions. Only the PDM functions in the interface, the problems with the use of assemblies as described in Chapter 11.5.1 are not resolved by the wbd
- **CADMATIC:** a CADMATIC user works on outfitting systems or components. The use case distribution as described in the engineering system has also separate use case groups for different outfitting systems. This means that there is feasibility for the CADMATIC outfitting application to work together with the PDM functions. On the other hand there is no functionality within the application that prescribes a work package arrangement as in the engineering system.

As mentioned above, the functionality to support the PDM functions as they are described together with the engineering system is not necessary for the feasibility of the PDM functions as they are managerial functions and not necessarily system functions.

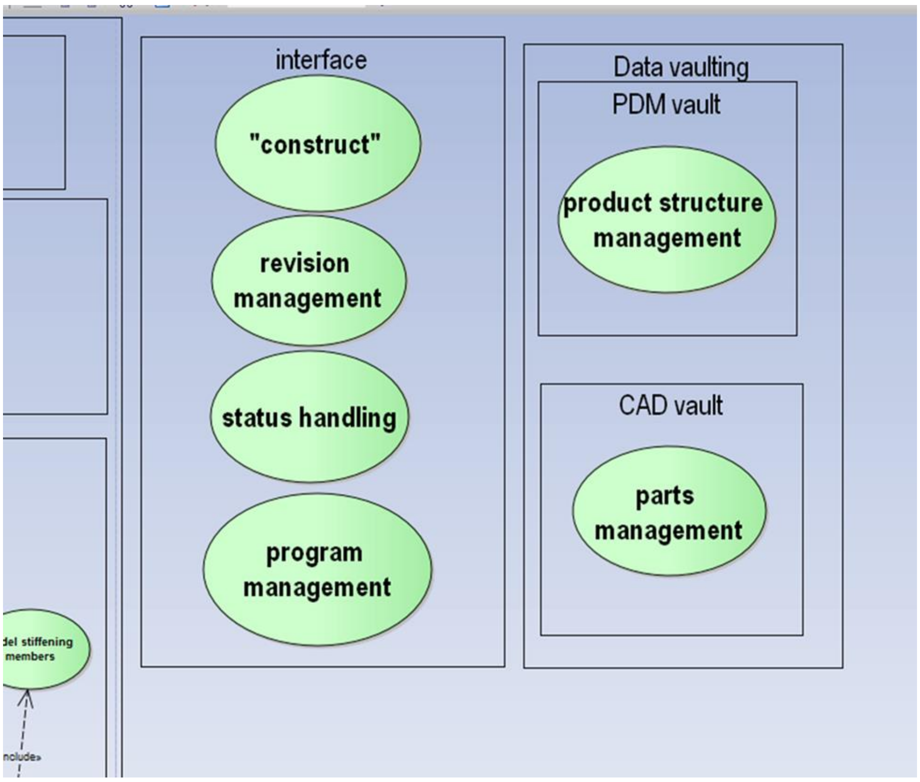



Figure 47, PDM functions

## 10.5 Recommendations

The conclusion of the system feasibility is that it is partly feasible because of the similarities of the CADMATIC outfitting software and the PDM solution. For the PDM solution to function properly for NUPAS-CADMATIC recommendations will be done to provide a fitting solution for the use of PDM together with NUPAS-CADMATIC hull and outfitting software. Not only will recommendations be done for the software but, the PDM solution will need alterations to provide feasibility. In the same arrangement as the system feasibility section will the recommendations be discussed.

### 10.5.1 Assemblies

As discussed before the use of assemblies plays a great role in this PDM solution. To extend the use of libraries that contain parts that are used to model constructions these parts/objects are combined to assemblies to improve the engineering of ships.

As discussed in the above feasibility section, it is difficult for NUPAS hull application to use pre-shaped pieces of hull construction in the modelling of a complete hull construction. The high level of topology used by the hull application goes against this use of assemblies. Also the norms set that is necessary when modelling a hull construction in the NUPAS hull application poses problems. The CADMATIC outfitting application is more suitable for working with assemblies because of the pre-shaped 3D objects that are used. Difficulties occur when assemblies are formed and ship specific properties are not contained in the assemblies.

- **NUPAS:** The difficulty of using assemblies should be overcome. By the use of work breakdown structure (wbd) functionality the assemblies should be formed. The wbd functionality which is meant to coincide with a building strategy and thus a possible use case arrangement. Depending on the requirements of the user it should be able to form assemblies in the form of hull blocks.
  When these assemblies are formed they have to regain the topology relations with the rest of the model. And the edges of the assembly which connect to the used model should be able to reconnect. Only the hull construction blocks that have been selected to form an assembly should be used in the PDM solution no other parts/objects of the NUPAS hull construction application. These stored assemblies will be very standardized but this does not mean that they cannot change when used in a project. Even though the PDM solution is very strict on changing assemblies.
- **CADMATIC:** the CADMATIC 3D objects are stored in a library and used by the application through referencing. In the same way the assemblies should be used. To solve the problem with the ship specific properties, the outer connections of the assembly should be able to interact with the connecting object. Either the assembly inherits the properties from the connecting object or the other way around. The outer connections of the assembly will most likely be pipes, ducting, or cabling and those objects will inherit or render properties. These properties should extended to the associated pipes, ducting, or/and cabling in the assembly. The use of assemblies should be more than just a combination of 3D objects. An outfitting system consisting of multiple 3D objects that is used in multiple ships should have the possibility to become a module. This modularization is described in Chapter 8.8.

When both applications are able to use assemblies the next step is to have assemblies that consist of both hull and outfitting components.

### 10.5.2 Data vaulting

The way of storing data differs much between the two applications and the PDM solution. At least for the CADMATIC outfitting application there are similarities in way of working which should benefit the feasibility of the data vaulting. The intended collaboration of data storage is meant to function in the COS server environment. The data vaulting done by NUPAS-CADMATIC hull application is done locally and not centralized. The different data vaulting methods are because of the different structure of data produced by the different software packages. These packages are working together to form a complete ship. As the data vaulting way is different the collaboration between the two packages should be optimal and this doesn't necessarily mean that the data vaulting methods need to be the same. Most important is that the functionality of the modelling has flawless collaboration between the two applications.

- **NUPAS:** The way of working where the project data is stored locally should not be changed. This way of working gives the NUPAS-CADMATIC hull application its strength which is being flexible because of the topology. But as discussed before the use of assemblies is essential for this PDM solution and it should be possible to re-use modelled blocks or pieces of a hull construction. By the use of the Hdx these formed assemblies should be shared to the PDM and CAD vault to make the use of assemblies possible.
- **CADMATIC:** the NUPAS-CADMATIC outfitting application uses a very similar data vaulting way of working there is the referencing which is in line with the way of working of the PDM solution. The library used in by the application should also be able to store assemblies together with the separate 3D objects. Furthermore the creation of database replicas should be limited. Especially, the project data bases should not be replicated because it increases the chances for concurrency errors, also the transfer of ownership, of objects can cause difficulties in an ongoing project. Copies of the main library is not really a problem provided that the library does not change.

### 10.5.3 Interface functions

In the feasibility of the PDM functions together with the NUPAS-CADMATIC hull and outfitting applications is described that the applications have a good feasibility for collaborating with the interface functions. Moreover, the functioning of those functions is very much dependant of how the engineering system is arranged. the feasibility of the collaboration between the PDM functions in the interface and the two NUPAS-CADMATIC applications is good, provided that the recommendations for the use of assemblies and data vaulting are executed.

## 10.6 NUPAS-CADMATIC hull and outfitting in PDM architecture

The two NUPAS-CADMATIC applications have their recommendations which are directed towards the use of assemblies, a central data vault, and collaboration between the two applications. The two NUPAS-CADMATIC applications need to fit in the PDM solution for which an architecture is described. The focus of the PDM solution is to use PDM by adding functionality to the data storage and application system by the introduction of assemblies and by introducing managerial functions, which should make the engineering process more efficient.

A managerial function that has not come to its right is change management. The revision management that is already in the interface already does some kind of change management, but it only make a choice which kind of assembly change is executed. When the engineering process work according to a use case breakdown structure or another work breakdown structure progress stages or milestones can be identified. A change, or configuration management which limits changes in the design which affect finished progress stages could help improve the engineering process. Again just as the other managerial PDM functions this is very dependent on the specific engineering process and has no need for a change in the application system or the engineering process. The architecture makes the managerial functions possible, but most importantly, it needs to make the use of assemblies possible.

Furthermore, the PDM solution and its architecture is designed to handle multiple projects but the focus is on the implementation of PDM and not necessarily on the distribution of projects over different locations which use the same centralized data storage. Figure 48 depicts the NUPAS-CADMATIC hull and outfitting components as they currently are within the architecture environment of the PDM solution. The architecture is depicted with only one project. The CAD vault, which should contain the intelligent data still contains the intelligent data but in separate databases just as in the situation as it is currently for NUPAS-CADMATIC. This basically is the same configuration as described in Figure 46. The interface does together with the managerial functions also the data distribution. This data distribution is done by the HA and COS this is because the managerial functions provide rules and requirements for the data distribution, but as explained before highly dependent on the specific engineering process.

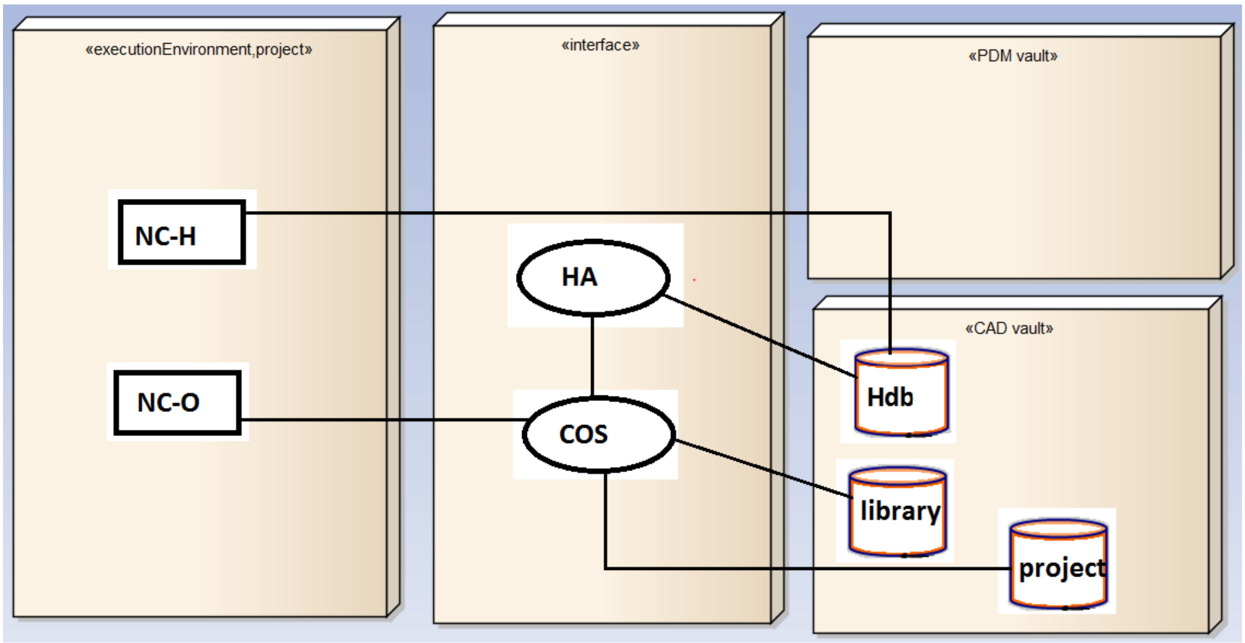



**Figure 48, current NC components in PDM architecture**

The next figure, Figure 49 is expanded with components to make the use of assemblies as described before possible. The added components are two data bases which are meant to carry out the task of storing the assemblies. First the Assembly-Hull database (A-Hdb) this data base is still in the CAD vault because the assemblies that consist of hull construction are not a combination of 3D objects they can be seen as finished pieces of hull construction with their own topological relations. The hull construction assemblies are intelligent data and therefore need to be stored in the CAD vault. The database that manages the assemblies of outfitting. The outfitting application makes use of 3D objects and the assembly library (A-library) is in the PDM vault because it only needs information that describes which 3D objects belong together to form an assembly.

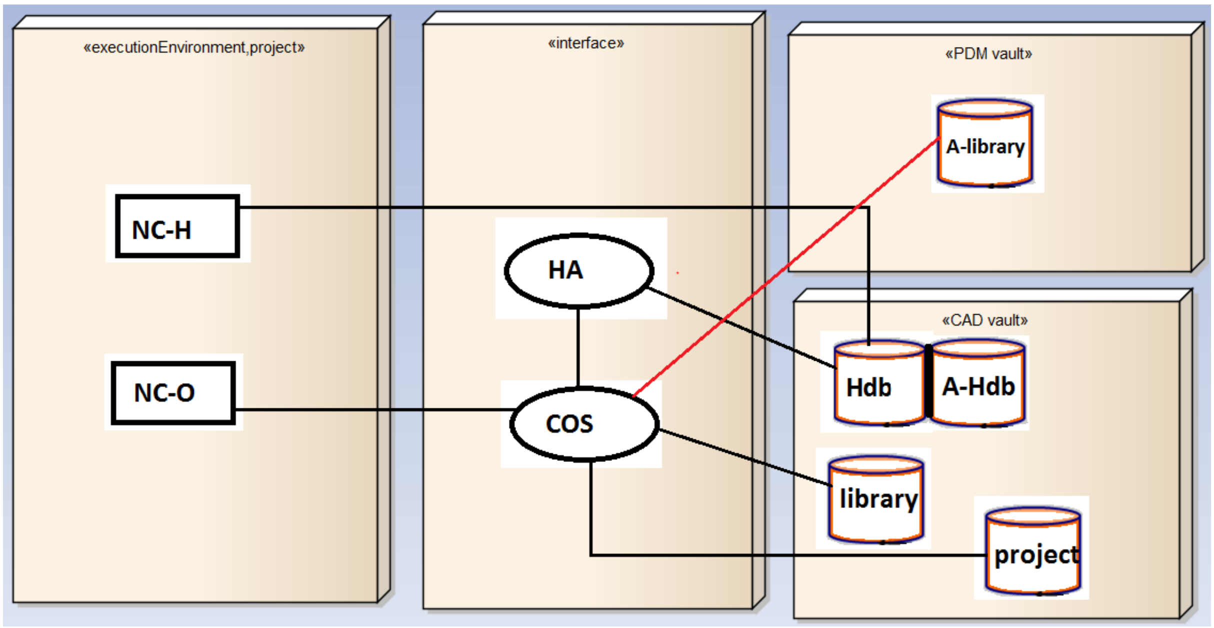



Figure 49, NUPAS-CADMATIC in architecture plus assembly databases

Figure 49 describes recommendations to NC hull and outfitting at the current situation. As described in Chapter 11.1, instead of the Hull Agent the HCA will take over and both the NC hull and outfitting data will be stored in the COS database. Figure 50 shows the NUPAS-CADMATIC situation when using the HCA. In this situation the hull construction models and outfitting model share databases and the hull construction assemblies can nog be stored in the PDM vault too.

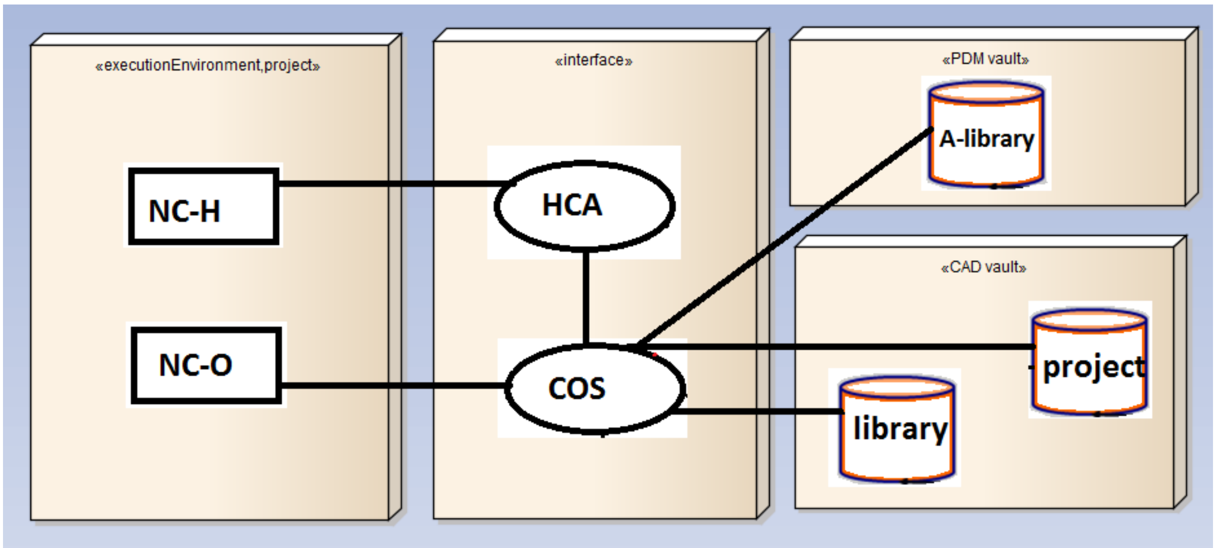



Figure 50, NUPAS-CADMATIC in the architecture making use of HCA

## 10.7 Future

The recommendations that are described do not lead to major changes in the NUPAS-CADMATIC packages. They can be seen as a first step. Moreover, the recommendations are by far not as far-reaching as the future plans of NCG. During the design of the PDM solution, it is presumed that the collaboration between the hull and outfitting application works smoothly, which is mostly the case but in practice there are still some problems occasionally. The future vision is that the two application become one integrated application, where the hull construction components are also 3D objects and the assemblies consist of hull and outfitting objects. This enables the use of one centralized databank. The recommendations are a start towards one integrated hull construction and outfitting application, which will be unique and very advantageous for the engineering of ships.

# 11 Conclusion and recommendations

This research will be concluded by answering the sub-research questions and finishing the conclusion with the answer to the research question. The research questions as stated in the introduction are:

What is the current situation of, the use of PDM together with CAD/CAM software? And how should a concept solution that meets the requirements of an optimal integration collaboration between PDM and CAD/CAM software take shape? The sub-questions are:

1. What is the current role and impact of PDM systems on ship design, engineering and production processes?
2. How are commonly used PDM systems in the shipbuilding industry suitable to use in combination with CAD/CAM systems and what are the needs and expectations of shipbuilders with respect to the connection to, or integration with, their existing CAD/CAM packages?
3. What are the requirements for an optimal integration/collaboration between CAD/CAM software and a PDM-system or strategy? And by what factors can a successful integration/collaboration be measured?
4. Design a concept solution that meets the requirements of an optimal integration/collaboration. Test the designed solution using NUPAS-CADMATIC as CAD/CAM system.

The role of PDM in shipbuilding can be best denominated as introducing, which means that almost every shipyard is doing something related to PDM, but there is no full integration. The products produced in the shipbuilding industry are mostly one of a kind products that are designed and engineered using multiple CAD/CAM applications. The use of multiple CAD/CAM applications and the production of one of a kind products are factors that hold back the integration of PDM. On the other hand, these one of a kind products consist of a great amount of similar parts/components. The introduction of PDM makes the re-use of engineering data possible, which can benefit the use of similar parts/components. The re-use of engineering data increases the degree of standardization in the engineering process. this standardization decreases the flexibility of the process as shown in Figure 38.

How and which PDM system should be used in combination with CAD/CAM applications depends very much on which CAD/CAM applications are used. Not only the used CAD/CAM applications determine how PDM or which existing PDM system should be used. The complete process and way of working on a shipyard determines the best use of PDM in combination with CAD/CAM applications. This may contradict some needs an expectation of shipbuilders who'd like to invest in a PDM system that provides a completely integrated PDM functionality.

After the research towards PDM and the existing PDM systems the engineering system was modelled. This resulted in a generic model of a shipbuilding engineering process. the model showed that actions **collect, construct,** and **save** are the connections of the engineering process with PDM. Connected to the engineering model are the interface and the data vault they represent a model of the PDM system developed in this research.

To fully integrated PDM, the five functions of which PDM is made up need to be executed. This requires adaption to the data storage system that works in collaboration with the CAD/CAM systems that are used in a certain engineering process. Not only does it need adaption to systems the way of working needs to be changed as well. For PDM to become beneficial the storage system needs to be able to handle assemblies and have secure centralized data storage. Furthermore if multiple CAD/CAM systems are used they need to be able to work together. The PDM functionality on a system level works best when the way of working on the engineering work floor also adapts to the use of PDM. Proper change and program management need to be executed. The use of PDM should increase the speed and quality of the engineering process. The concept solution accomplishes this by making use of standardized assemblies. This leads to a decreased time that engineers are modelling. The Use Case Breakdown Structure of two existing ships expanded from the use case model of the engineering process shows that most of the engineering time is spend on activities concerning the modelling of these assemblies. Which means that a significant part of the engineering process will benefit. Furthermore a centralized data vault with well managed access of users should reduce errors while executing concurrent engineering which improves the quality of the engineering process.

To measure the benefits of a concept solution that has incorporated the PDM functions, the engineering process needs to be structured in a way that the tasks of the engineers can be timed and the amount of errors counted. When this is the case, which is in the generic modelled engineering process, the performance indicators can be used to compare different engineering processes or different parts of the same engineering process.

The described PDM functionality is not completely feasible with the NUPAS-CADMATIC hull and outfitting applications. When the use of assemblies with 3D-objects is made possible a step in the right direction is made, further more every user of NUPAS-CADMATIC hull and outfitting needs to adjust its engineering process to make best use of PDM with the NUPAS-CADMATIC hull and outfitting applications. The optimal integration collaboration between PDM and CAD/CAM is very much dependent on the specific situation of an engineering process where the PDM functions are the same, but the execution of the functions depends on the specific situation.

The use of PDM has many benefits for the engineering process of shipbuilding and not only the engineering process, but possibly the complete shipbuilding process. Therefore the obvious recommendation that a shipyard should use PDM. How the PDM should be implements is a much less obvious matter. Looking at this research where the engineering process has been described before a PDM solution has been developed and connected, it is recommended that the engineering process in need of improvements should first be completely researched. When the process is complete mapped the choice between either an existing PDM system, newly developed PDM system, change in way of working, or a combination of these three can be made. Furthermore the implementation of PDM should be a step by step process.

# Bibliography


Chan, E., & K.M. Yu. (2007). A concurrency control model for PDM systems. *Computers in Industry*, 823-831.

CIMCO Integration. (n.d.). *white paper: CIMCO PDM.* Retrieved from http://www.cimco-software.ru/products/brochures/CIMCO_PDM-EN-web.pdf

*CIMCO MDM 7*. (n.d.). Retrieved from CIMCO Integration: http://www.cimco.com/product_pdm_description.php3

Coenen, J. M. (2008). *Controlling Engineering to Order Processes in Shipbuilding a Model Based Approach.* Delft, University of Technology.

D. Tony Liu, & X. William Xu. (2001). A review of web-based product data management systems. *Computers in Industry*, 12.

DSS SOLIDWORKS. (2010). *white paper: The intelligent PDM vault from DS SolidWorks.* Retrieved from http://www.solidworks.com/sw/docs/PDM_2010_ENG_FINAL.pdf

Eynard, B., Gallet, T., & Roucoules, L. (2006). PDM system implementation based on UML. *Mathematics and computers in simulation*, 13.

*Introduction To OMG's Unified Modelling Language (UML)*. (2015). Retrieved from OMG: http://www.omg.org/gettingstarted/what_is_uml.htm

J.L.G. Dietz. (2001). DEMO: Towards a discipline of organisation engineering. *European journal of operational research*, 13.

J.P. la Fontaine, M.G.R. Hoogeboom, & J.S. Könst. (2008). *Product Data Management.* Maj Engineering Publishing.

*KONFIG Technical Advantages*. (2015). Retrieved from Auto-trol Technology: http://www.auto-trol.com/technical-advantages

M. Welsh, J. Lynch, & B. Brun. (1992). A data model for integration of the precommissioning life-cycle stages of the shipbuilding product. *Journal of Ship Production*, 220-234.

Mallens, P., Jan, D., & Hommes, B.-J. (n.d.). *The value of business process modeling with DEMO prior to information systems modeling with UML.* Delft, University of Technology: Faculty of Information Technology and Systems, Department of Information Systems and Software Engineering.

Oh, Y., Han, S.-h., & Suh, H. (2001). Mapping product structures between CAD and PDM systems using UML. *Computer-Aided Design*, 9.

P.M. Veeke, H., A. Ottjes, J., & Lodewijks, G. (2008). *The Delft Systems Approach.* Delft: Springer.

*PDM / Product Data Management*. (2015). Retrieved from Siemens: http://www.plm.automation.siemens.com/nl_nl/plm/pdm.shtml

*PDM software or PLM software: what's the difference?* (n.d.). Retrieved from PDXpert: http://www.buyplm.com/plm-software/product-data-management-pdm-software.aspx

*PDMPlus Features*. (2015). Retrieved from cm·stat: http://cmstat.com/software/pdmplus-features/

PDXpert. (2013). *white paper: Discover 10 reasons why PDXpert PLM can ·Increase sales revenues ·Reduce costs ·Lower administrative expense.* Retrieved from http://www.buyplm.com/publications/PDXpert-PLM.pdf

*Product Data*. (2015). Retrieved from plmxpert: http://plmxpert.nl/nl/oplossingen/product-data

*Product Data Management (PDM)*. (2015). Retrieved from PTC Product & Service Advantage: http://www.ptc.com/product-lifecycle-management/product-data-management

*Product Data Management*. (2015). Retrieved from aras corp: http://www.aras.com/solutions/plm-solution.aspx?name=product-data-management

*Product Data Management Capabilities*. (2015). Retrieved from DSS SolidWorks: http://www.solidworks.com/sw/products/product-data-management/capabilities.htm

PTC Inc. (2011). *data sheet: Windchill PDMLink.* Retrieved from http://www.ptc.com/File%20Library/Product%20Families/Windchill/PDM%20Process/Windchill_PDMLink.pdf

PTC Inc. (2013). *data sheet: PTC Windchill PDM Essentials.* Retrieved from http://www.ptc.com/File%20Library/Product%20Families/Windchill/PDM%20Process/Windchill_PDM_Essentials.pdf

R.I. Whitfield, A.H.B. Duffy, J. Meehan, & Z. Wu. (2003). Ship Product Modeling. *Journal of Ship Production*, 230-245.

Rosemann, M., & Green, P. (2002). Developing a meta model for the Bunge-Wand-Weber ontological constructs. *Information Systems (is)*, 17.

Scheer, A.-W., & Nüttgens, M. (n.d.). *ARIS Architecture and Reference Models for Business Process Management.* Saarbrücken: Institut für Wirtschaftsinformatik, Universität des Saarlandes.

Siemens Industry Software. (2014). *white paper: Getting started with product data management.* Retrieved from http://www.plm.automation.siemens.com/nl_nl/Images/18955_tcm703-86807.pdf

Song, I.-Y., & Froehlich, K. (1995). Entity-Relationship modeling. *IEEE Potentials*, 6.

*The eCl@ss standard*. (n.d.). Retrieved from eCl@ss: http://www.eclass.de/eclasscontent/standard/overview.html.en

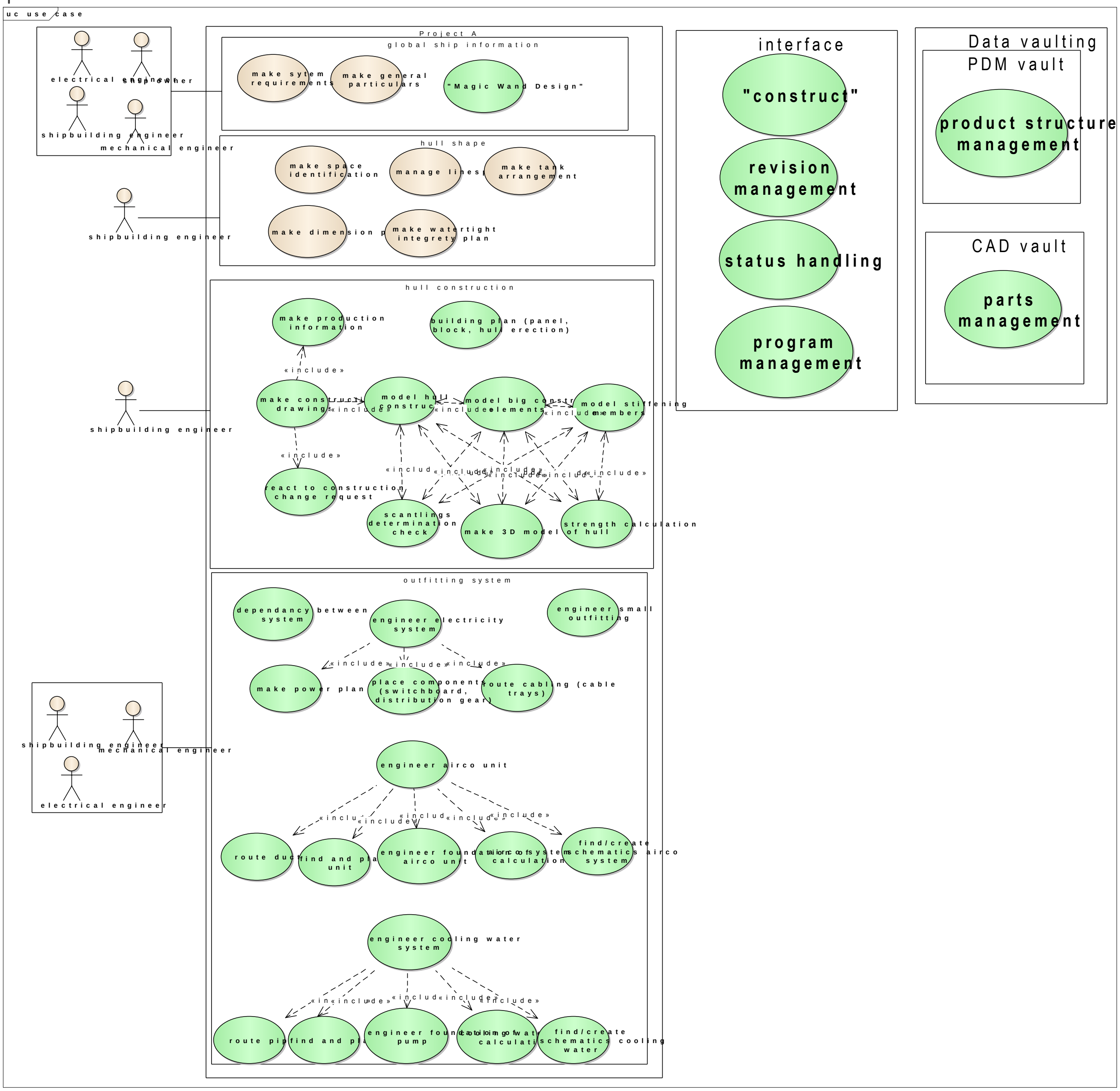
*Topology*. (2015). Retrieved from Wolfram MathWorld: http://mathworld.wolfram.com/Topology.html

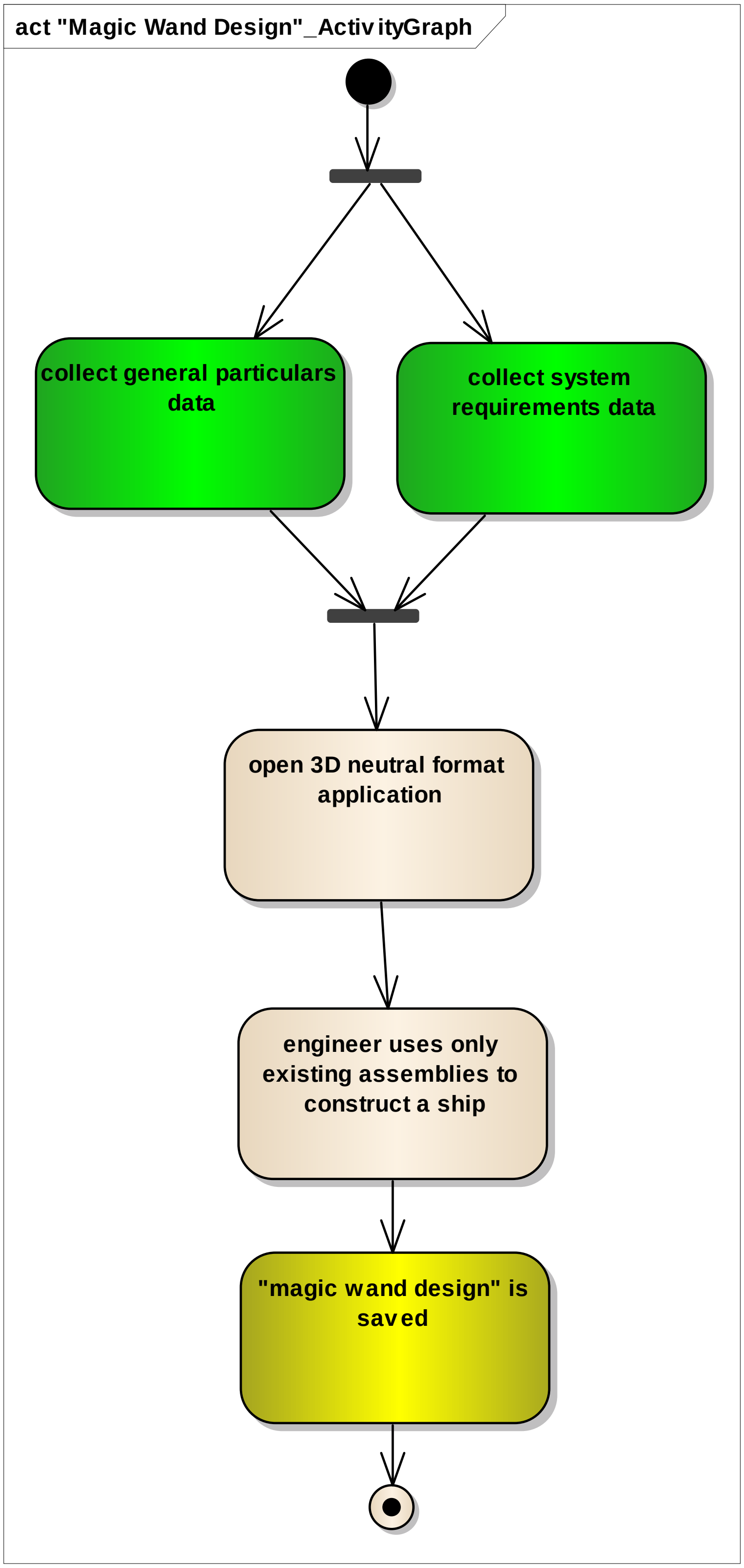
W. Ambler, S. (2005). *The Elements of UML 2.0 Style.* Cambridge University Press.

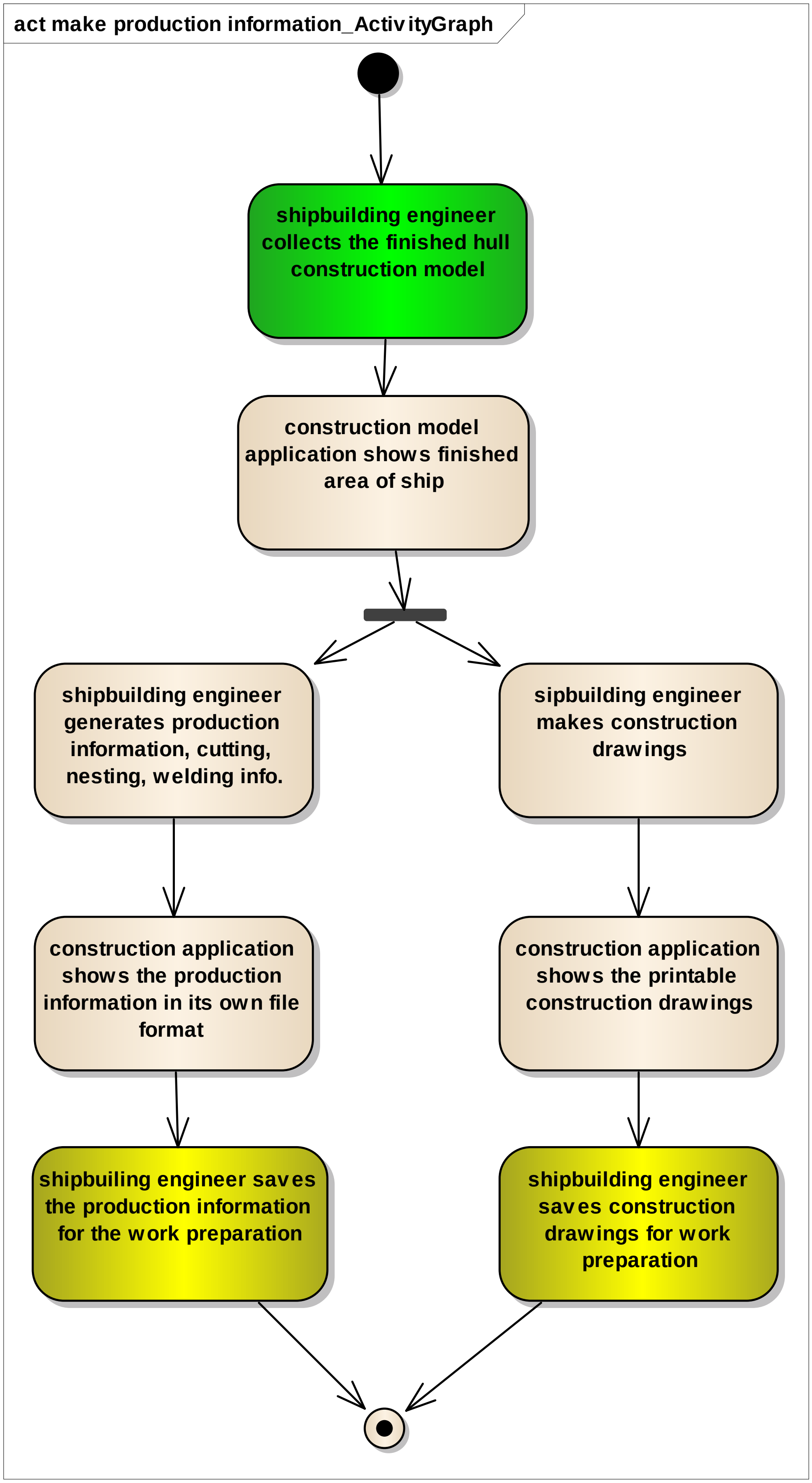
*What is UML?* (2015). Retrieved from SPARX systems: http://www.sparxsystems.com.au/resources/uml2_tutorial/

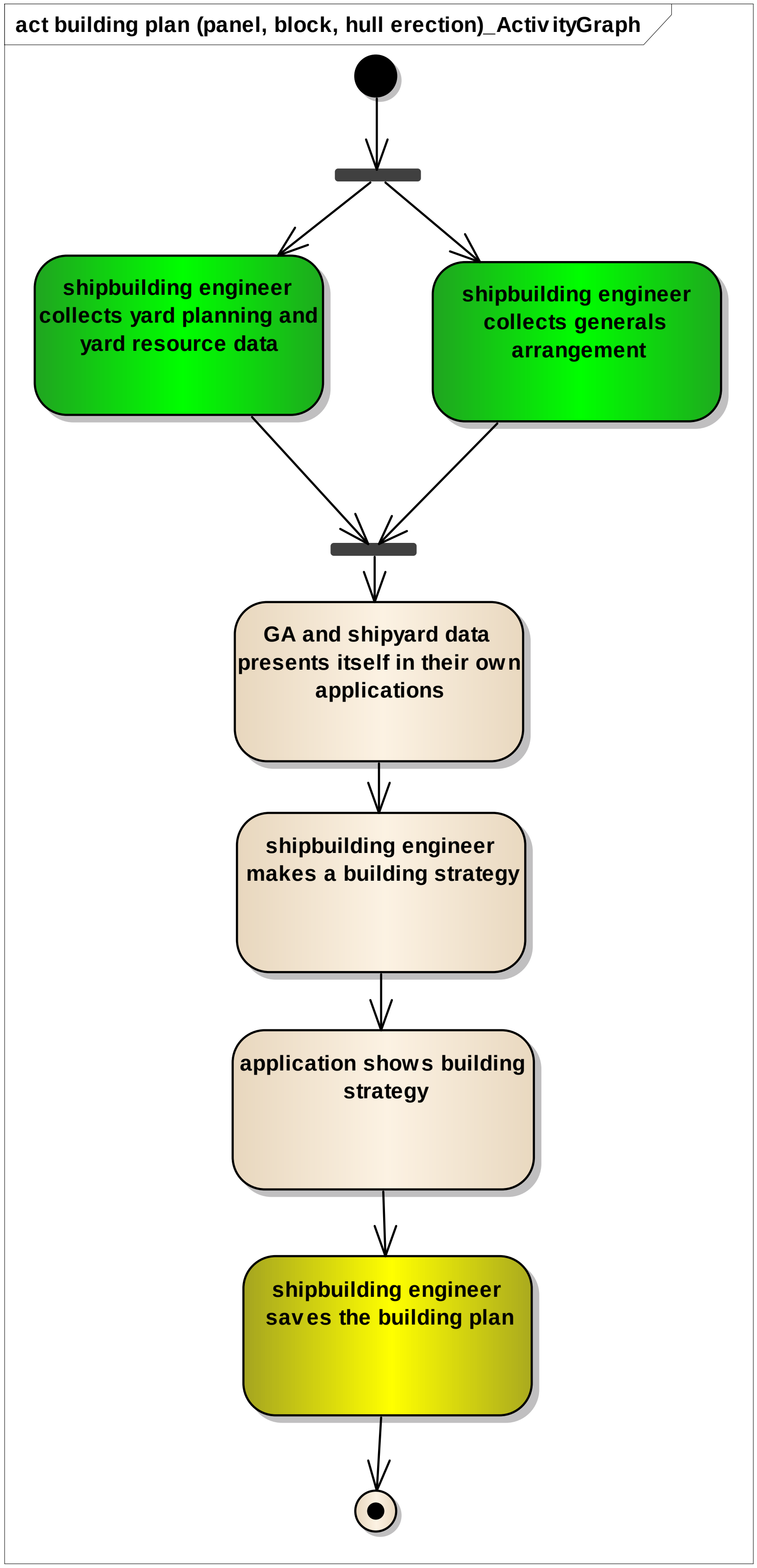
# Appendix A

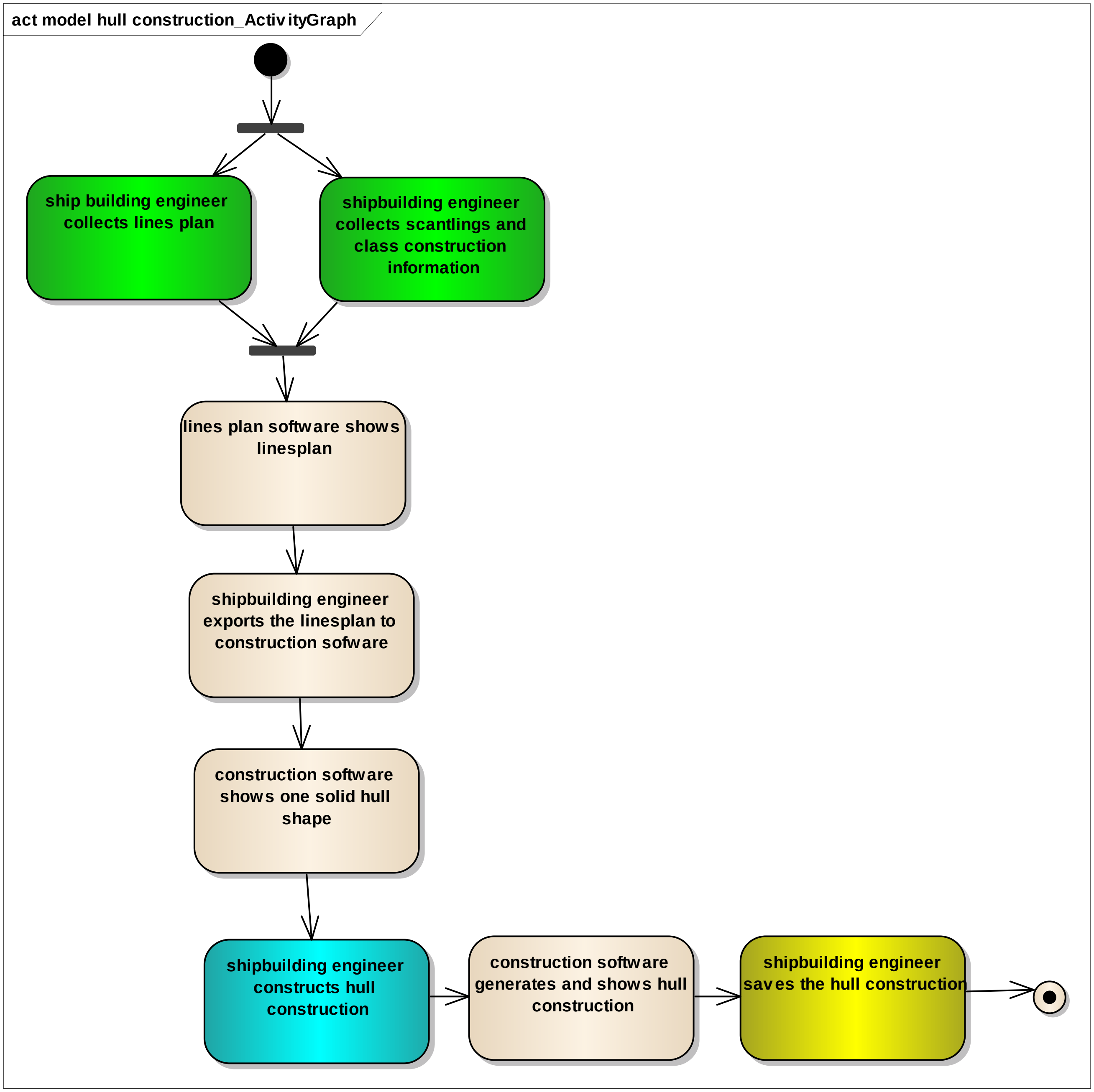
## Top level use case model

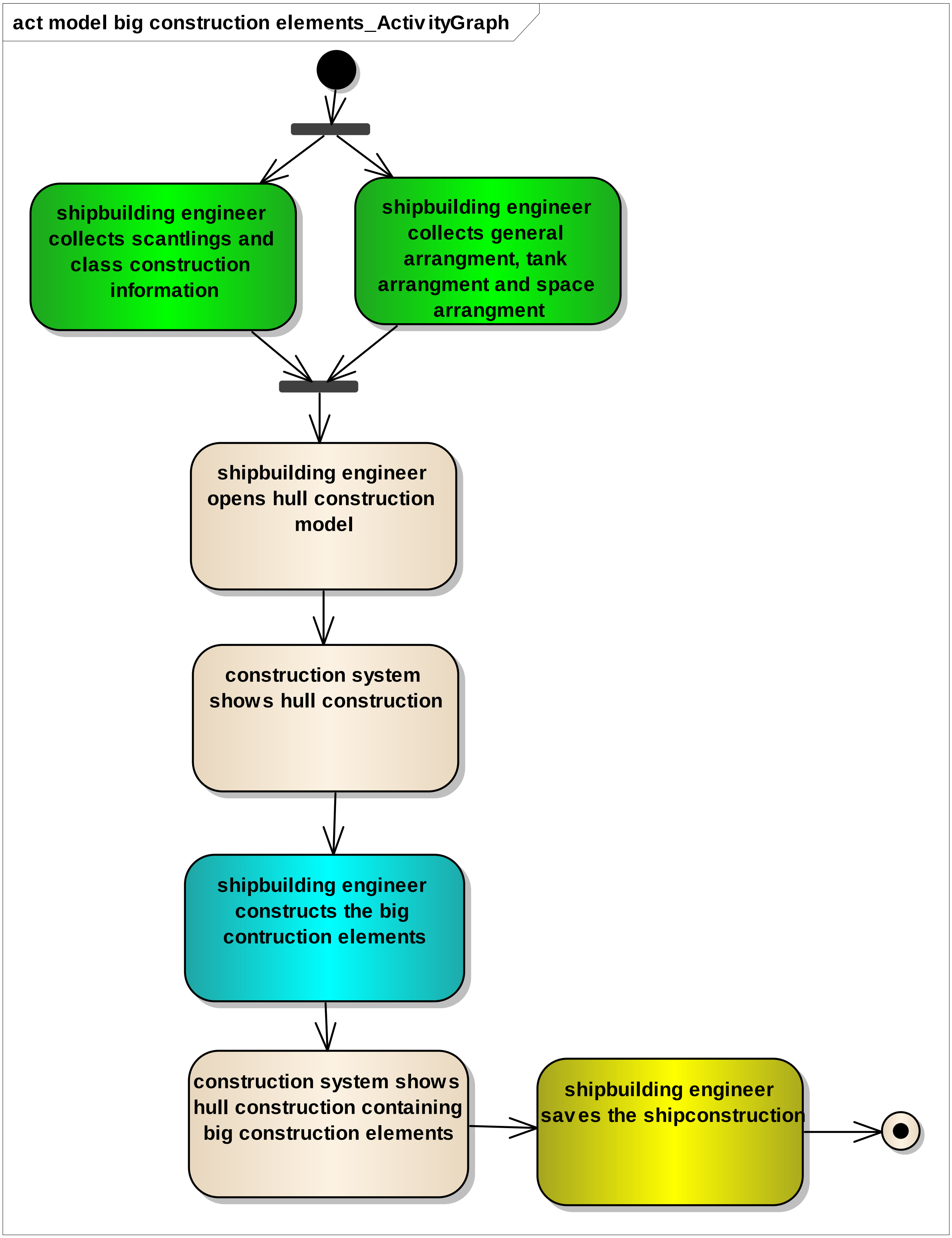


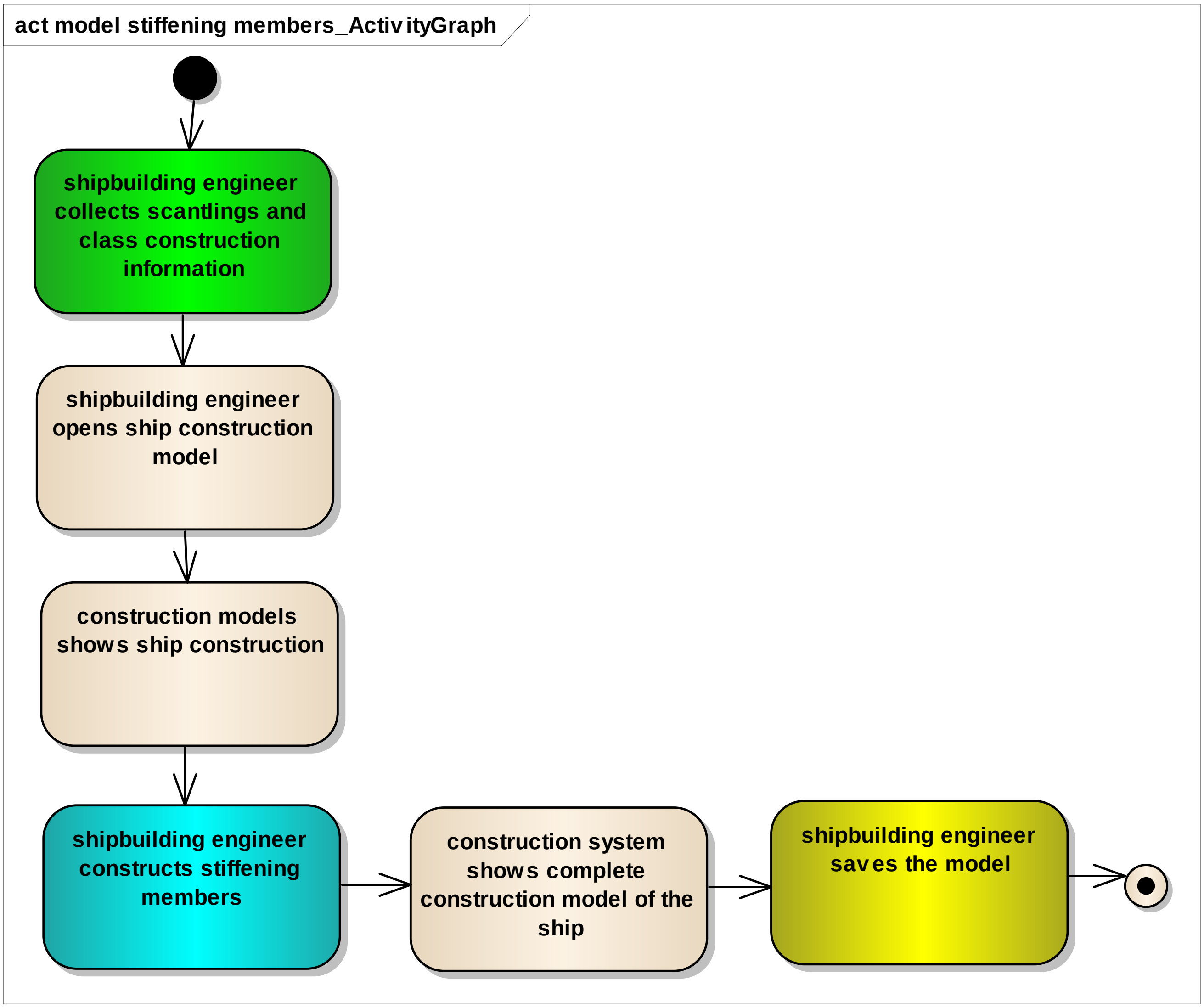

act "Magic Wand Design"_ActivityGraph
collect general particulars data
collect system requirements data
open 3D neutral format application
engineer uses only existing assemblies to construct a ship
"magic wand design" is saved

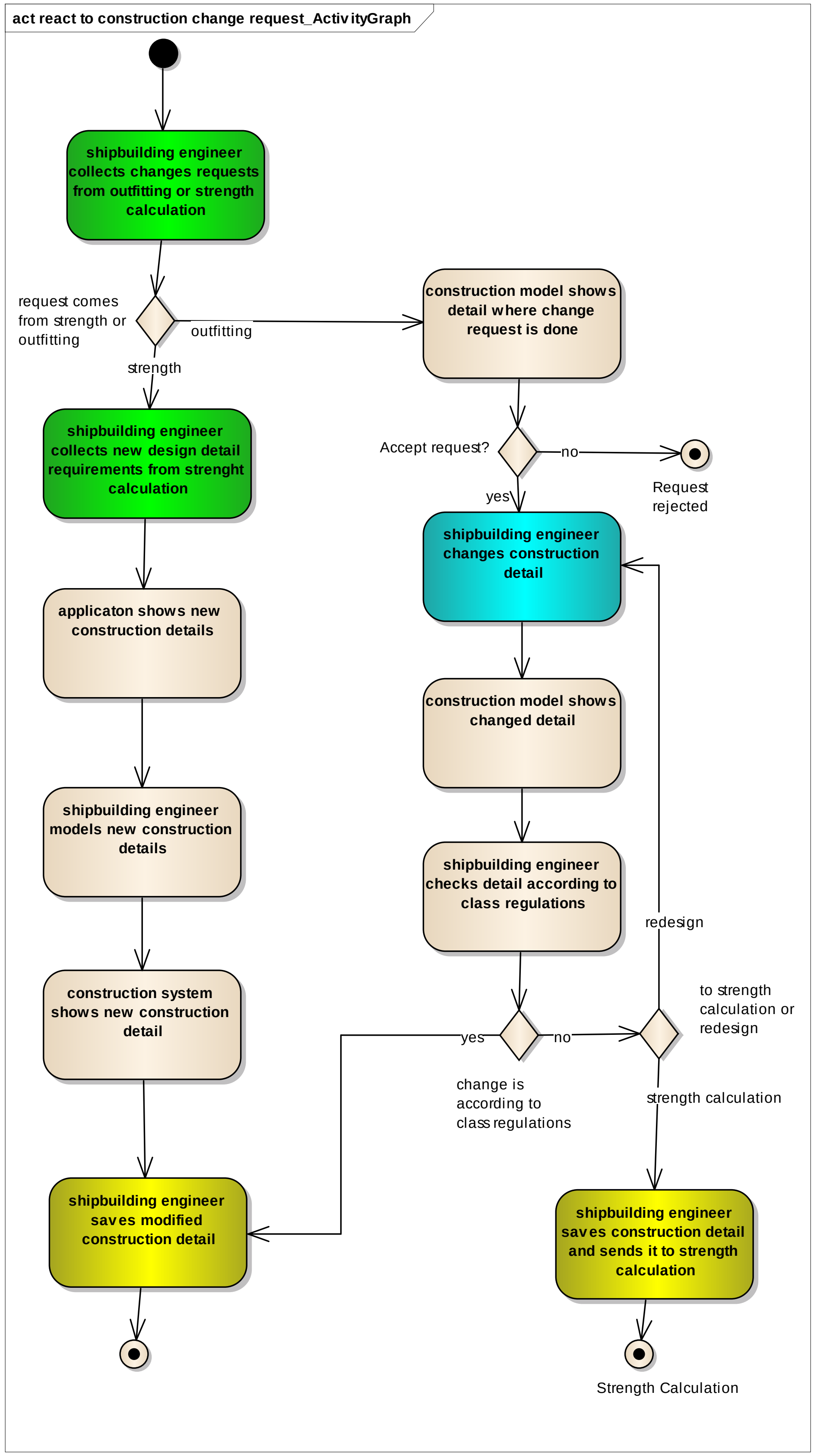

act make production information_ActivityGraph
shipbuilding engineer collects the finished hull construction model
construction model application shows finished area of ship
shipbuilding engineer generates production information, cutting, nesting, welding info.
sipbuilding engineer makes construction drawings
construction application shows the production information in its own file format
construction application shows the printable construction drawings
shipbuiling engineer saves the production information for the work preparation
shipbuilding engineer saves construction drawings for work preparation

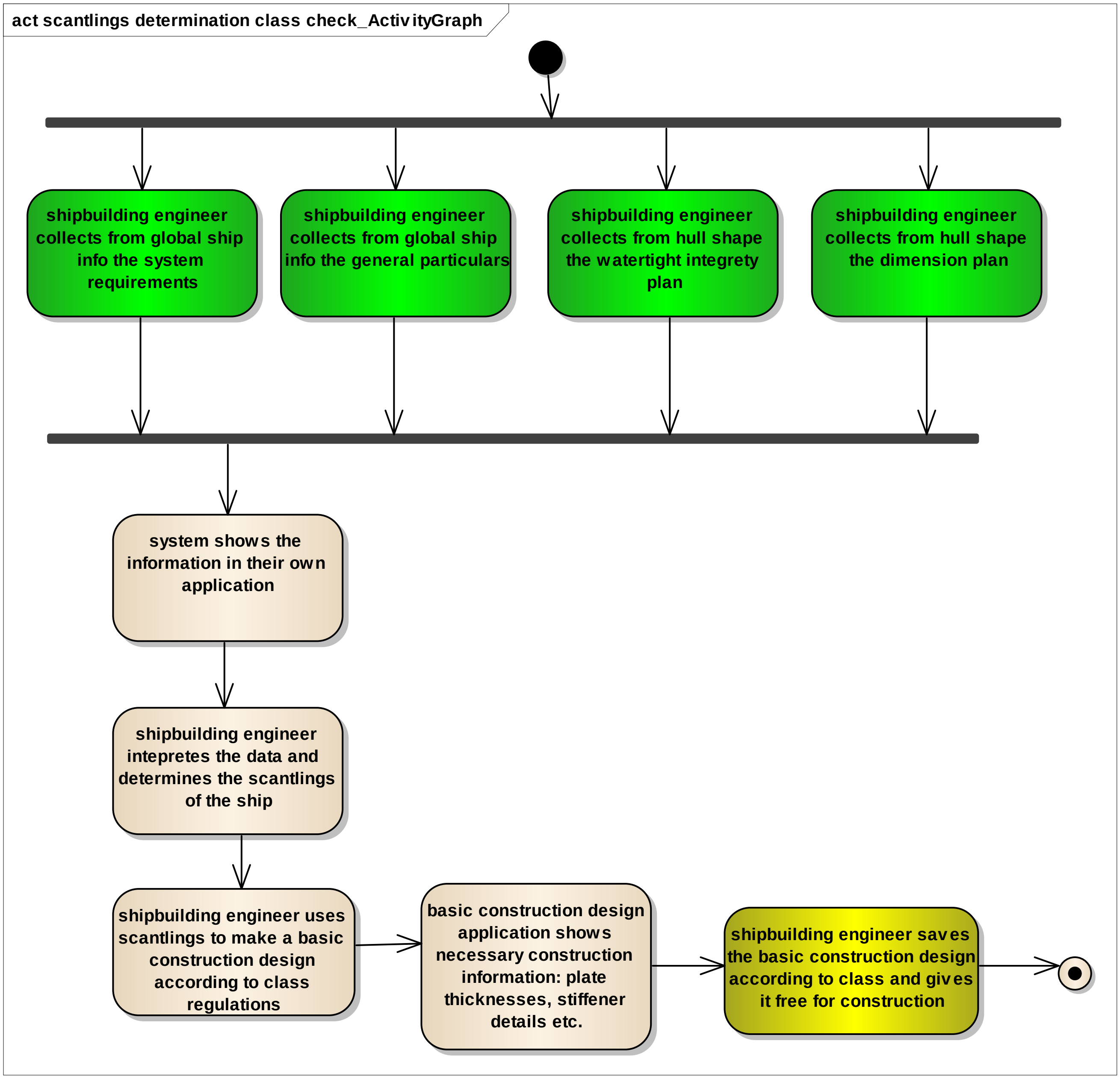

act building plan (panel, block, hull erection)_ActivityGraph
shipbuilding engineer collects yard planning and yard resource data
shipbuilding engineer collects generals arrangement
GA and shipyard data presents itself in their own applications
shipbuilding engineer makes a building strategy
application shows building strategy
shipbuilding engineer saves the building plan


act model hull construction_ActivityGraph
ship building engineer collects lines plan
shipbuilding engineer collects scantlings and class construction information
lines plan software shows linesplan
shipbuilding engineer exports the linesplan to construction sofware
construction software shows one solid hull shape
shipbuilding engineer constructs hull construction
construction software generates and shows hull construction
shipbuilding engineer saves the hull construction

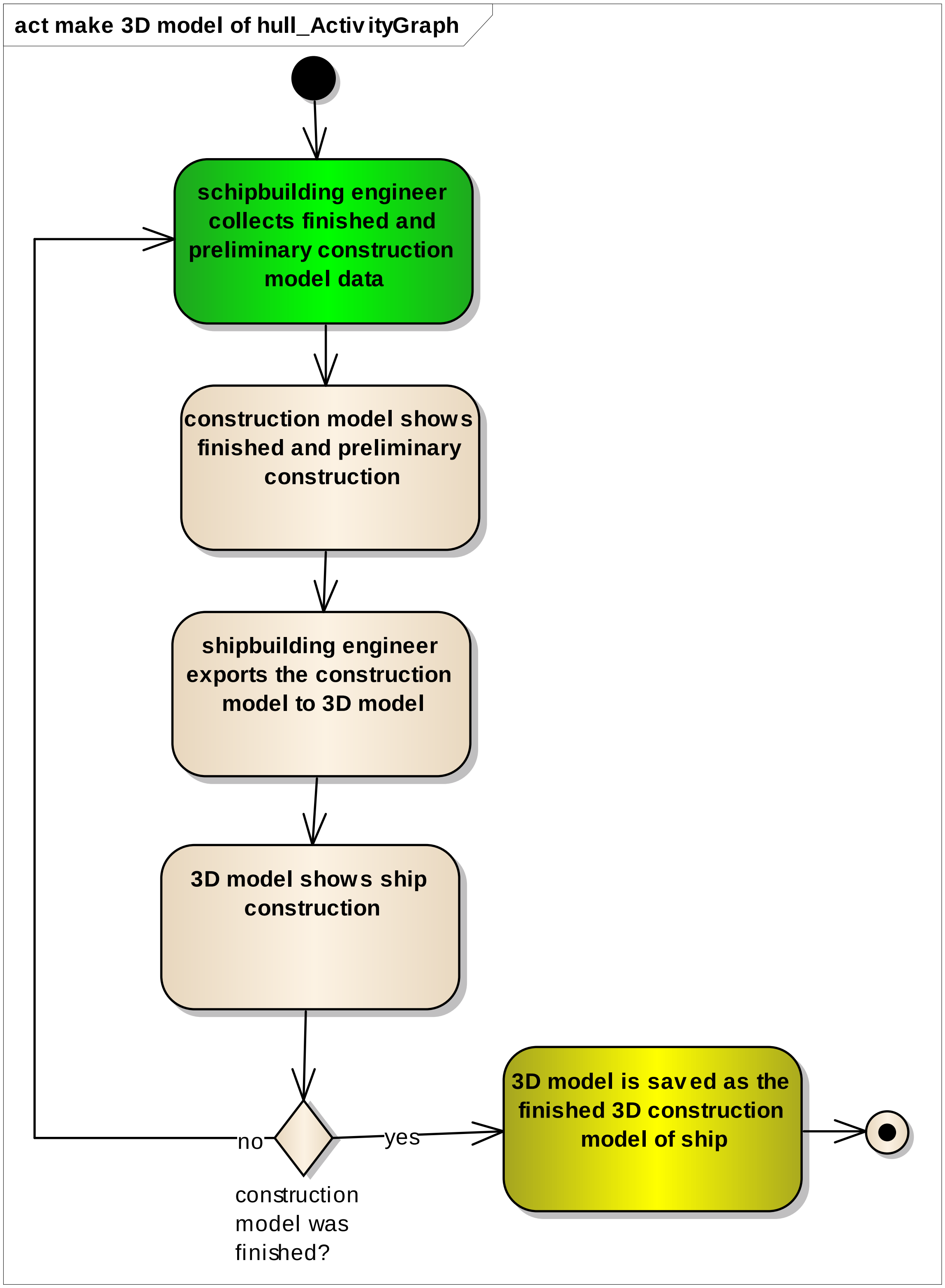

act model big construction elements_ActivityGraph
shipbuilding engineer collects scantlings and class construction information
shipbuilding engineer collects general arrangment, tank arrangment and space arrangment
shipbuilding engineer opens hull construction model
construction system shows hull construction
shipbuilding engineer constructs the big contruction elements
construction system shows hull construction containing big construction elements
shipbuilding engineer saves the shipconstruction

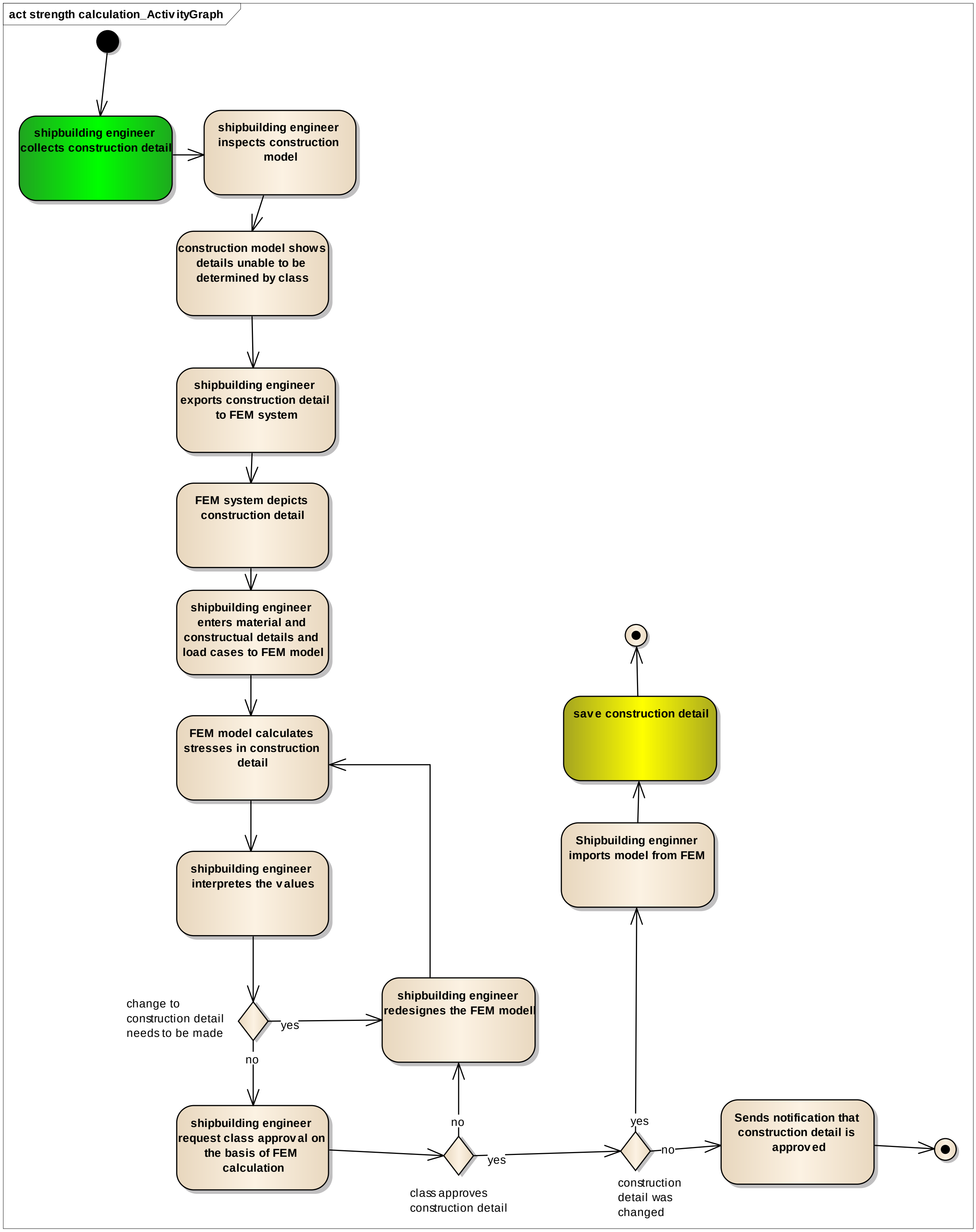

act model stiffening members_ActivityGraph
shipbuilding engineer collects scantlings and class construction information
shipbuilding engineer opens ship construction model
construction models shows ship construction
shipbuilding engineer constructs stiffening members
construction system shows complete construction model of the ship
shipbuilding engineer saves the model

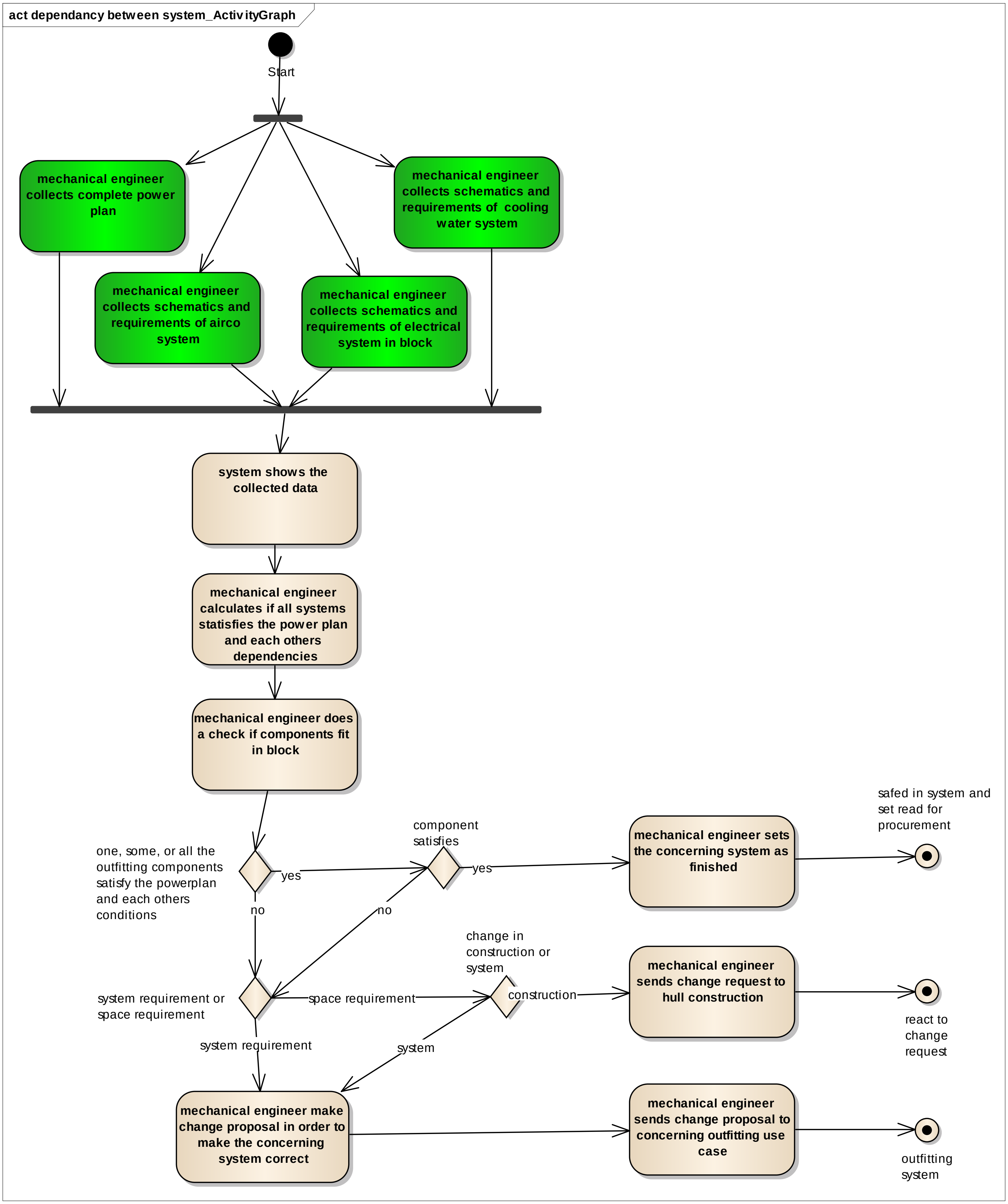

act react to construction change request_ActivityGraph
shipbuilding engineer collects changes requests from outfitting or strength calculation
request comes from strength or outfitting
outfitting
strength
construction model shows detail where change request is done
Accept request?
no
Request rejected
yes
shipbuilding engineer collects new design detail requirements from strenght calculation
shipbuilding engineer changes construction detail
applicaton shows new construction details
construction model shows changed detail
shipbuilding engineer models new construction details
shipbuilding engineer checks detail according to class regulations
redesign
construction system shows new construction detail
to strength calculation or redesign
yes
no
change is according to class regulations
strength calculation
shipbuilding engineer saves modified construction detail
shipbuilding engineer saves construction detail and sends it to strength calculation
Strength Calculation

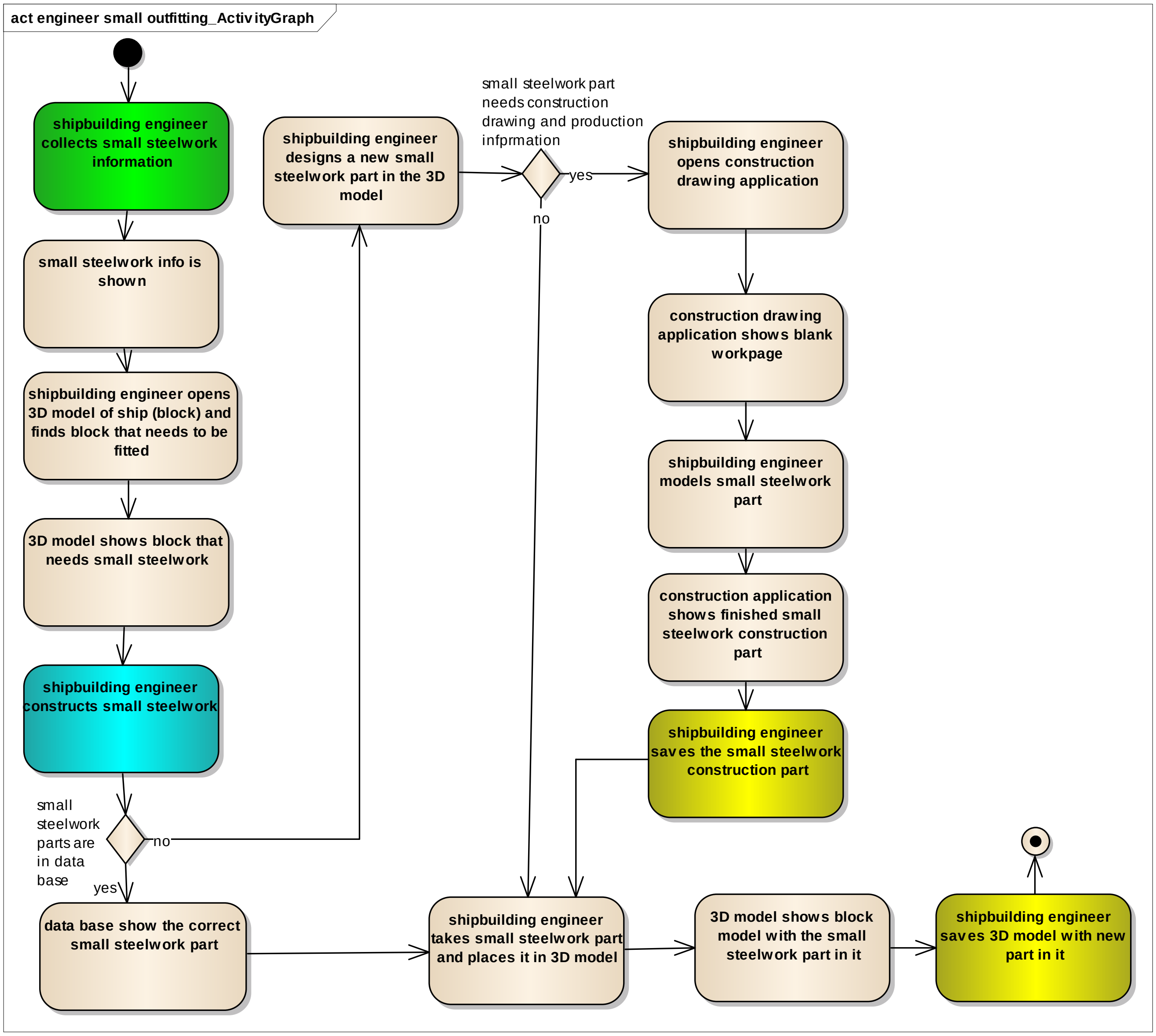


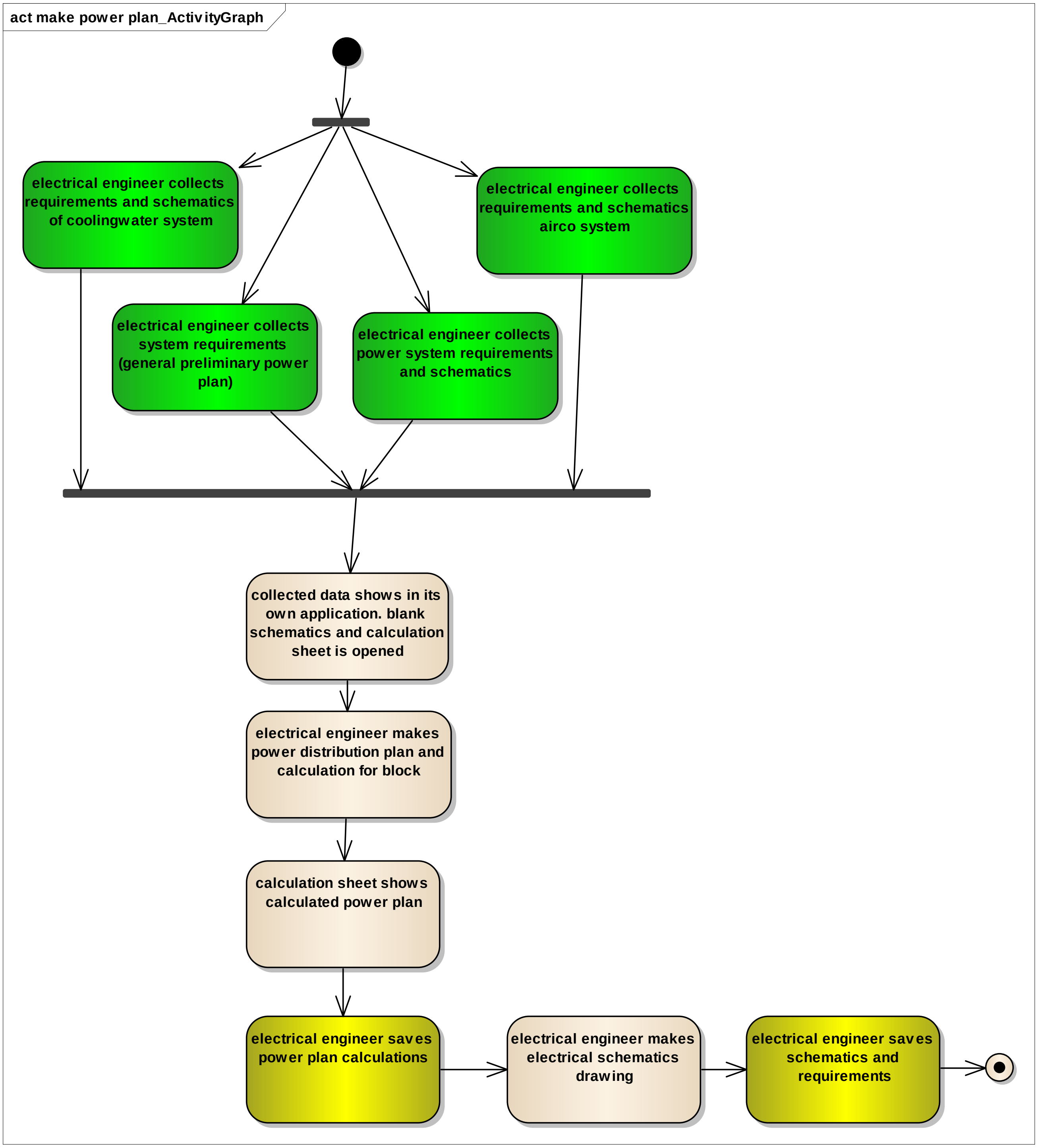
act scantlings determination class check_ActivityGraph
shipbuilding engineer collects from global ship info the system requirements
shipbuilding engineer collects from global ship info the general particulars
shipbuilding engineer collects from hull shape the watertight integrety plan
shipbuilding engineer collects from hull shape the dimension plan
system shows the information in their own application
shipbuilding engineer intepretes the data and determines the scantlings of the ship
shipbuilding engineer uses scantlings to make a basic construction design according to class regulations
basic construction design application shows necessary construction information: plate thicknesses, stiffener details etc.
shipbuilding engineer saves the basic construction design according to class and gives it free for construction

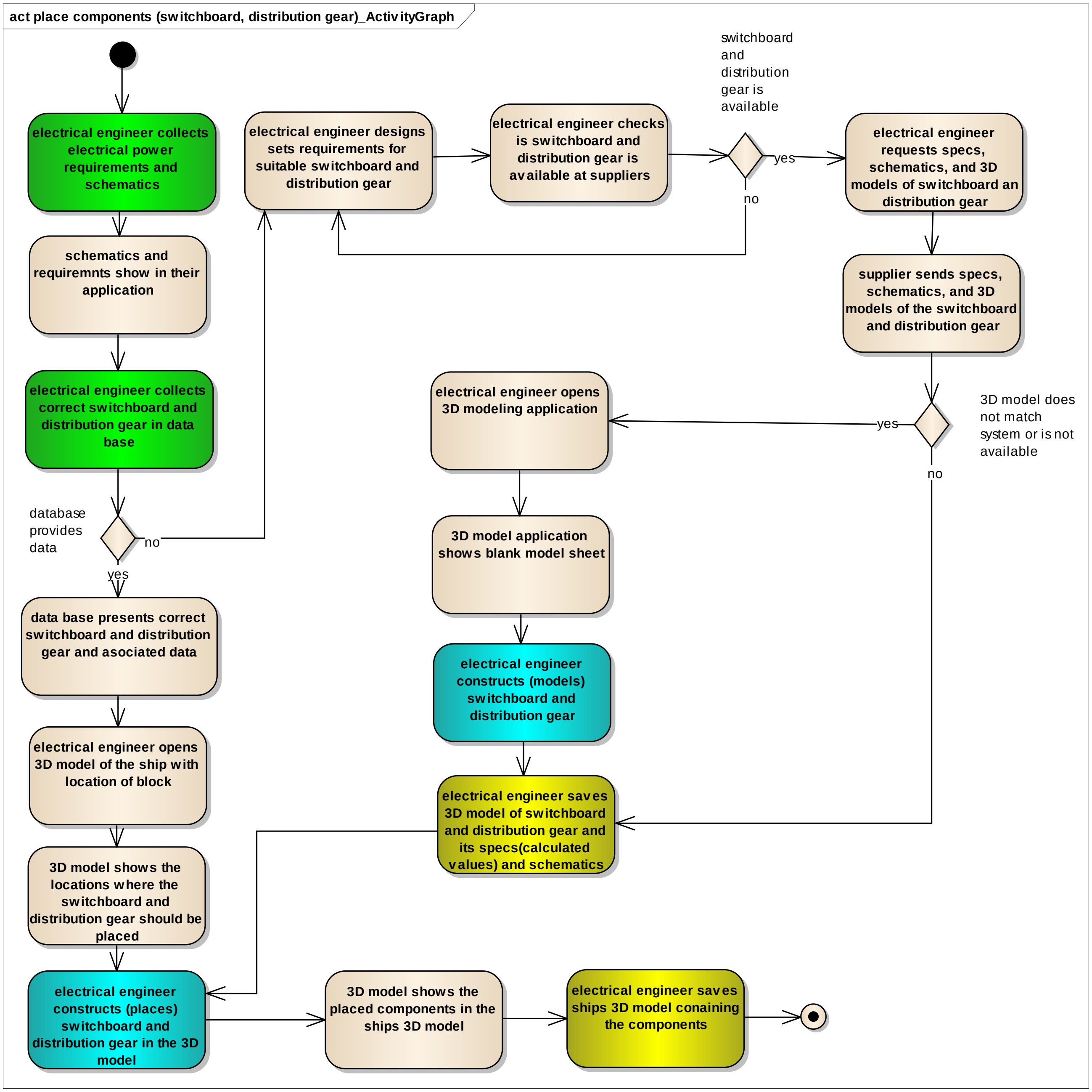

act make 3D model of hull_ActivityGraph
schipbuilding engineer collects finished and preliminary construction model data
construction model shows finished and preliminary construction
shipbuilding engineer exports the construction model to 3D model
3D model shows ship construction
no
yes
construction model was finished?
3D model is saved as the finished 3D construction model of ship

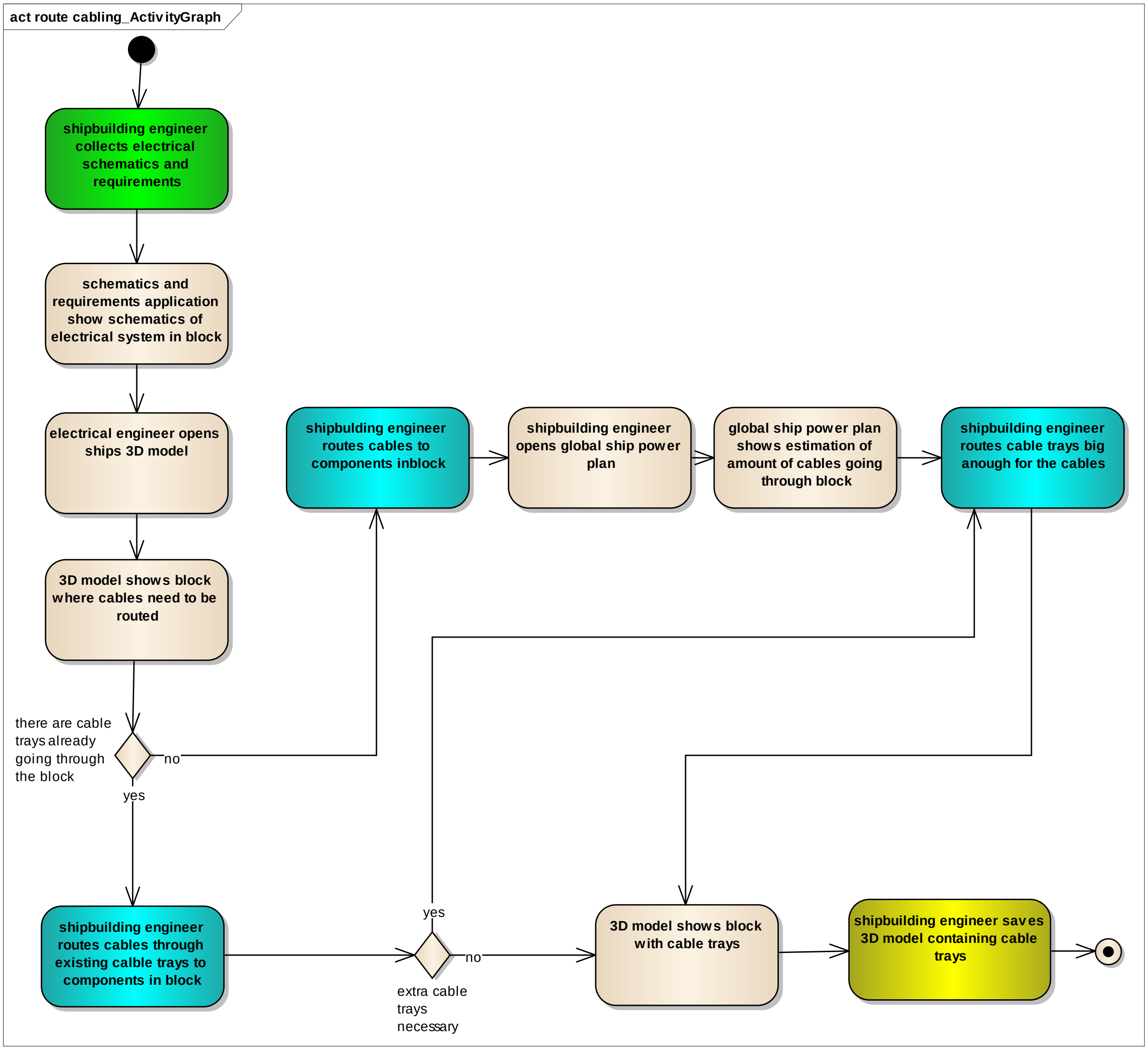

act strength calculation_ActivityGraph
shipbuilding engineer collects construction detail
shipbuilding engineer inspects construction model
construction model shows details unable to be determined by class
shipbuilding engineer exports construction detail to FEM system
FEM system depicts construction detail
shipbuilding engineer enters material and constructual details and load cases to FEM model
FEM model calculates stresses in construction detail
shipbuilding engineer interpretes the values
change to construction detail needs to be made
yes
no
shipbuilding engineer redesignes the FEM modell
shipbuilding engineer request class approval on the basis of FEM calculation
no
yes
class approves construction detail
yes
no
construction detail was changed
Sends notification that construction detail is approved
Shipbuilding enginner imports model from FEM
save construction detail

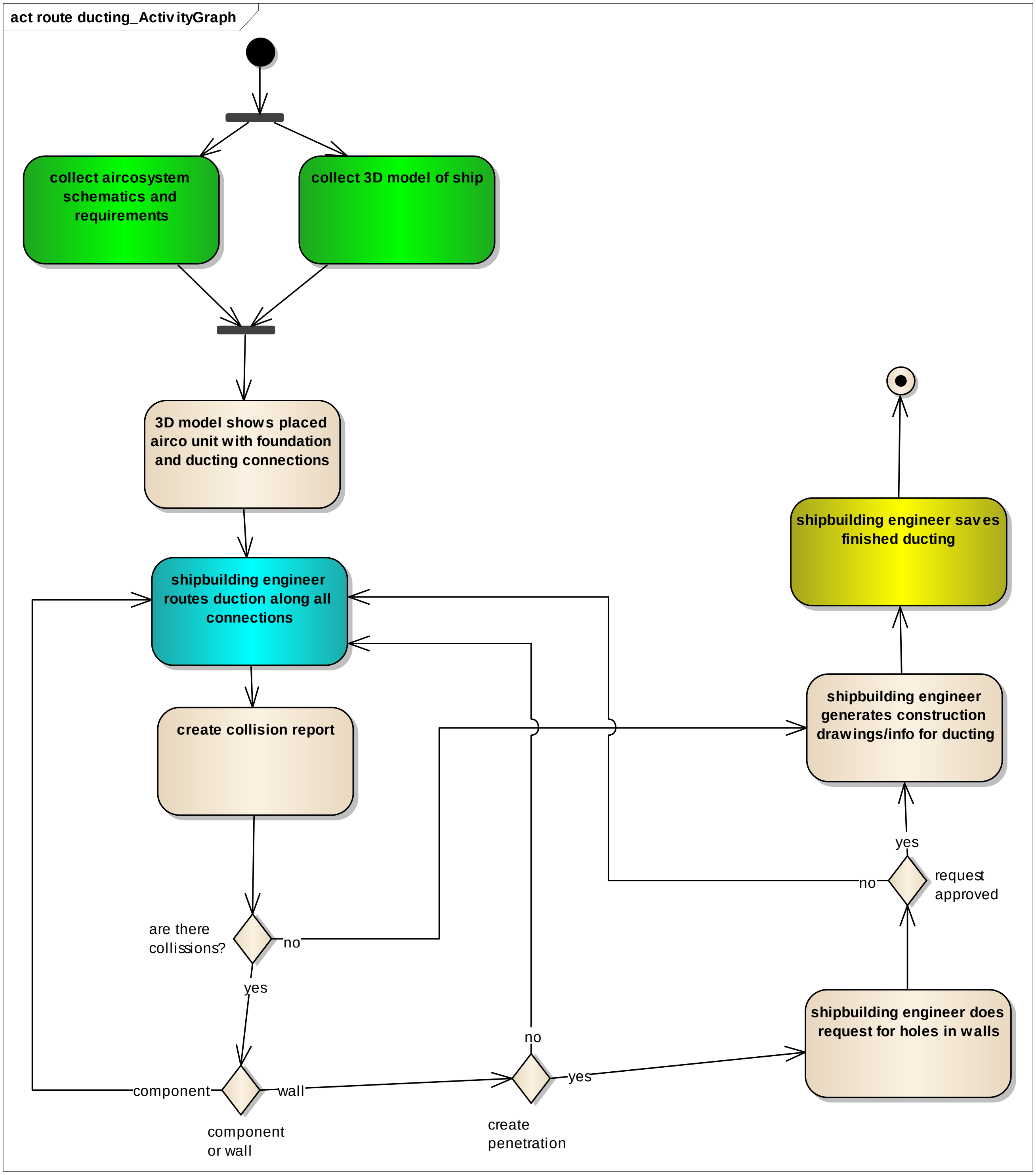

act dependancy between system_ActivityGraph
Start
mechanical engineer collects complete power plan
mechanical engineer collects schematics and requirements of cooling water system
mechanical engineer collects schematics and requirements of airco system
mechanical engineer collects schematics and requirements of electrical system in block
system shows the collected data
mechanical engineer calculates if all systems statisfies the power plan and each others dependencies
mechanical engineer does a check if components fit in block
one, some, or all the outfitting components satisfy the powerplan and each others conditions
yes
no
component satisfies
yes
no
mechanical engineer sets the concerning system as finished
safed in system and set read for procurement
system requirement or space requirement
space requirement
change in construction or system
construction
mechanical engineer sends change request to hull construction
react to change request
system requirement
system
mechanical engineer make change proposal in order to make the concerning system correct
mechanical engineer sends change proposal to concerning outfitting use case
outfitting system

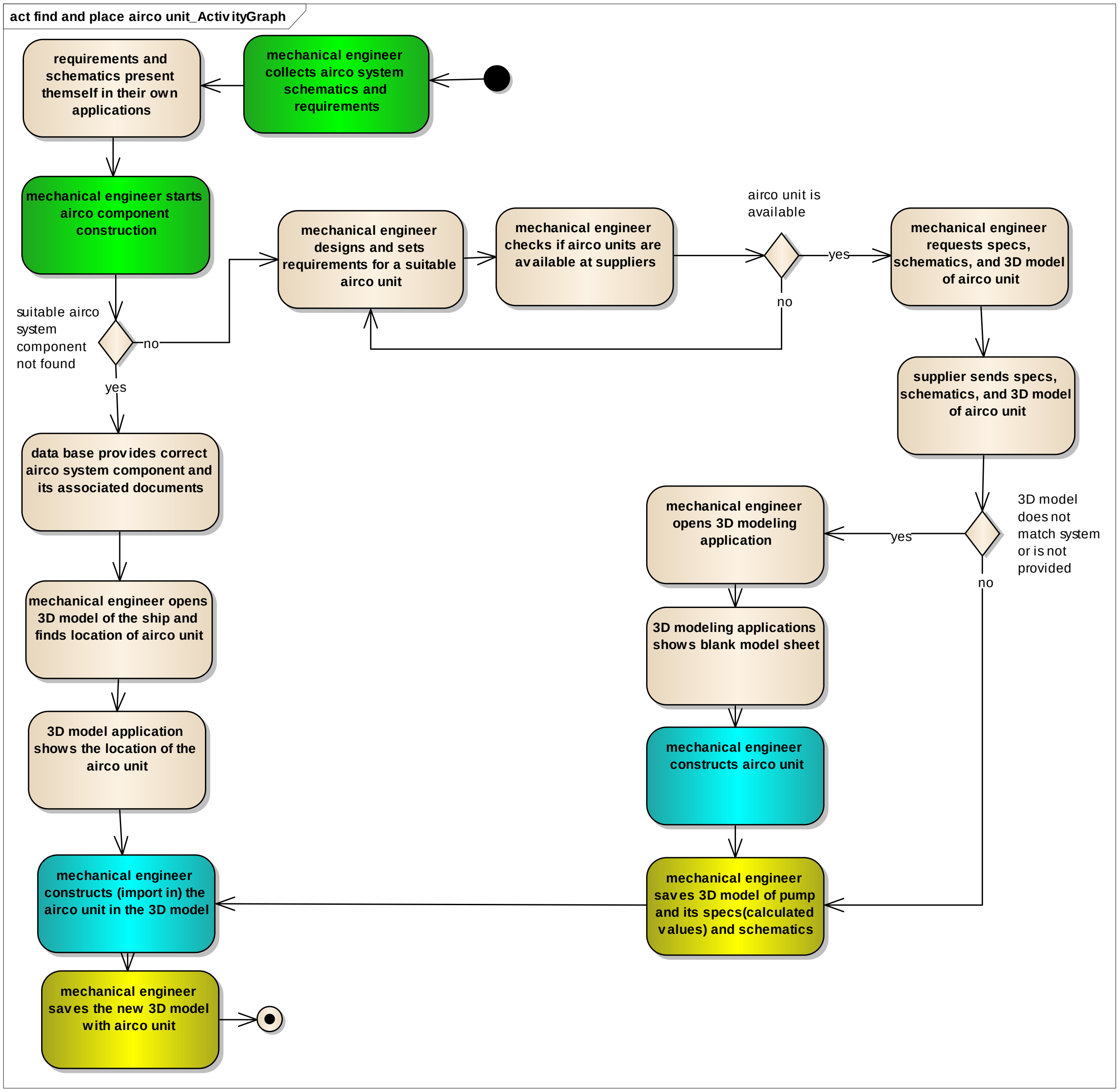

act engineer small outfitting_ActivityGraph
shipbuilding engineer collects small steelwork information
small steelwork info is shown
shipbuilding engineer opens 3D model of ship (block) and finds block that needs to be fitted
3D model shows block that needs small steelwork
shipbuilding engineer constructs small steelwork
small steelwork parts are in data base
no
yes
data base show the correct small steelwork part
shipbuilding engineer designs a new small steelwork part in the 3D model
small steelwork part needs construction drawing and production infprmation
yes
no
shipbuilding engineer opens construction drawing application
construction drawing application shows blank workpage
shipbuilding engineer models small steelwork part
construction application shows finished small steelwork construction part
shipbuilding engineer saves the small steelwork construction part
shipbuilding engineer takes small steelwork part and places it in 3D model
3D model shows block model with the small steelwork part in it
shipbuilding engineer saves 3D model with new part in it

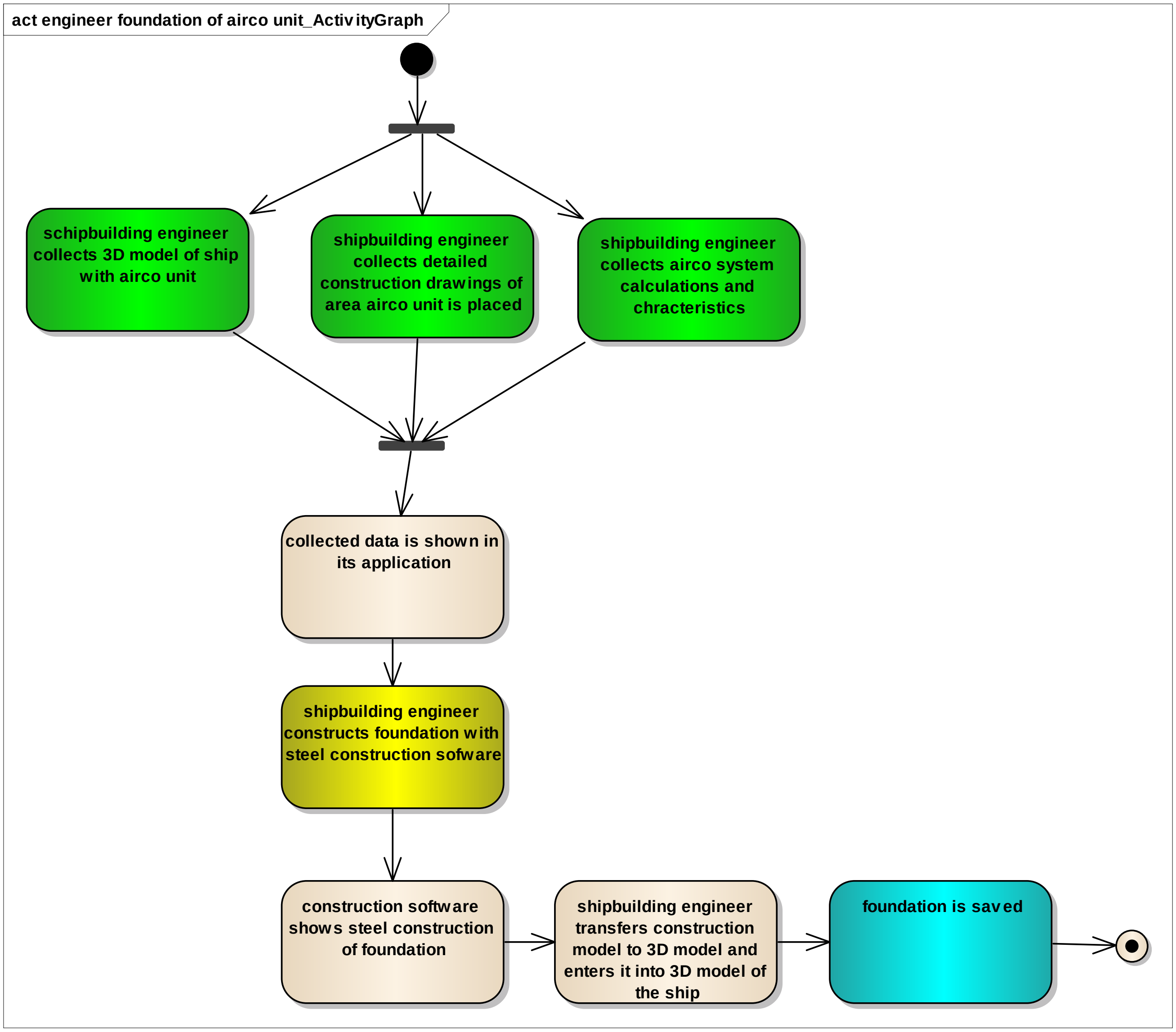

act make power plan_ActivityGraph
electrical engineer collects requirements and schematics of coolingwater system
electrical engineer collects requirements and schematics airco system
electrical engineer collects system requirements (general preliminary power plan)
electrical engineer collects power system requirements and schematics
collected data shows in its own application. blank schematics and calculation sheet is opened
electrical engineer makes power distribution plan and calculation for block
calculation sheet shows calculated power plan
electrical engineer saves power plan calculations
electrical engineer makes electrical schematics drawing
electrical engineer saves schematics and requirements

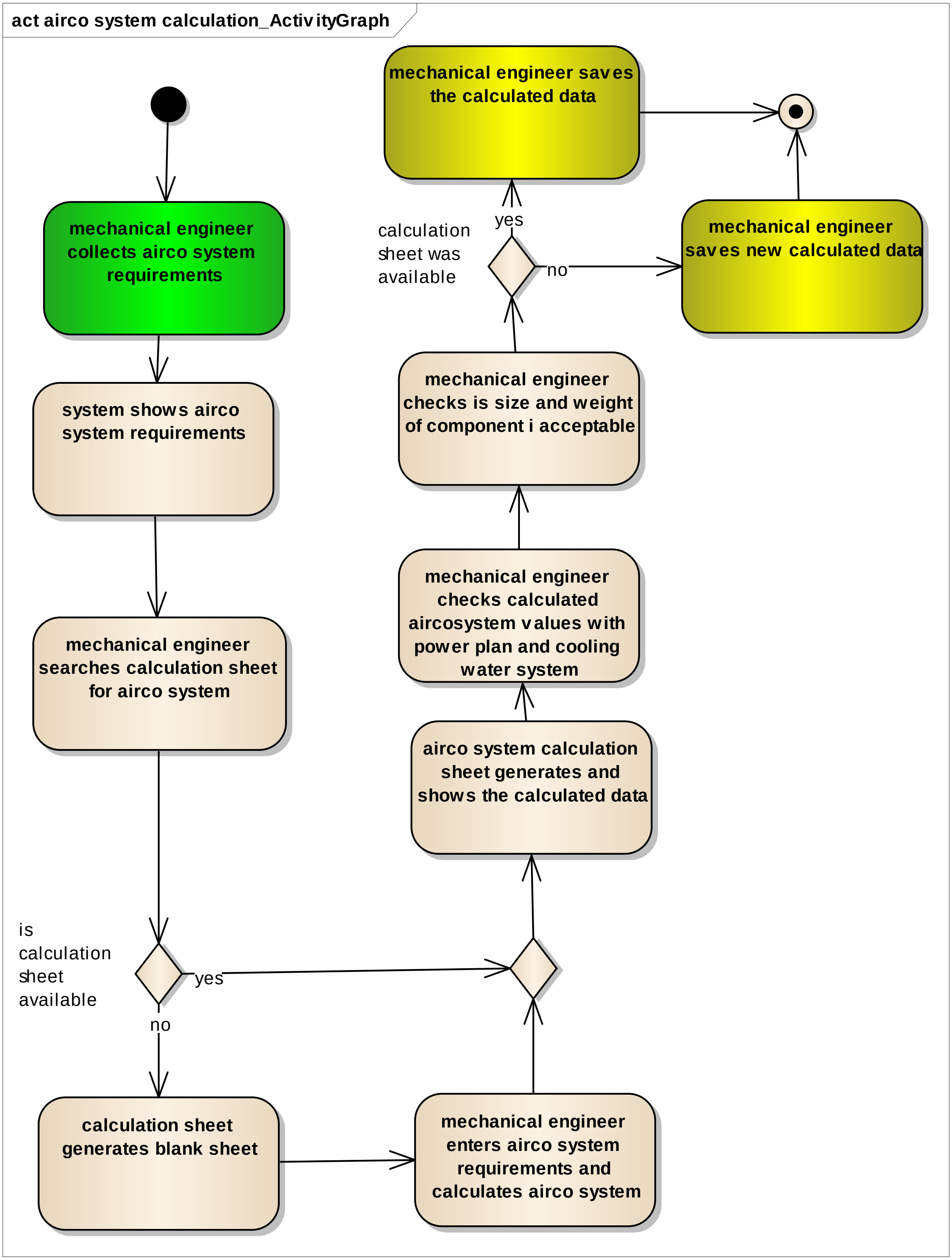

act place components (switchboard, distribution gear)_ActivityGraph
electrical engineer collects electrical power requirements and schematics
schematics and requiremnts show in their application
electrical engineer collects correct switchboard and distribution gear in data base
database provides data
no
yes
data base presents correct switchboard and distribution gear and asociated data
electrical engineer opens 3D model of the ship with location of block
3D model shows the locations where the switchboard and distribution gear should be placed
electrical engineer constructs (places) switchboard and distribution gear in the 3D model
3D model shows the placed components in the ships 3D model
electrical engineer saves ships 3D model conaining the components
electrical engineer designs sets requirements for suitable switchboard and distribution gear
electrical engineer checks is switchboard and distribution gear is available at suppliers
switchboard and distribution gear is available
yes
no
electrical engineer requests specs, schematics, and 3D models of switchboard an distribution gear
supplier sends specs, schematics, and 3D models of the switchboard and distribution gear
3D model does not match system or is not available
yes
no
electrical engineer opens 3D modeling application
3D model application shows blank model sheet
electrical engineer constructs (models) switchboard and distribution gear
electrical engineer saves 3D model of switchboard and distribution gear and its specs(calculated values) and schematics

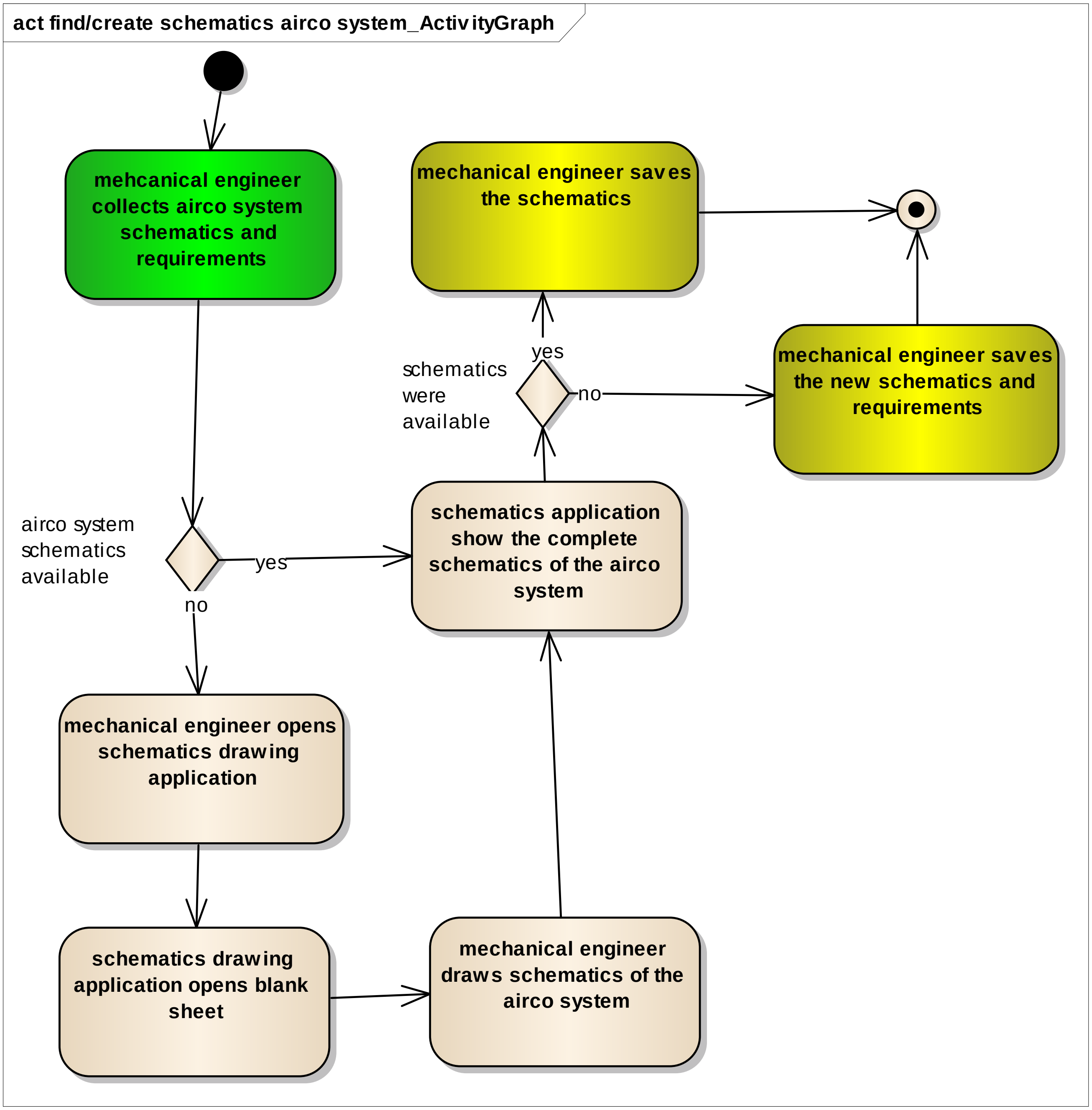

act route cabling_ActivityGraph
shipbuilding engineer collects electrical schematics and requirements
schematics and requirements application show schematics of electrical system in block
electrical engineer opens ships 3D model
3D model shows block where cables need to be routed
there are cable trays already going through the block
no
yes
shipbuilding engineer routes cables to components inblock
shipbuilding engineer opens global ship power plan
global ship power plan shows estimation of amount of cables going through block
shipbuilding engineer routes cable trays big anough for the cables
shipbuilding engineer routes cables through existing calble trays to components in block
yes
no
extra cable trays necessary
3D model shows block with cable trays
shipbuilding engineer saves 3D model containing cable trays

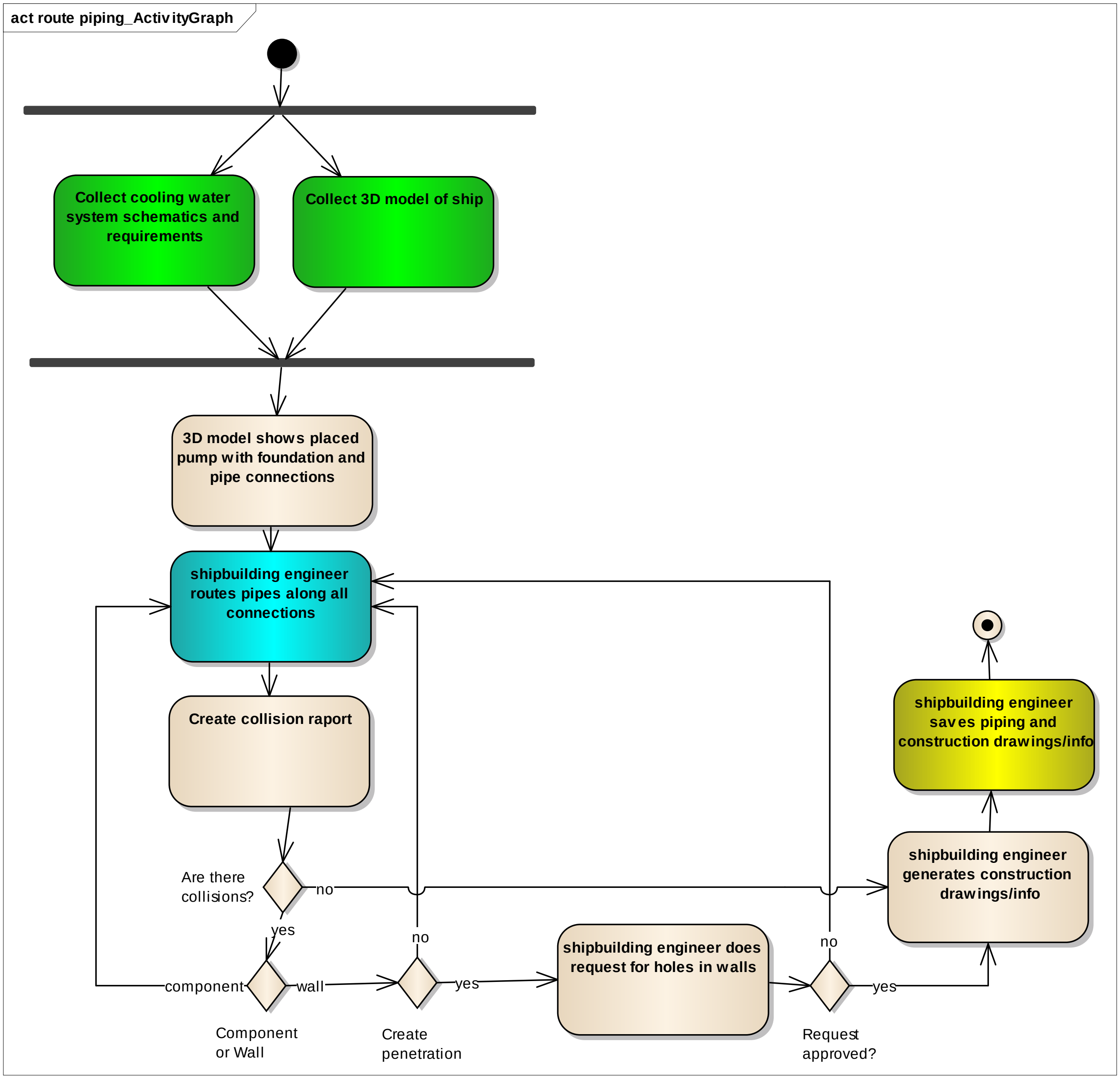

act route ducting_ActivityGraph
collect aircosystem schematics and requirements
collect 3D model of ship
3D model shows placed airco unit with foundation and ducting connections
shipbuilding engineer routes duction along all connections
create collision report
are there collissions?
no
yes
component
wall
component or wall
no
yes
create penetration
shipbuilding engineer does request for holes in walls
request approved
yes
no
shipbuilding engineer generates construction drawings/info for ducting
shipbuilding engineer saves finished ducting

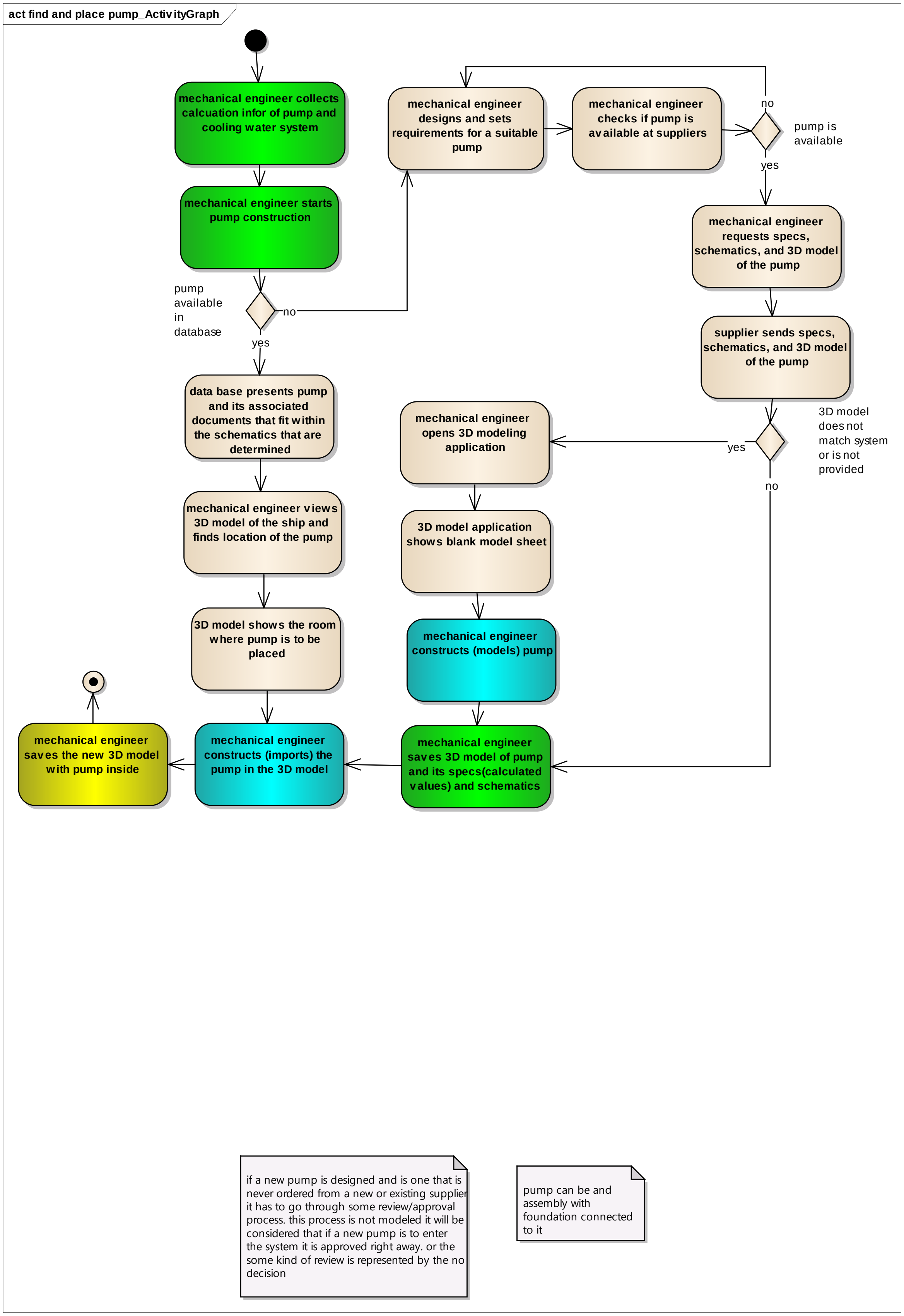

act find and place airco unit_ActivityGraph
requirements and schematics present themself in their own applications
mechanical engineer collects airco system schematics and requirements
mechanical engineer starts airco component construction
mechanical engineer designs and sets requirements for a suitable airco unit
mechanical engineer checks if airco units are available at suppliers
airco unit is available
yes
no
mechanical engineer requests specs, schematics, and 3D model of airco unit
suitable airco system component not found
no
yes
supplier sends specs, schematics, and 3D model of airco unit
data base provides correct airco system component and its associated documents
mechanical engineer opens 3D modeling application
yes
3D model does not match system or is not provided
no
mechanical engineer opens 3D model of the ship and finds location of airco unit
3D modeling applications shows blank model sheet
3D model application shows the location of the airco unit
mechanical engineer constructs airco unit
mechanical engineer constructs (import in) the airco unit in the 3D model
mechanical engineer saves 3D model of pump and its specs(calculated values) and schematics
mechanical engineer saves the new 3D model with airco unit

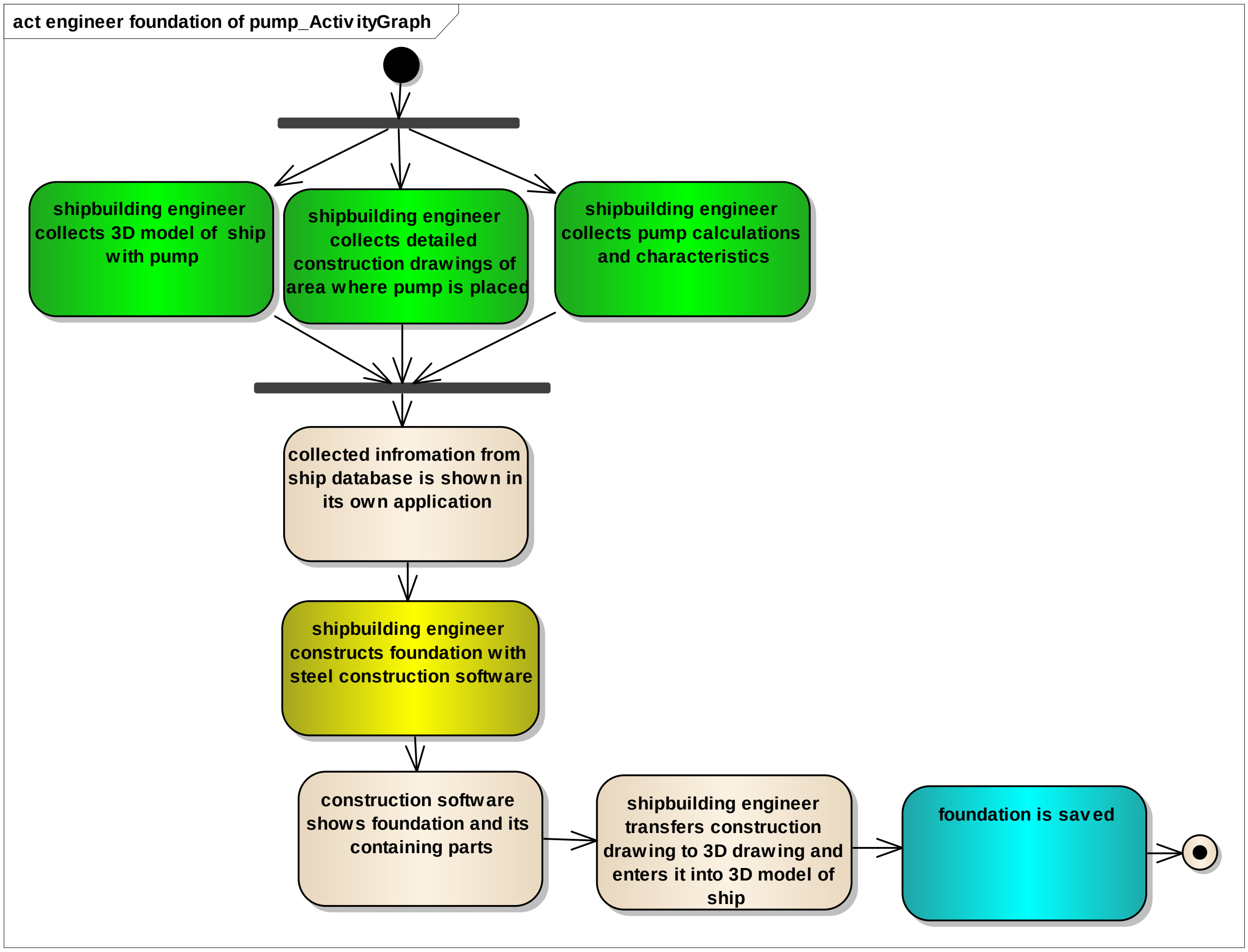

act engineer foundation of airco unit_ActivityGraph
schipbuilding engineer collects 3D model of ship with airco unit
shipbuilding engineer collects detailed construction drawings of area airco unit is placed
shipbuilding engineer collects airco system calculations and chracteristics
collected data is shown in its application
shipbuilding engineer constructs foundation with steel construction sofware
construction software shows steel construction of foundation
shipbuilding engineer transfers construction model to 3D model and enters it into 3D model of the ship
foundation is saved

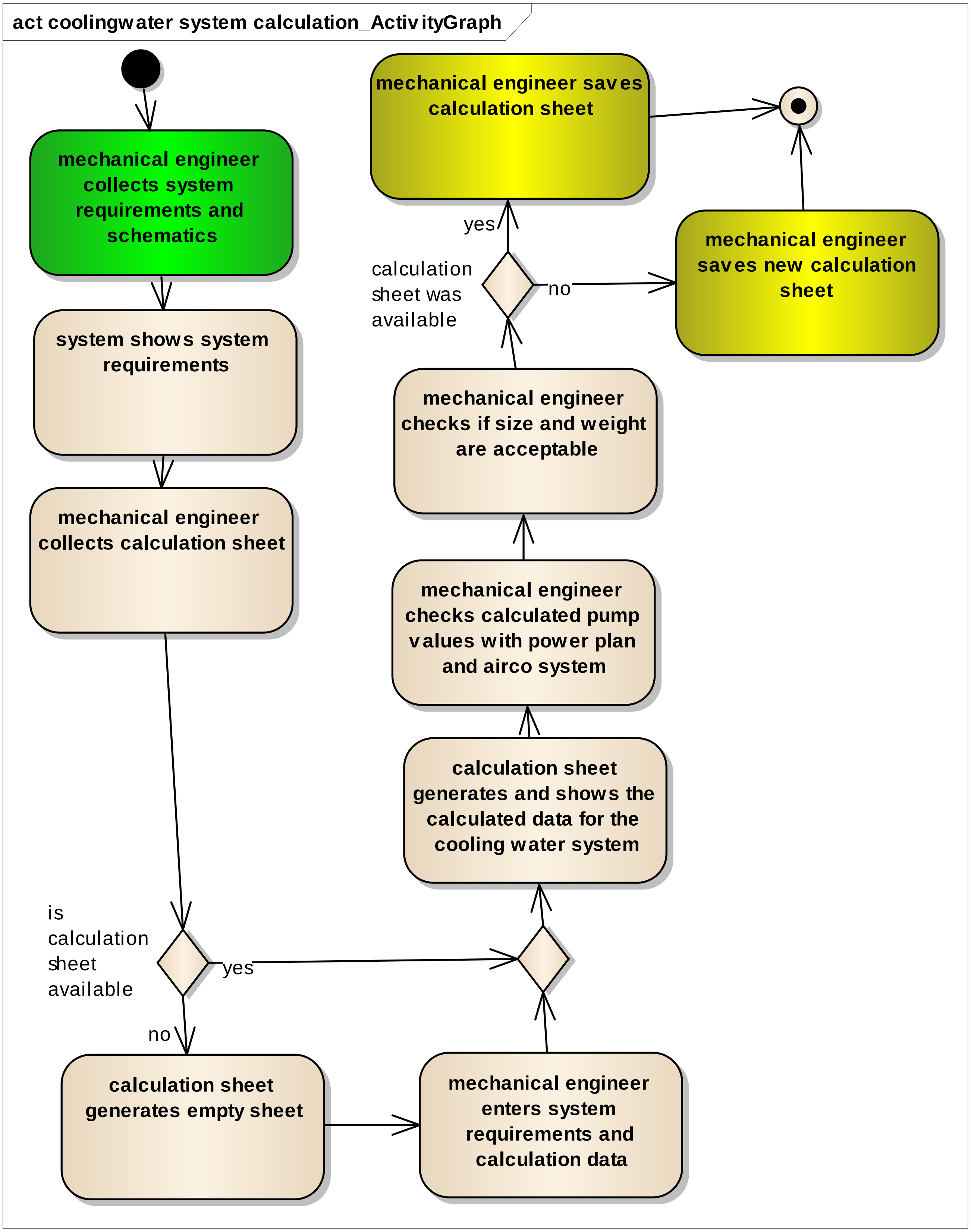

act airco system calculation_ActivityGraph
mechanical engineer collects airco system requirements
system shows airco system requirements
mechanical engineer searches calculation sheet for airco system
is calculation sheet available
yes
no
calculation sheet generates blank sheet
mechanical engineer enters airco system requirements and calculates airco system
airco system calculation sheet generates and shows the calculated data
mechanical engineer checks calculated aircosystem values with power plan and cooling water system
mechanical engineer checks is size and weight of component i acceptable
calculation sheet was available
yes
no
mechanical engineer saves the calculated data
mechanical engineer saves new calculated data

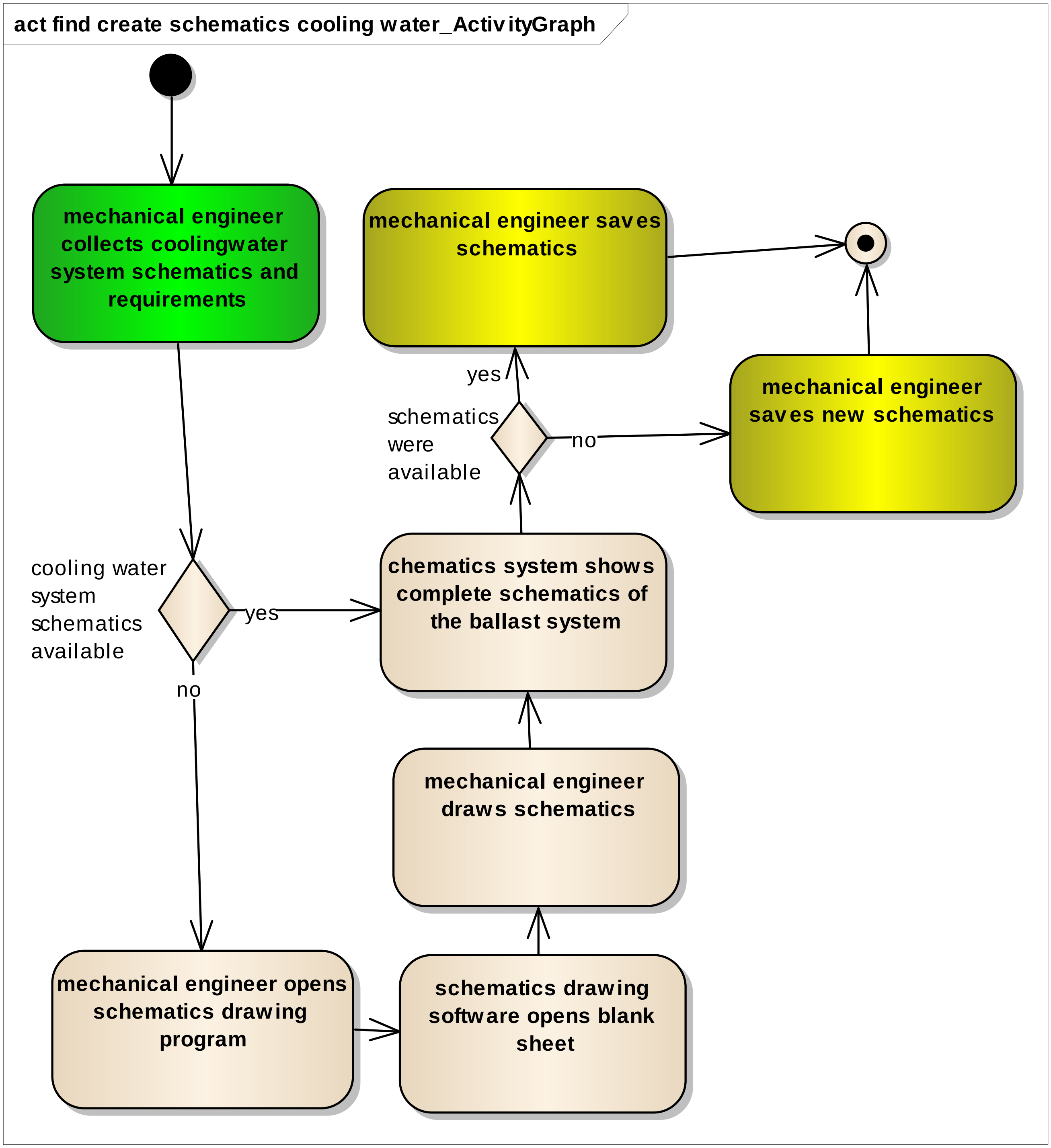

act find/create schematics airco system_ActivityGraph
mehcanical engineer collects airco system schematics and requirements
airco system schematics available
yes
no
mechanical engineer opens schematics drawing application
schematics drawing application opens blank sheet
mechanical engineer draws schematics of the airco system
schematics application show the complete schematics of the airco system
schematics were available
yes
no
mechanical engineer saves the schematics
mechanical engineer saves the new schematics and requirements

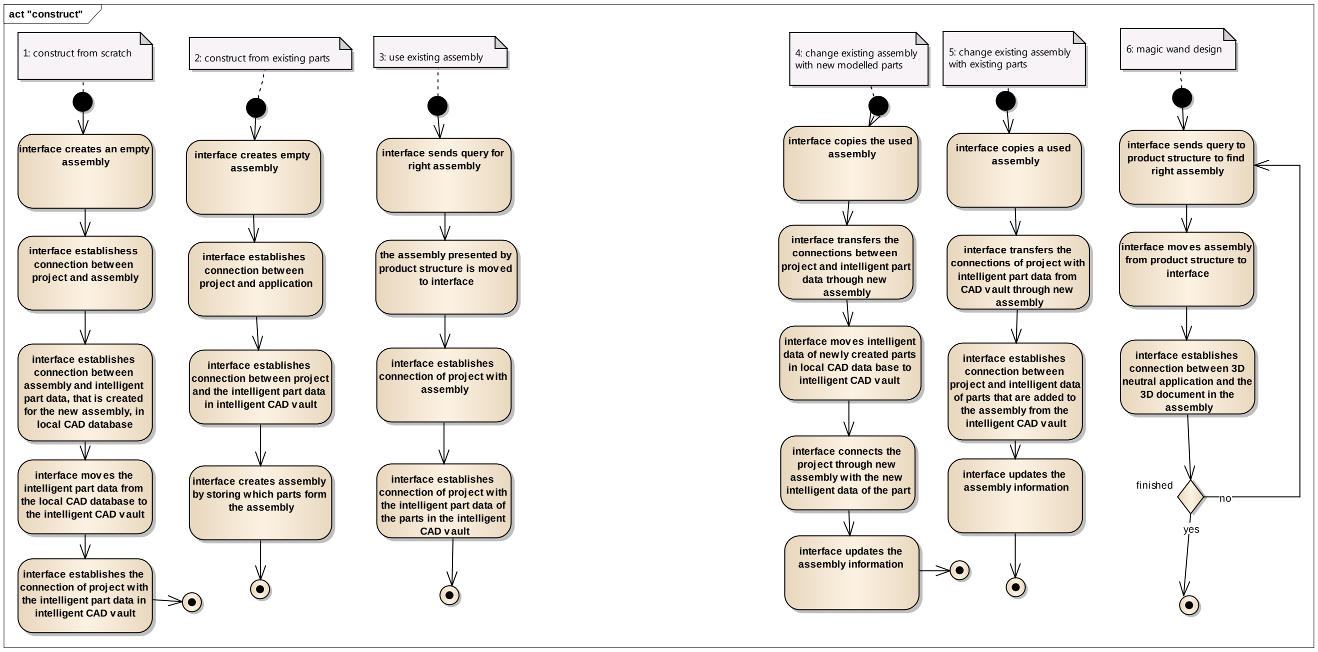

act route piping_ActivityGraph
Collect cooling water system schematics and requirements
Collect 3D model of ship
3D model shows placed pump with foundation and pipe connections
shipbuilding engineer routes pipes along all connections
Create collision raport
Are there collisions?
no
yes
component
wall
Component or Wall
no
yes
Create penetration
shipbuilding engineer does request for holes in walls
no
yes
Request approved?
shipbuilding engineer generates construction drawings/info
shipbuilding engineer saves piping and construction drawings/info


act find and place pump_ActivityGraph
mechanical engineer collects calcuation infor of pump and cooling water system
mechanical engineer starts pump construction
pump available in database
no
yes
data base presents pump and its associated documents that fit within the schematics that are determined
mechanical engineer views 3D model of the ship and finds location of the pump
3D model shows the room where pump is to be placed
mechanical engineer constructs (imports) the pump in the 3D model
mechanical engineer saves the new 3D model with pump inside
mechanical engineer designs and sets requirements for a suitable pump
mechanical engineer checks if pump is available at suppliers
no
pump is available
yes
mechanical engineer requests specs, schematics, and 3D model of the pump
supplier sends specs, schematics, and 3D model of the pump
3D model does not match system or is not provided
yes
no
mechanical engineer opens 3D modeling application
3D model application shows blank model sheet
mechanical engineer constructs (models) pump
mechanical engineer saves 3D model of pump and its specs(calculated values) and schematics
if a new pump is designed and is one that is never ordered from a new or existing supplier it has to go through some review/approval process. this process is not modeled it will be considered that if a new pump is to enter the system it is approved right away. or the some kind of review is represented by the no decision
pump can be and assembly with foundation connected to it

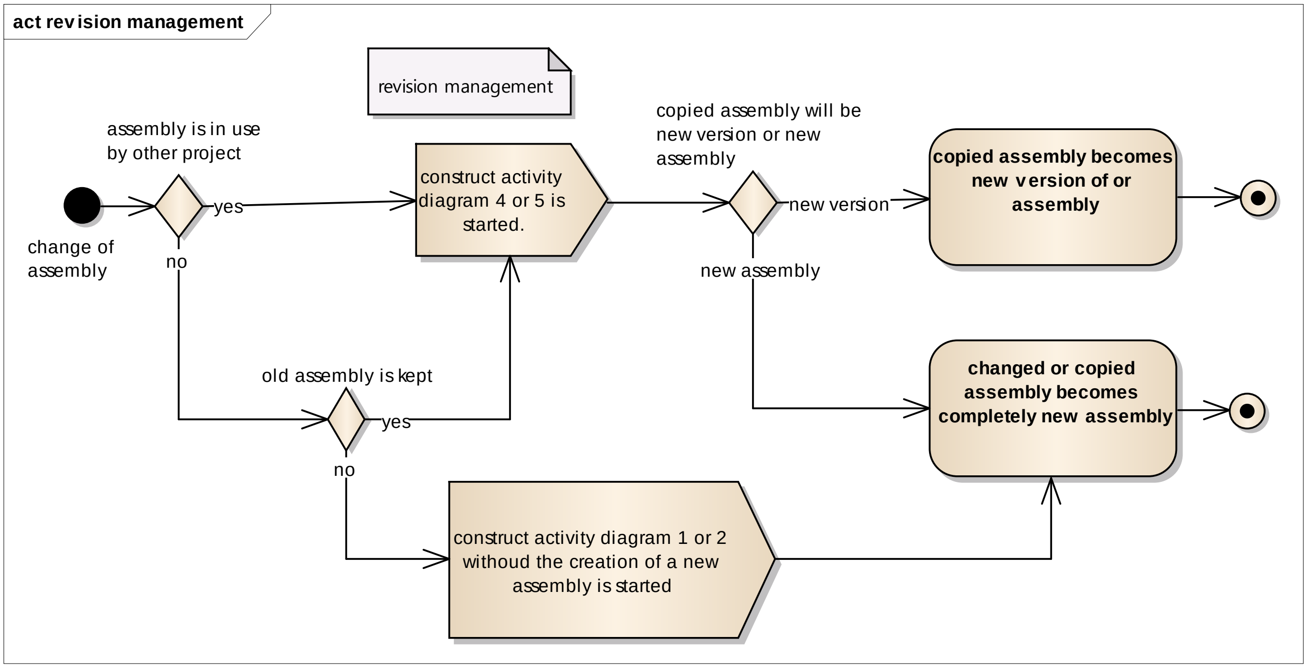

act engineer foundation of pump_ActivityGraph
shipbuilding engineer collects 3D model of ship with pump
shipbuilding engineer collects detailed construction drawings of area where pump is placed
shipbuilding engineer collects pump calculations and characteristics
collected infromation from ship database is shown in its own application
shipbuilding engineer constructs foundation with steel construction software
construction software shows foundation and its containing parts
shipbuilding engineer transfers construction drawing to 3D drawing and enters it into 3D model of ship
foundation is saved

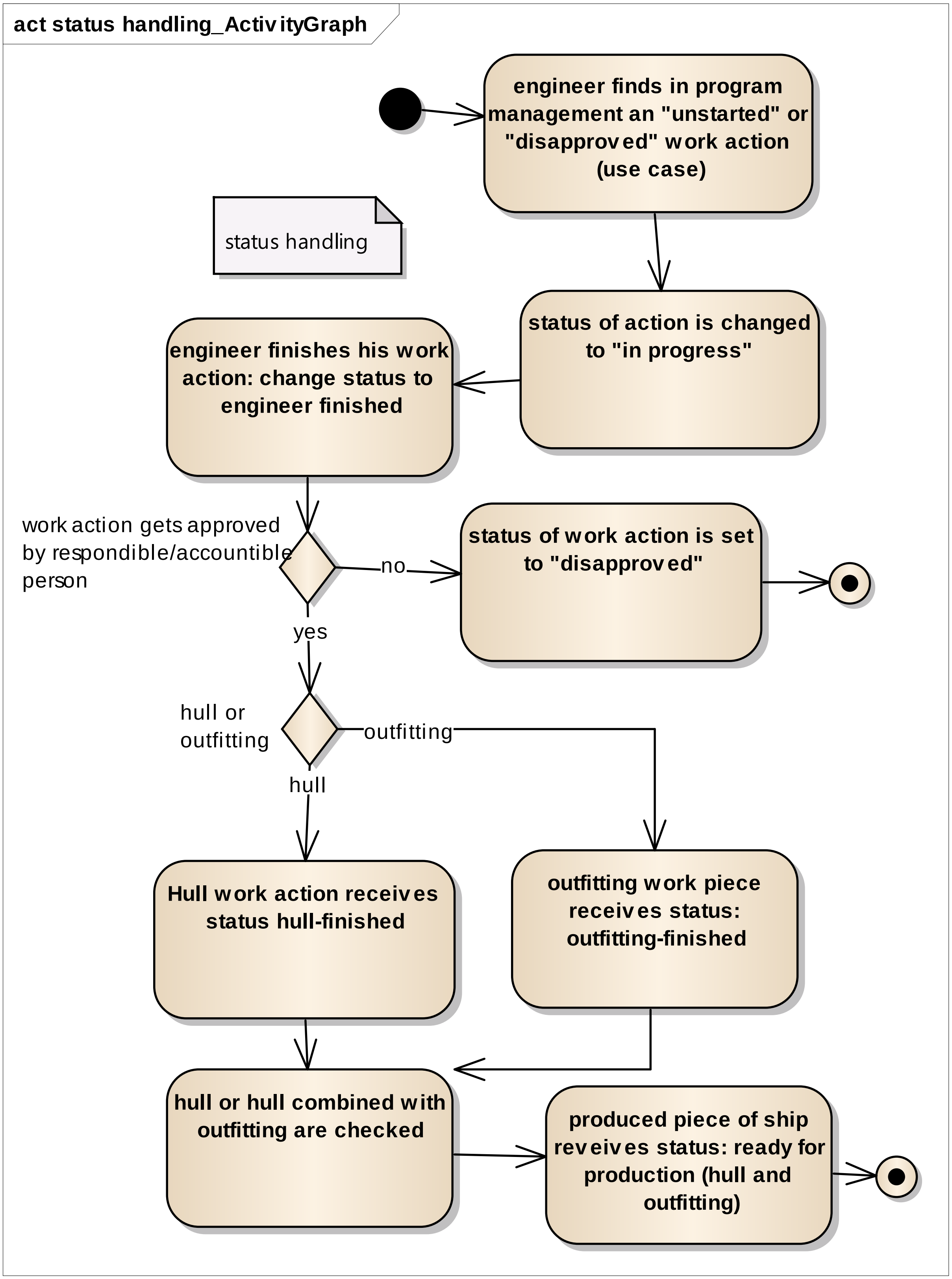

act coolingwater system calculation_ActivityGraph
mechanical engineer collects system requirements and schematics
system shows system requirements
mechanical engineer collects calculation sheet
is calculation sheet available
yes
no
calculation sheet generates empty sheet
mechanical engineer enters system requirements and calculation data
calculation sheet generates and shows the calculated data for the cooling water system
mechanical engineer checks calculated pump values with power plan and airco system
mechanical engineer checks if size and weight are acceptable
calculation sheet was available
yes
no
mechanical engineer saves calculation sheet
mechanical engineer saves new calculation sheet

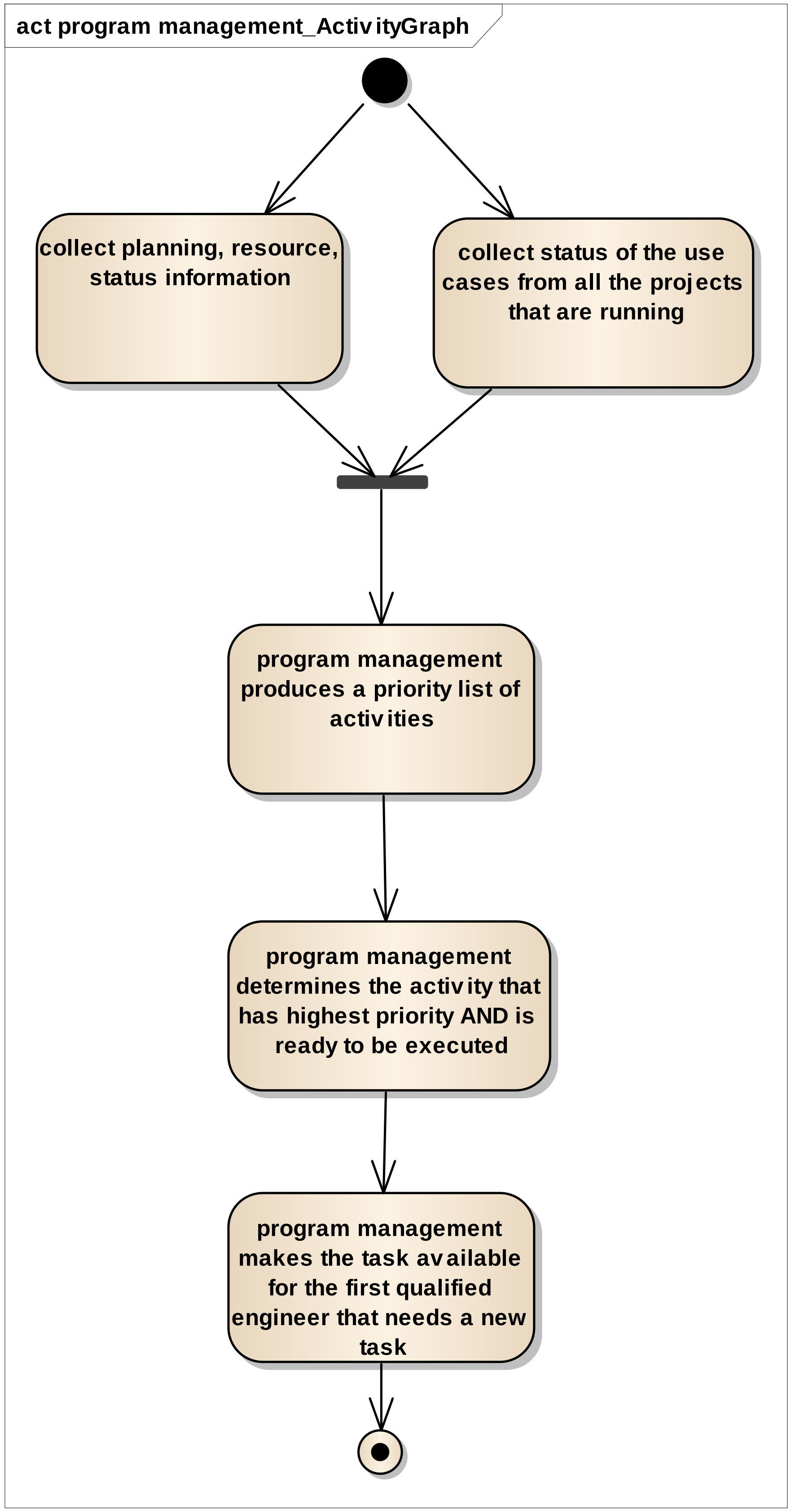

act find create schematics cooling water_ActivityGraph
mechanical engineer collects coolingwater system schematics and requirements
mechanical engineer saves schematics
yes
schematics were available
no
mechanical engineer saves new schematics
cooling water system schematics available
yes
chematics system shows complete schematics of the ballast system
no
mechanical engineer draws schematics
mechanical engineer opens schematics drawing program
schematics drawing software opens blank sheet

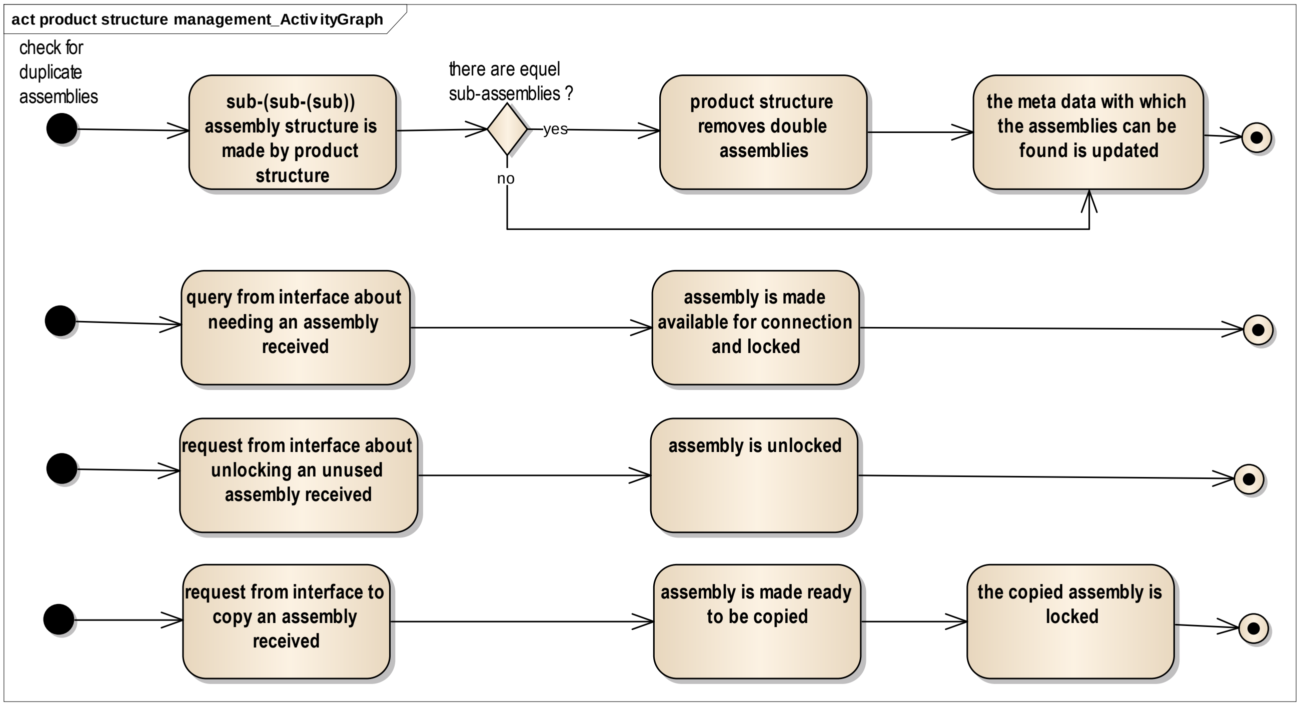


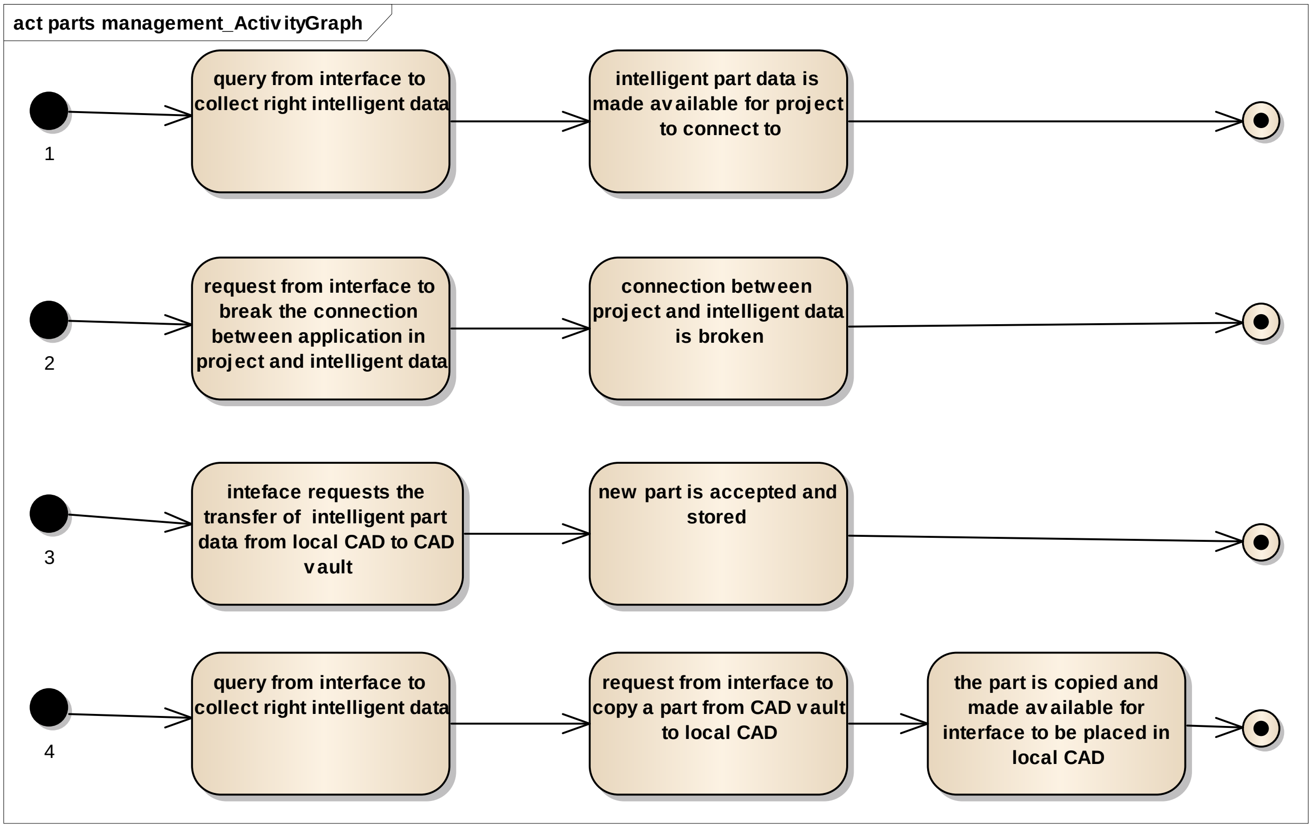
act "construct"
1: construct from scratch
interface creates an empty assembly
interface establishess connection between project and assembly
interface establishes connection between assembly and intelligent part data, that is created for the new assembly, in local CAD database
interface moves the intelligent part data from the local CAD database to the intelligent CAD vault
interface establishes the connection of project with the intelligent part data in intelligent CAD vault
2: construct from existing parts
interface creates empty assembly
interface establishes connection between project and application
interface establishes connection between project and the intelligent part data in intelligent CAD vault
interface creates assembly by storing which parts form the assembly
3: use existing assembly
interface sends query for right assembly
the assembly presented by product structure is moved to interface
interface establishes connection of project with assembly
interface establishes connection of project with the intelligent part data of the parts in the intelligent CAD vault
4: change existing assembly with new modelled parts
interface copies the used assembly
interface transfers the connections between project and intelligent part data trhough new assembly
interface moves intelligent data of newly created parts in local CAD data base to intelligent CAD vault
interface connects the project through new assembly with the new intelligent data of the part
interface updates the assembly information
5: change existing assembly with existing parts
interface copies a used assembly
interface transfers the connections of project with intelligent part data from CAD vault through new assembly
interface establishes connection between project and intelligent data of parts that are added to the assembly from the intelligent CAD vault
interface updates the assembly information
6: magic wand design
interface sends query to product structure to find right assembly
interface moves assembly from product structure to interface
interface establishes connection between 3D neutral application and the 3D document in the assembly
finished
no
yes


act revision management
revision management
assembly is in use
by other project
change of
assembly
yes
no
construct activity
diagram 4 or 5 is
started.
copied assembly will be
new version or new
assembly
new version
new assembly
copied assembly becomes
new version of or
assembly
old assembly is kept
yes
no
changed or copied
assembly becomes
completely new assembly
construct activity diagram 1 or 2
withoud the creation of a new
assembly is started

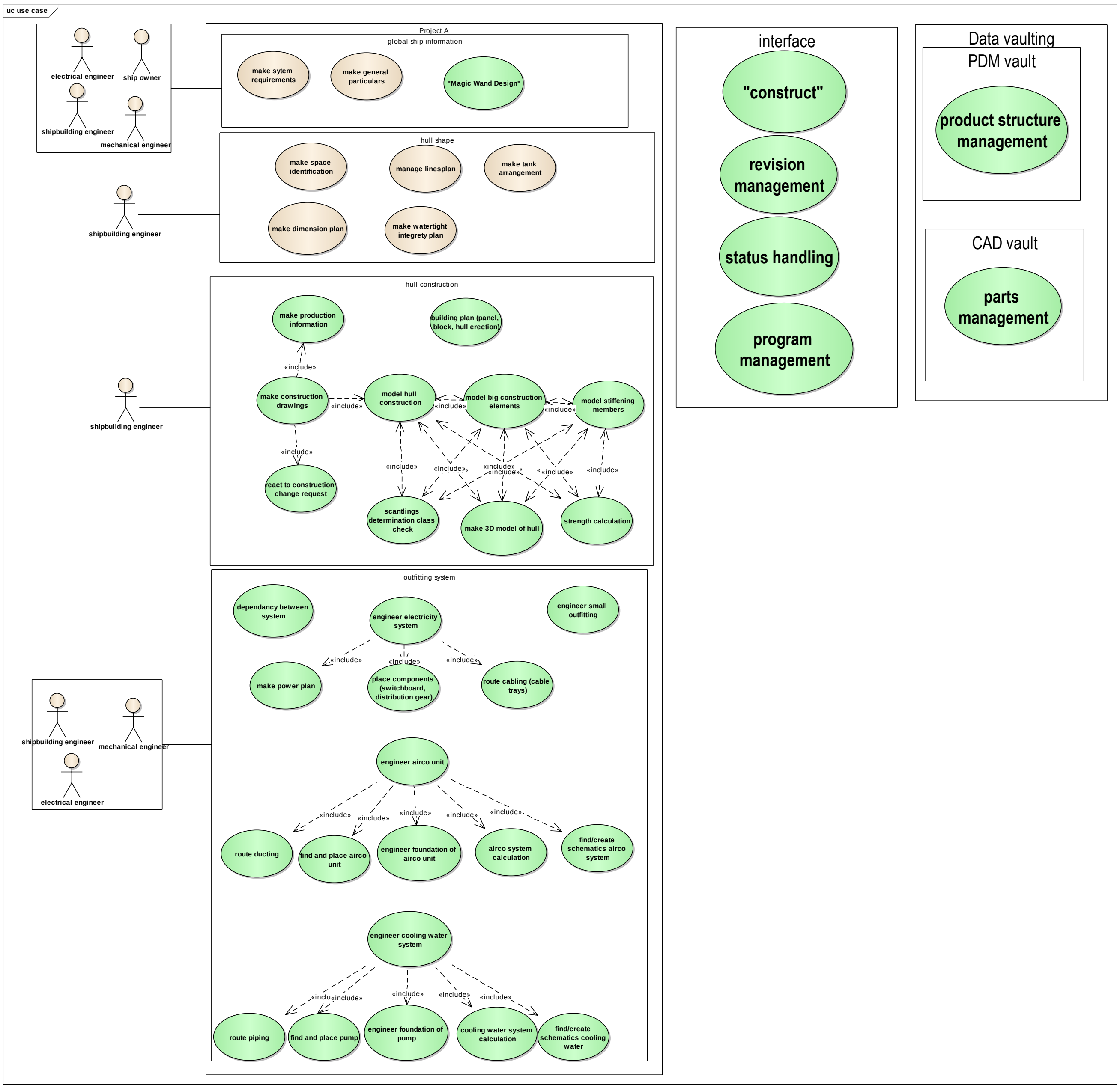

act status handling_ActivityGraph
engineer finds in program management an "unstarted" or "disapproved" work action (use case)
status handling
status of action is changed to "in progress"
engineer finishes his work action: change status to engineer finished
work action gets approved by respondible/accountible person
no
status of work action is set to "disapproved"
yes
hull or outfitting
outfitting
hull
Hull work action receives status hull-finished
outfitting work piece receives status: outfitting-finished
hull or hull combined with outfitting are checked
produced piece of ship reveives status: ready for production (hull and outfitting)


act program management_ActivityGraph
collect planning, resource, status information
collect status of the use cases from all the projects that are running
program management produces a priority list of activities
program management determines the activity that has highest priority AND is ready to be executed
program management makes the task available for the first qualified engineer that needs a new task


act product structure management_ActivityGraph
check for duplicate assemblies
sub-(sub-(sub)) assembly structure is made by product structure
there are equel sub-assemblies ?
yes
no
product structure removes double assemblies
the meta data with which the assemblies can be found is updated
query from interface about needing an assembly received
assembly is made available for connection and locked
request from interface about unlocking an unused assembly received
assembly is unlocked
request from interface to copy an assembly received
assembly is made ready to be copied
the copied assembly is locked


act parts management_ActivityGraph
1
query from interface to collect right intelligent data
intelligent part data is made available for project to connect to
2
request from interface to break the connection between application in project and intelligent data
connection between project and intelligent data is broken
3
inteface requests the transfer of intelligent part data from local CAD to CAD vault
new part is accepted and stored
4
query from interface to collect right intelligent data
request from interface to copy a part from CAD vault to local CAD
the part is copied and made available for interface to be placed in local CAD

# Diagram: use case



# Appendix B

# DEO VOLENTE: Conoship multi-purpose/heavylift design

Shipbuilder: ..............Conoship International BV, The Netherlands (Hull from OAG, Poland)
Vessel's name: ................................*Deo Volente*
Hull number: ..............................................001
IMO number: ........................................9391658
Owner/operator: .............Hartman Seatrade BV/ Amasus Shipping BV, The Netherlands
Designer: ..................Conoship International BV/ Hartman Marine BV, The Netherlands
Model test establishment used: ........MARIN BV, The Netherlands
Flag: ......................................The Netherlands
Total number of sister ships already completed: ............................Nil
Total number of sister ships still on order: .........................................5

CONOSHIP International was formed some 50 years ago to coordinate the design and marketing activities of the large group of, mainly coaster, shipbuilding companies then based in the northern part of Holland. Out of nearly 20 shipyards participating at that time, only three (plus four overseas 'associates') now remain, precipitating a change in direction by Conoship, which has transformed it into a consulting organisation undertaking concept design projects in conjunction with an owner, then sourcing a suitable builder worldwide.

*Deo Volente* was contracted in this way, with hull construction allocated to OAG, a Conoship subsidiary at Szczecin in Poland. Design engineering was handled in The Netherlands by Vuyk Engineering Groningen BV, and fitting out and commissioning of the vessel was also completed there.

An interesting claim made for the *Deo Volente* design is that it has produced 'the fastest cargo vessel of less than 3000gt in the world', with a service speed of 18knots derived from a machinery installation carried out by Wolfards & Wessels Werktuigbouw, and centred on a Wärtsilä package incorporating an 8L32 main engine developing 3680kW at 750rev/min, a 4000mm diameter CP propeller, and a SV 75-P48 gearbox. Electrical requirements are served from a 1600kW shaft-driven alternator running at 1500rev/min, and two Scania 455kW/1500rev/min diesel-driven sets. Also supplied is an HRP 300kW bow thruster with FP blades.

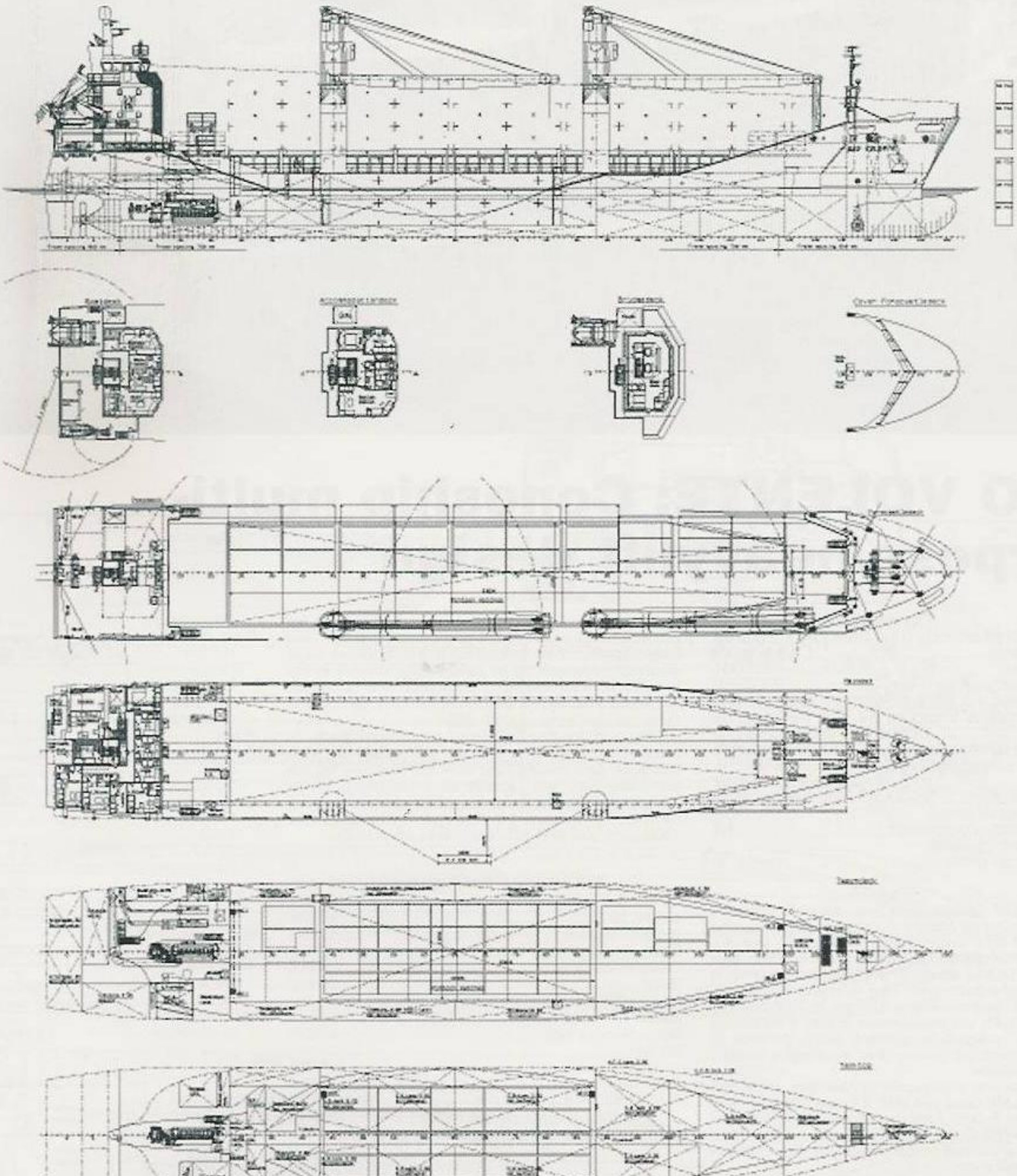
The slender hullform necessary to achieve this high performance has restricted space at the lower fore end of the single, 61.60m x 11.50m x 8.17m deep, double-skin cargo hold, and this area has been allocated to water ballast deep tanks. Nevertheless, the hold is fully serviceable for containers, general and bulk cargoes, project cargoes, and dangerous goods.

The upper deck has forecastle and poop erections, with the hold closed by pontoon hatches suitable for carrying heavy cargoes, and positioned on a high, full length trunk, in an arrangement which allows large diameter cable reels to be carried fully covered in the hold. Removable tweendeck pontoon covers can divide the space into upper and lower compartments, and these can also serve as grain divisions.

Tank top loading is 15tonnes/m², and cargo is handled using two large Liebherr CBB 120(81)/16(24) hydraulic deck cranes, built into the structure on the starboard side, and capable of a combined lift of 240tonnes. Automated control is possible using Liebherr's proprietary Litronic system. For lifts above 130tonnes, anti-heel stabilising tanks are available.

## TECHNICAL PARTICULARS

Length, oa ..........104.80m
Length, bp ..........98.20m
Breadth, moulded ..........15.60m
Depth, moulded to upper deck ..........7.40m
Draught, scantling ..........5.81m
Gross ..........2999gt
Deadweight ..........3500dwt
Speed, service ..........18knots
Cargo capacity, bale ..........4396m³
Bunkers
  heavy oil ..........420m³
Water ballast ..........2560m³
Fuel consumption, main engine ..........14tonnes/day
Classification ..........Bureau Veritas 1 3/3E, + Deepsea, + Mach, + AUT-MS, General Cargo/Container ship, Heavy Cargo (15tonnes/m²), Unrestricted Seagoing Service
Heel control equipment ..........Anti-heel tanks
Main engine
  Design ..........Wärtsilä
  Model ..........8L32
  Manufacturer ..........Wärtsilä
  Number ..........1
  Type of fuel used ..........HFO
  Output ..........3680kW/750rev/min
Gearbox
  Make ..........Wärtsilä
  Model ..........SV 75-P48
  Number ..........1
Propeller
  Designer/manufacturer ..........Wärtsilä/Lips
  Number ..........1 x high skew
  Pitch ..........Controllable
  Diameter ..........4000mm
Main engine-driven alternator
  Number ..........1
  Make/type ..........-
  Output/speed ..........1600kW/1500rev/min
Diesel-driven alternators
  Number ..........2
  Engine make ..........Scania
  Type of fuel used ..........HFO
  Alternator make/type ..........-
  Output/speed ..........2 x 455kW/1500rev/min
Boilers
  Number ..........1
  Make ..........-
  Type ..........hot water
Cargo cranes
  Number ..........2
  Make ..........Liebherr
  Type ..........CBB 120(81)/16(24)
  Performance, each crane ..........120tonnes/3.5-16m
  100tonnes/16-20m
  80tonnes/20-24m
  Tandem load 240tonnes (max)
Mooring equipment
  Number ..........1 x mooring winch/windlass
  1 x mooring winch
  Make ..........-
  Type ..........electro-hydraulic
Lifesaving equipment
  Number/type ..........1 x 14-person freefall lifeboat
  2 x 14-person liferaft
  1 x 6-person liferaft
  1 x 6-person man overboard boat
Hatch covers
  Designer/manufacturer ..........Rodenstaal BV
  Type ..........movable pontoons
Containers
  Total capacity ..........236TEU
  On deck ..........163TEU
  In holds ..........73TEU
Ballast pumps
  Number/type ..........2 x 250m³/h pumps
  2 x 50m³/h ejectors
  1 x 400m³/h heeling pump
Complement
  Officers ..........4
  Crew ..........3
  Pilot ..........1
Passengers ..........2
Bow thruster
  Make ..........HRP
  Number/type ..........1 x fixed pitch
  Output ..........350kW/1500rev/min
Contract date ..........-
Launch/float-out date ..........30 June 2006
Delivery date ..........1 February 2007



# PACIFIC ORCA: wind farm installation vessel

Shipbuilder: **Samsung Heavy Industries Co., Ltd. Geoje Shipyard, South Korea**
Vessel's name: ***Pacific Orca***
Hull No: **1940**
Owner/operator: **Swire Pacific Offshore Operations (Pte) Ltd., Singapore/Swire Blue Ocean A/S**
Country: **Denmark**
Designer: **Knud E. Hansen A/S**
Country: **Denmark**
Model test establishment: **Samsung Ship Model Basin, South Korea**
Flag: **Limassol, Cyprus**
IMO numbers: **9601326**

ON 27 July, 2012 Samsung Heavy Industries Co., Ltd Geoje Shipyard in South Korea delivered the first of two wind turbine installation vessels to Swire Pacific offshore Operations (Pte) Ltd, *Pacific Orca*. The second vessel, *Pacific Ofsprey*, was delivered 28 December 2012. This was the culmination of a contract that entered into force on 11 August, 2010. The two new vessels will be operated by the Danish daughter company Swire Blue Ocean A/S.

*Pacific Orca* and *Pacific Osprey* have been designed especially for the installation of offshore wind turbines and for support in the offshore oil and gas sector. The 161m long and 49m wide vessels, which are the largest of their kind, are equipped with six 105m long truss type legs and an electric rack-and-pinion jacking system. The six-legged design was chosen for the greatest safety and reliability under the most extreme weather and sea conditions while being jacked 17m above the sea surface on up to 60m water depth. Should 60m water depth not be enough the legs are designed so that they can be lengthened by further 15m.

The forward legs are closer together than the midship and aft legs to refine the hull lines in way of the shoulders and with a relatively long bow the vessels are designed to make good speed even in higher sea states, where similar vessels with blunter bows would be stopped.

The vessels are equipped with a diesel electric propulsion plant that features a DP-2 dynamic positioning system with four Azipod thrusters aft and two tunnel thrusters and two retractable azimuth thrusters in the bow.

With a cargo deck area of 4,300m$^2$ and a jackable deadweight of not less than 8,400tonnes, the vessels offer great flexibility in the carriage and installation of offshore wind turbine foundations of all types and sizes, and they are also ideal for decommissioning oil rigs.

The deck is served by two cranes a 1,200tonne main crane, which works around the aft leg in a starboard direction for a 360degs unobstructed rotation, and a 50tonne auxiliary crane, which is fitted on a cantilever on the jacking frame of the midship leg which also works in the starboard direction and has a rotation of 300degs. A knuckle-boom crane for loads up to 4tonnes and man-riding can be easily moved between two foundations; one forward and one aft of the main crane.

The accommodation block forward holds 111 single cabins all with en-suite bathrooms as well as the necessary crew facilities as messes and day rooms, offices and conference rooms etc. A helicopter landing deck for medium size helicopters is fitted above and forward of the accommodation block.

With their superior capacity and flexibility these new vessels are an important and timely innovation for the industry as it moves into deeper waters and more challenging operations.

## TECHNICAL PARTICULARS

Length oa
- Hull excl. helicopter deck: 161.3m
- Incl. helicopter deck: 164.9m

Length bp: 155.6m
Breadth, moulded: 49.0m
Depth to main deck, moulded: 10.4m
Draught, moulded
- Design: 5.5m
- Max. summer: 6.0m
- Air draught at design draught: 99.5m

Gross tonnage: 14,000gt
Lightweight: 24,390tonnes
Deadweight
- At design draught: 9,890dwt
- At max. summer draught: 13,155dwt
- For jacking: 8,400dwt

Block co-efficient: 0.78
Service speed: 13.0knots
Classification society and notations: GL 100 A5 Offshore Support Vessel Self-elevating Unit WTIS EP Helil SPS (except SRtP)
Tank capacities
- Marine gas oil: 4,285m$^3$
- Lube oil: 44m$^3$
- Fresh water – potable: 1,533m$^3$

Water ballast: 11,905m$^3$
Cargo deck
- Deck area: 4,300m$^2$

Uniformly distributed load
- Aft & amidships: 21tonnes/m$^2$
- Forward: 15tonnes/m$^2$
- Grid system of strong points: Mesh 1.4 x 1.4m
- Max strong point loads aft: 250tonnes downwards / 200tonnes upwards

Automatic anti-heeling system:
- Pump capacity: 2,000m$^3$ per hour
- Change of trim moment: 82,600tm per hour

Diesel generator sets
- Number of generator sets: 8
- Engine make/type: MAN L27/38
- Type of fuel: Marine gas oil
- Output: 720rpm
- Alternator make/type: ABB AMG 0710LS10 LSE
- Rated electrical power: 3024kW

Bow tunnel thrusters
- Number of thrusters: 2
- Make/type/capacity: Brunvoll FU100LTC2750, 2.2 MW

Bow retractable azimuth thrusters
- Number of thrusters: 2
- Make/type/capacity: Brunvoll AR100LNA2600, 2.2 MW

Stern thrusters
- Number of thrusters: 4
- Make/type/capacity: ABB Compact Azipod, 3.4 MW

Dynamic positioning system
- Type: DP-2

Legs and spud cans
- Number of legs: 6
- Type: 3-chorded truss type w. split-pipe-chords
- Length: 105m (may be lengthened by 15m)
- Max. leg protrusion below BL: 80m
- Chord distance: 9.7m
- Rack thickness: 6inch
- Spud can area: 95.4m$^2$

Jacking system
- Design and make: BLM
- Type: High-speed electrical rack-and-pinion
- Jacking units: 6 double-pinion D110 units per chord
- Jacking speed:
  - Raising / lowering legs: 2.4 m/min
  - Raising / lowering hull: 1.2 m/min

Operational conditions for jacking
- Wind speed: 20m/s
- Significant wave height: 2.5m (subject to actual conditions onsite)

Main crane
- Make: NOV Amclyde
- Type: Rope luffing "work-around-leg"
- Main hoists: 2 x 600t side by side for 1200t 31m in tandem
- Max. load-radius: 91m
- Aux hoist: 500tonnes 50m
- Max. load-radius: 107m
- Whip hoist: 50tonnes 112m, approved for man-riding
- Tuggers: 7 x 5tonnes SWL
- Max operational wind speed: 20m/s

Auxiliary crane
- Make: NOV Amclyde
- Type: Hydraulic
- Main hoist: 35tonnes 6.5 to 30m
- Aux hoist: 25tonnes 6.5 to 40m, approved for man-riding

Knuckle-boom crane
- Make: NOV Amclyde
- Type: Hydraulic with telescopic jib
- Hoist: 2tonnes 25m, 4tonnes 14m
- Man-riding radius: 30m by operating telescopic jib

Mooring equipment
- Make/type: RRM Electric MW 250F / CU 87 U3 /MW250F

Helicopter landing deck
- D-diameter: 22m
- Load-bearing capacity: 12.8tonnes

Life boats
- Number and capacity: 2 x 60persons
- Make/type: Norsafe JYN 80 with LH-140 davits

Integrated bridge control system
- Make: Samsung Automation SSAS-Master

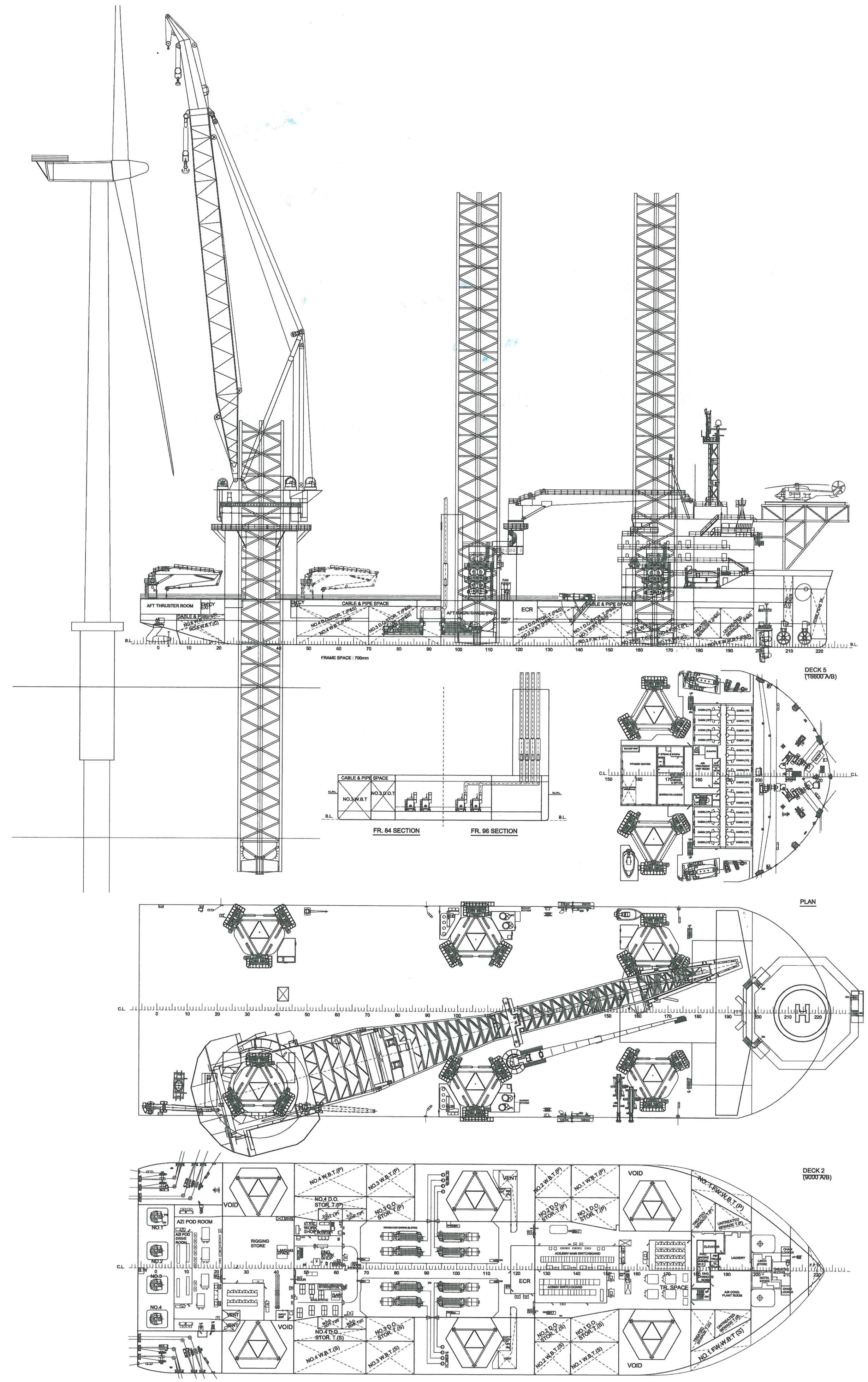
Complement
- Number of cabins: 111 with en-suite bathrooms

Fire detection system
- Make/type: Tyco T2000

Fire extinguishing system
- Engine room: Unitor 50-$CO_2$ HP system

Waste disposal plants
- Galley waste handling system: USON Marine
- Incinerator: Hyundai – Atlas / Maxi NG150SL WS
- Sewage plant: Omnipure / 5528

Contract date: 11 August 2010
Delivery date: 27 July 2012



# Appendix C

The tables below show the use case breakdown structures and how the amount of use cases divided over the specialisms is determined. The left column states the amount of use cases per specialism. The specialisms columns show is a certain use case belongs to a specialism. On the right the amount of use cases columns count the use cases per specialisms and are summarized on the bottom of that column.

| | | | | | | | | | | | | | amount of use cases | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| amount | DEO VOLENTE | | | | specialism | | | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | make system requirements | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | make general particulars | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make space identification | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | manage linesplan | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make tank arrangement | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make dimension plan | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make watertight integrety plan | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | | building plan | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | | strength calculation | | | | | | 5 | 6 | | 8 | | | | | 1 | 1 | | 1 |
| 1 | | | react to construction change request | | | | | | 5 | | | | | | | | 1 | | | |
| 15 | | | make production information | | | | | | 5 | | | | | | | | 15 | | | |
| 15 | | | make 3D model | | | | | | 5 | | | | | | | | 15 | | | |
| 15 | | | scantlings determination class check | | | | | | 5 | | | | | | | | 15 | | | |
| 15 | | | model hull construction | | | | | | 5 | | | | | | | | 15 | | | |
| 15 | | | model big construction elements | | | | | | 5 | | | | | | | | 15 | | | |
| 15 | | | model stiffening members | | | | | | 5 | | | | | | | | 15 | | | |
| 15 | | | make production information | | | | | | 5 | | | | | | | | 15 | | | |
| | | | | | | | | | | | | | | | | | | | | |
| 11 | | | | dependancy between systems | 1 | 2 | 3 | 4 | | | | 8 | 11 | 11 | 11 | 11 | | | | 11 |
| 15 | | | | engineer small outfitting | 1 | 2 | 3 | 4 | | | 7 | 8 | 15 | 15 | 15 | 15 | | | 15 | 15 |
| | | | main engine HFO | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place engine | 1 | | | | | | | | 1 | | | | | | | |
| 2 | | | | route HFO piping | 1 | | | | | | | | 2 | | | | | | | |
| 1 | | | | route luboil piping | 1 | | | | | | | | 1 | | | | | | | |
| 1 | | | | engineer foundation | 1 | | | | | | | | 1 | | | | | | | |
| 1 | | | | find/create schematics | 1 | | | | | | | | 1 | | | | | | | |
| 1 | | | | engine system calculation | 1 | | | | | | | | 1 | | | | | | | |
| 1 | | | | engineer gearbox | 1 | | | | | | | | 1 | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| | | | electricity generation | | | | | | | | | | | | | | | | | |
| 3 | | | | find and place alternator | | | | | | | | 8 | | | | | | | | 3 |
| 3 | | | | engineer foundation | | | | | | | | 8 | | | | | | | | 3 |
| 1 | | | | make power plan | | | | | | | | 8 | | | | | | | | 1 |
| 15 | | | | route cabling | | | | | | | | 8 | | | | | | | | 15 |
| 2 | | | | route piping HFO | | | | | | | | 8 | | | | | | | | 2 |
| 15 | | | | place switchboards and distribution gear | | | | | | | | 8 | | | | | | | | 15 |
| | | | | | | | | | | | | | | | | | | | | |
| | | | cooling water (no pumps mentioned in significant ships) | | | | | | | | | | | | | | | | | |
| 1 | | | | cooling water system calculation | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | find/create schematics cooling water system | | | | | | | | 8 | | | | | | | | 1 |
| 2 | | | | route piping | | | | | | | | 8 | | | | | | | | 2 |
| 1 | | | | find and place pumps | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | engineer foundation | | | | | | | | 8 | | | | | | | | 1 |
| | | | | | | | | | | | | | | | | | | | | |
| | | | boiler | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place boiler | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | engineer foundation | | | | | | | | 8 | | | | | | | | 1 |
| 2 | | | | route piping water/HFO | | | | | | | | 8 | | | | | | | | 2 |
| 1 | | | | boiler system calculation | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | find/create boiler schematics | | | | | | | | 8 | | | | | | | | 1 |
| | | | | | | | | | | | | | | | | | | | | |
| | | | ballast | | | | | | | | | | | | | | | | | |
| 2 | | | | find and place pumps | | | | | | 6 | | 8 | | | | | | 2 | | 2 |
| 2 | | | | find and place ejectors | | | | | | 6 | | 8 | | | | | | 2 | | 2 |
| 1 | | | | find andplace heeling pump | | | | | | 6 | | 8 | | | | | | 1 | | 1 |
| 5 | | | | engineer foundation | | | | | | | | 8 | | | | | | | | 5 |
| 15 | | | | route piping | | | | | | | | 8 | | | | | | | | 15 |
| 1 | | | | ballast system calculation | | | | | | 6 | | 8 | | | | | | 1 | | 1 |
| 1 | | | | find/create ballast water schematics | | | | | | 6 | | 8 | | | | | | 1 | | 1 |

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEO VOLENTE continuation | | | | bow thruster | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | find and place bow thruster | | | 1 | | | | | | | | 1 | | | | | | | |
| 1 | | | | | bow thruster calculation | | | 1 | | | | | | | | 1 | | | | | | | |
| 1 | | | | | engineer foundation | | | 1 | | | | | | | | 1 | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | cranes | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | find and place cranes | | | | | | | | | 7 | | | | | | | | 2 | |
| 2 | | | | | engineer foundation | | | | | | | | | 7 | | | | | | | | 2 | |
| 2 | | | | | find/create crane schematics | | | | | | | | | 7 | | | | | | | | 2 | |
| 2 | | | | | crane system calculation | | | | | | | | | 7 | | | | | | | | 2 | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | hatch covers | | | | | | | | | | | | | | | | | | | |
| 4 | | | | | find and place hatch covers | | | | | | | 5 | | 7 | | | | | | 4 | | 4 | |
| 4 | | | | | engineer foundation | | | | | | | 5 | | 7 | | | | | | 4 | | 4 | |
| 4 | | | | | hatch cover system calculation | | | | | | | 5 | | 7 | | | | | | 4 | | 4 | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | heating/cooling (HVAC) | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | find/create heating cooling schematics | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | | calculate heating/cooling system | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 2 | | | | | route ducting | | | | 2 | | | | | | 8 | | 2 | | | | | | 2 |
| 1 | | | | | find and place airco unit(s) | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | | engineer foundations | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | live saving | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | find and place live saveing equipment | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | | engineer foundations | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | | live saving calculations | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | | find/create life saving schematics | | | | | | | | | 7 | | | | | | | | 1 | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | mooring equipment | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | find and place mooring equipment | | | | | | 4 | | | 7 | | | | | 1 | | | 1 | |
| 1 | | | | | engineer foundations | | | | | | 4 | | | 7 | | | | | 1 | | | 1 | |
| 1 | | | | | mooring equipement calculations | | | | | | 4 | | | 7 | | | | | 1 | | | 1 | |
| 1 | | | | | find/create mooring equipment schematics | | | | | | 4 | | | 7 | | | | | 1 | | | 1 | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | hotel systems | | | | | | | | | | | | | | | | | | | |
| 1 | | | | | drinkwater | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | | sewage | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | | interior | | | | 2 | | | | | | | | 1 | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | specialisms | | | amount | % | | | | | | | | | 45 | 43 | 34 | 38 | 127 | 16 | 51 | 121 |
| | | 1 | main engine/gearboxes/thrust | | | 45 | 9% | | | | | | | | | | | | | | | | |
| | | 2 | accommodation | | | 43 | 9% | | | | | | | | | | | | | | | | |
| | | 3 | automation | | | 34 | 7% | | | | | | | | | | | | | | | | |
| | | 4 | hydraulics | | | 38 | 8% | | | | | | | | | | | | | | | | |
| | | 5 | hull/structure | | | 127 | 27% | | | | | | | | | | | | | | | | |
| | | 6 | hydrostatics | | | 16 | 3% | | | | | | | | | | | | | | | | |
| | | 7 | secondary steel, hatches, winc | | | 51 | 11% | | | | | | | | | | | | | | | | |
| | | 8 | mechanical engineering systen | | | 121 | 25% | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | 475 | 100% | | | | | | | | | | | | | | | | |

| amount | PACIFIC ORCA | | | | specialism | | | | | | | | aantal use cases | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | make system requirements | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | make general particulars | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make space identification | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | manage linesplan | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make tank arrangement | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make dimension plan | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | make watertight integrety plan | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | | building plan | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | | | strength calculation | | | | | | 5 | 6 | | 8 | | | | | 1 | 1 | | 1 |
| 1 | | | react to request | | | | | | 5 | | | | | | | | 1 | | | |
| 30 | | | make production information | | | | | | 5 | | | | | | | | 30 | | | |
| 30 | | | make 3D model | | | | | | 5 | | | | | | | | 30 | | | |
| 30 | | | scantlings determination class check | | | | | | 5 | | | | | | | | 30 | | | |
| 30 | | | model hull construction | | | | | | 5 | | | | | | | | 30 | | | |
| 30 | | | model big construction elements | | | | | | 5 | | | | | | | | 30 | | | |
| 30 | | | model stiffening members | | | | | | 5 | | | | | | | | 30 | | | |
| 30 | | | make production information | | | | | | 5 | | | | | | | | 30 | | | |
| | | | | | | | | | | | | | | | | | | | | |
| 17 | | | | dependancy between systems | 1 | 2 | 3 | 4 | | | | 8 | 17 | 17 | 17 | 17 | | | | 17 |
| 30 | | | | engineer small outfitting | 1 | 2 | 3 | 4 | | | 7 | 8 | 30 | 30 | 30 | 30 | | | 30 | 30 |
| | | | main engines MGO | | | | | | | | | | | | | | | | | |
| 8 | | | | find and place engine | 1 | | | | | | | | 8 | | | | | | | |
| 1 | | | | engine system calculation | 1 | | | | | | | | 1 | | | | | | | |
| 8 | | | | engineer foundation | 1 | | | | | | | | 8 | | | | | | | |
| 1 | | | | find/create schematics | 1 | | | | | | | | 1 | | | | | | | |
| 2 | | | | route MGO piping | 1 | | | | | | | | 2 | | | | | | | |
| 1 | | | | route luboil piping | 1 | | | | | | | | 1 | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| | | | electricity generation | | | | | | | | | | | | | | | | | |
| 1 | | | | make power plan | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | find/create schematics | | | | | | | | 8 | | | | | | | | 1 |
| 30 | | | | place swithcboards and distribution gear | | | | | | | | 8 | | | | | | | | 30 |
| | | | | | | | | | | | | | | | | | | | | |
| | | | cooling water (no pumps mentioned) | | | | | | | | | | | | | | | | | |
| 1 | | | | cooling water system calculation | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | find/create schematics cooling water system | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | find and place pumps | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | engineer foundation | | | | | | | | 8 | | | | | | | | 1 |
| 2 | | | | route piping | | | | | | | | 8 | | | | | | | | 2 |
| | | | | | | | | | | | | | | | | | | | | |
| | | | ballast (no pumps mentioned) | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place pumps | | | | | | 6 | | 8 | | | | | | 1 | | 1 |
| 1 | | | | ballast system calculation | | | | | | 6 | | 8 | | | | | | 1 | | 1 |
| 1 | | | | find/create ballast water schematics | | | | | | 6 | | 8 | | | | | | 1 | | 1 |
| 1 | | | | engineer foundation | | | | | | | | 8 | | | | | | | | 1 |
| 8 | | | | rout piping | | | | | | | | 8 | | | | | | | | 8 |
| | | | | | | | | | | | | | | | | | | | | |
| | | | bow thruster | | | | | | | | | | | | | | | | | |
| 2 | | | | find and place bow thruster | 1 | | | | | | | | 2 | | | | | | | |
| 1 | | | | bow thruster calculation | 1 | | | | | | | | 1 | | | | | | | |
| 2 | | | | engineer foundation | 1 | | | | | | | | 2 | | | | | | | |
| 1 | | | | find/create bow thruster schematics | 1 | | | | | | | | 1 | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| | | | bow azimuth | | | | | | | | | | | | | | | | | |
| 2 | | | | find and place thruster | 1 | | | | | | | | 2 | | | | | | | |
| 1 | | | | azimuth thruster calculation | 1 | | | | | | | | 1 | | | | | | | |
| 2 | | | | engineer foundation | 1 | | | | | | | | 2 | | | | | | | |
| 1 | | | | find/create azimuth schematics | 1 | | | | | | | | 1 | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PACIFIC ORCA continuation | | | stern azimuth | | | | | | | | | | | | | [redacted] | | | | | | | | |
| 4 | | | | find and place thruster | | | | 1 | | | | | | | | [redacted] | 4 | | | | | | | |
| 1 | | | | azimuth thruster calculation | | | | 1 | | | | | | | | [redacted] | 1 | | | | | | | |
| 4 | | | | engineer foundation | | | | 1 | | | | | | | | [redacted] | 4 | | | | | | | |
| 1 | | | | find/create azimuth schematics | | | | 1 | | | | | | | | [redacted] | 1 | | | | | | | |
| | | | | | | | | | | | | | | | | [redacted] | | | | | | | | |
| | | | jack up system | | | | | | | | | | | | | [redacted] | | | | | | | | |
| 6 | | | | find and place legs | | | | | | | | | | | 8 | [redacted] | | | | | | | | 6 |
| 1 | | | | calculation legs | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |
| 6 | | | | jacking units | | | | | | | | | | | 8 | [redacted] | | | | | | | | 6 |
| 1 | | | | jacking system calculation | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |
| 1 | | | | find/create jacking system schematics | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |
| 6 | | | | engineer jacking foundation | | | | | | | | | | | 8 | [redacted] | | | | | | | | 6 |
| | | | | | | | | | | | | | | | | [redacted] | | | | | | | | |
| | | | main crane | | | | | | | | | | | | | [redacted] | | | | | | | | |
| 1 | | | | find and place crane | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| 1 | | | | engineer foundation | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| 1 | | | | find/create crane schematics | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| 1 | | | | calculate crane system | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| | | | | | | | | | | | | | | | | [redacted] | | | | | | | | |
| | | | auxiliary crane | | | | | | | | | | | | | [redacted] | | | | | | | | |
| 1 | | | | find and place crane | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| 1 | | | | engineer foundation | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| 1 | | | | find/create crane schematics | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| 1 | | | | calculate crane system | | | | | | | | | | 7 | | [redacted] | | | | | | | 1 | |
| | | | | | | | | | | | | | | | | [redacted] | | | | | | | | |
| | | | knuckle boom crane | | | | | | | | | | | | | [redacted] | | | | | | | | |
| 2 | | | | find and place crane | | | | | | | | | | 7 | | [redacted] | | | | | | | 2 | |
| 2 | | | | engineer foundation | | | | | | | | | | 7 | | [redacted] | | | | | | | 2 | |
| 2 | | | | find/create crane schematics | | | | | | | | | | 7 | | [redacted] | | | | | | | 2 | |
| 2 | | | | calculate crane system | | | | | | | | | | 7 | | [redacted] | | | | | | | 2 | |
| | | | | | | | | | | | | | | | | [redacted] | | | | | | | | |
| | | | heating/cooling (HVAC) | | | | | | | | | | | | | [redacted] | | | | | | | | |
| 1 | | | | find/create heating cooling schematics | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |
| 1 | | | | calculate heating/cooling system | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |
| 5 | | | | route ducting | | | | | | | | | | | 8 | [redacted] | | | | | | | | 5 |
| 1 | | | | find and place HVAC unit(s) | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |
| 1 | | | | engineer foundation(s) | | | | | | | | | | | 8 | [redacted] | | | | | | | | 1 |

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PACIFIC ORCA continuation | | | live saving | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place live saveing equipment | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | engineer foundations | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | live saving calculations | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | find/place live daving schematics | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | engineer foundations | | | | | | | | | | 7 | | | | | | | | 1 | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | mooring equipment | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place mooring equipment | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | engineer foundations | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | mooring equipement calculations | | | | | | | | | | 7 | | | | | | | | 1 | |
| 1 | | | | find/create mooring equipment schematics | | | | | | | | | | 7 | | | | | | | | 1 | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | drinkwater | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place watermaker | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | find and place water pump | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | engineer drinkwater system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | calculate drinkwater system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | find/create schematics drinkwater system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 4 | | | | route piping | | | | | 2 | | | | | | 8 | | 4 | | | | | | 4 |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | sewage system | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place pumps | | | | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | engineer sewage system | | | | | | | | | | | 8 | | | | | | | | 1 |
| 1 | | | | calculate sewage system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | find/create sewage schematics | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 4 | | | | route piping | | | | | | | | | | | 8 | | | | | | | | 4 |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | heating cooling (HVAC hotel) | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | find and place HVAC units | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | engineer HVAC system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | calculate HVAC system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 1 | | | | find/create schematics HVAC system | | | | | 2 | | | | | | 8 | | 1 | | | | | | 1 |
| 4 | | | | route ducting | | | | | 2 | | | | | | 8 | | 4 | | | | | | 4 |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | interior | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | engineer electronical equipment | | | | | 2 | 3 | | | | | | | 4 | 4 | | | | | |
| 4 | | | | find/create schematics electronical equipment | | | | | 2 | 3 | | | | | | | 4 | 4 | | | | | |
| 4 | | | | find and place electronical equipment | | | | | 2 | 3 | | | | | | | 4 | 4 | | | | | |
| 4 | | | | calculate electrocnical equipemt | | | | | 2 | 3 | | | | | | | 4 | 4 | | | | | |
| 4 | | | | interior architecture | | | | | 2 | | | | | | | | 4 | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | specialisms | | | amount | % | | | | | | | | | 98 | 94 | 71 | 55 | 220 | 12 | 63 | 161 |
| | | 1 | main engine/gearboxes/thrusters | | | 98 | 13% | | | | | | | | | | | | | | | | |
| | | 2 | accommodation | | | 94 | 12% | | | | | | | | | | | | | | | | |
| | | 3 | automation | | | 71 | 9% | | | | | | | | | | | | | | | | |
| | | 4 | hydraulics | | | 55 | 7% | | | | | | | | | | | | | | | | |
| | | 5 | hull/structure | | | 220 | 28% | | | | | | | | | | | | | | | | |
| | | 6 | hydrostatics | | | 12 | 2% | | | | | | | | | | | | | | | | |
| | | 7 | secondary steel, hatches, windows, | | | 63 | 8% | | | | | | | | | | | | | | | | |
| | | 8 | mechanical engineering systems | | | 161 | 21% | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | 774 | 100% | | | | | | | | | | | | | | | | |