Graduate School Form 30
Updated 1/15/2015

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Weijie Zhang

Entitled

Implementation of Re-usable, Configurable Systems Engineering Model using Product Lifecycle Management Platform

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Hazim El-Mounayri	
Chair	
Jie Chen	
Dan Surber	
Shuning Li	

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Hazim El-Mounayri

Approved by: Sohel Anwar 7/28/2015

Head of the Departmental Graduate Program Date

IMPLEMENTATION OF RE-USABLE, CONFIGURABLE SYSTEMS ENGINEERING MODEL USING PRODUCT LIFECYCLE MANAGEMENT PLATFORM

A Thesis

Submitted to the Faculty

of

Purdue University

by

Weijie Zhang

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

August 2015

Purdue University

Indianapolis, Indiana

ACKNOWLEDGMENTS

I would like to send my graduate committee chair with deepest gratitude, Dr. Hazim El-Mounayri, for his great guidance, advice, supervision and supports during the entire research process. I also would like to thank my committee members, Dr. Jie Chen, Dr. Dan Surber, and Dr. Shuning Li for serving on the committee and giving me abundant suggestions for my thesis work.

I would like to extend my special thanks to my collaborator from ICTT System Sciences, Mr. Bill Schindel and Mr. Jason Sherey for generously providing Systematica$^{\text{TM}}$[1] Mapping Models, offering General Production Patterns, and assisting on Systems Engineering, Systematica Methodology, and Pattern Content. I am also grateful for meetings held with Mr. Schindel and Mr. Sherey.

I am grateful to my mentor, Dr. Shuning Li, for providing sufficient training and guidance on Siemens PLM software. Dr. Li shared with me her experience on using Siemens Teamcenter and constantly worked with me on trouble shooting the challenges in this research.

I acknowledge the IUPUI Mechanical Engineering Department for financial support in this research.

I would like to thank IPLI (Initiative for Product Lifecycle Innovation) and INCOSE (International Council on Systems Engineering) for providing professional material support and monthly roundtable meeting discussions.

[1]systematic is a trademark of System Sciences,LLC

TABLE OF CONTENTS

LIST OF FIGURES

SYMBOLS

$BMIDE$	Business Modeler Integrated Development Environment
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
$COTS$	Commercial off the Shelf
FEA	Finite Element Analysis
LOV	List Of Values
$MBSE$	Model-Based Systems Engineering
$PBSE$	Pattern-Based Systems Engineering
PDM	Product Data Management
PLM	Product Lifecycle Management
$S * Metamodel$	Systematica Metamodel
$S * Mapping$	Systematica Mapping
$S * Models$	Systematica Models
$S * Patterns$	Systematica Patterns
$SysML$	Systems Modeling Language

ABSTRACT

Zhang, Weijie. M.S.M.E., Purdue University, August 2015. Implementation of Reusable, Configurable Systems Engineering Model using Product Lifecycle Management Platform. Major Professor: Hazim El-Mounayri.

Industry is facing the challenge of increasing product complexity while at the same time reducing cost and time in a highly competitive global market. Product Lifecycle Management (PLM) and Systems Engineering have the potential to help companies avoid costly product development and launching, as well as failure during use; these two concepts not only share many common characteristics, but also complement each other. PLM provides an information management system that can seamlessly integrate enterprise data, business processes, business systems and, ultimately, people throughout all phases of the product lifecycle. Systems engineering is an interdisciplinary approach to designing, implementing, evaluating, and managing the complex human-made systems over their life cycle. The same underlying methods that improve management of products and services can be used to organize the framework in which PLM systems are implemented, integrated, and evolved. Though several studies have indicated that adopting Systems Engineering with PLM brings many benefits for industries, implementation of PLM based Systems Engineering with PLM has rarely been conducted.

Pattern-Based Systems Engineering (PBSE), a form of Model-Based Systems Engineering (MBSE) based on the use of Systematic Metamodel (S* Metamodel), represents a family of manufacturing system, and is used in the life cycle processes of ISO 15288, was implemented here using TEAMCENTER®PLM software as the platform. More specifically, we have implemented the key portion of the General Production Pattern based on S* Metamodel, and demonstrated the benefit through the manufacturing of oil filter case study. The above implementation have resulted in a powerful systems engineering model in PLM that leverages the capabilities of Teamcenter, to enable an enhanced systems engineering approach. Benefits brought to systems engineering practice include: the ability to capture and reflect stakeholders' requirements and changes in product design process promptly and accurately; the ability of systems engineers to create models quickly and prevent mistakes during modeling; the ability of systems engineers to do their job much easily by using reusable and reconfigurable models; the ability to re-use of previous designs in a new process.

1. INTRODUCTION

1.1 Background

1.1.1 Product Lifecycle Management

Currently much attention is being placed on PLM in many industries, such as the automotive, pharmaceutical, aerospace, and military weapons industries. These industries share similar expectations for enhancing their competitive ability in a globalized economic environment. The impact of this globalization brings many positive effects and opportunities, but it also requires that global companies meet new international and domestic challenges. As a result of the increased pressure to both reduce cost and create high technology products rapidly, business managers are looking for an appropriate tool or strategy to help monitor research and development for their products, and help management teams make right decisions at the right time [1]. This task can be accomplished by using the PLM method.

The product lifecycle includes the following steps: the requirements and planning of a product, the concept and design, the manufacturing production, the sales and distribution, and managing the product throughout its operational life until disposal and recycling [2]. In the product lifecycle, with the passage of each stage and process, products undergo a series of changes in functionality, performance, technical process, materials, supply, marketing, and other aspects of social resources [3]. The process produces a large amount of complex technical and business information. Under the pressure of globalization from market competition, product life cycles are being shortened, and new products must get to market faster. The only solution to keep products competitive is to quickly and correctly manage and use this information [3]. So, it is significant for an enterprise to rely on information to accelerate the process of enterprises to develop better products.

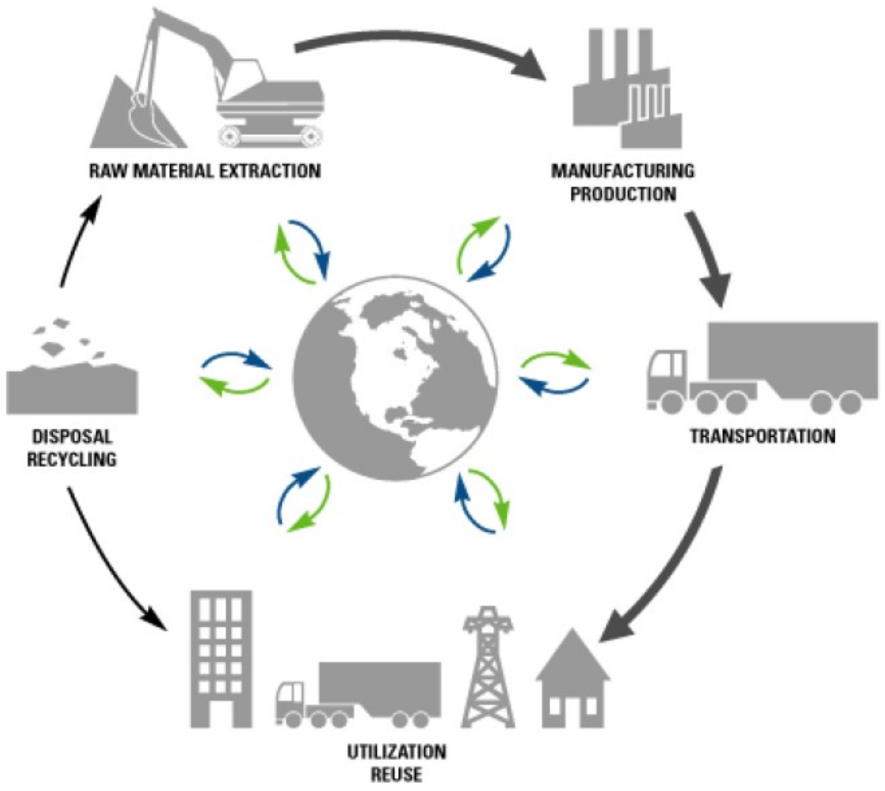


Figure 1.1. Generic lifecycle of product [4]

PLM is a concept with multiple interpretations, and the most common definition is “an information management system which allows the enterprise to integrate data, to integrate process, to integrate system, and to integrate people across the product lifecycle” [5]. PLM is also an advanced enterprise information strategy [6]. It makes people think about how to use the most effective methods to increase revenue and reduce costs. An efficient and comprehensive PLM solution enables companies to establish detailed, intuitive, and viable digital product information. It allows the collection of early comprehensive information from each participant and then enables the discovery and resolution of critical issues.

PLM was developed on the basis of Product Data Management(PDM), which is a file-based system to manage production data. PLM is database system used to manage processes [7]. PLM also refers to a type of software and process that uses

Internet technology to make everyone involved in the entire life cycle of products cooperate in product development, manufacturing, and sales management.

Management (CRM), and Enterprise Resource Planning (ERP), are currently applied widely. PLM is an information system that shares and leverages information across all systems. This bi-directional information exchange between PLM and other systems is important to facilitate information flow between multi-functional groups [8].

1.1.2 Systems Engineering

According to the research, "by using the Systems Engineering approach, project costs and timescales are managed and controlled more effectively by having greater control and awareness of the project requirements, interfaces and issues and the consequences of any changes. Research also indicates that effective use of systems engineering can save 10-20 percent or more of the project budget. [9]"

What is systems engineering? Systems engineering is an interdisciplinary approach to design, implement, evaluate, and manage the complex human-made systems over their life cycle [10]. It focuses on defining customer needs, functionality, and documenting requirements in the early stages of the development cycle, and then proceeds with design synthesis, allocation, evaluation, operation, and system validation [11]. The driving force pushing industries into adopting systems engineering is the increasing complexity of systems. Many systems cannot meet the needs of stakeholders in terms of development time, overall cost, and performance. The benefits of applying systems engineering concepts are: reduction in the total cost of ownership during the system's lifecycle, reduction in system acquisition time, and the reduction of risks within the product development phase of the system.

Systems can be classified as natural or human-made. Human-made systems are functional groups of hardware, software, and human interface that work together to meet the mission need [12]. In many cases, a system is not independent. If one system consists of two or more hierarchical levels, lower levels are conveniently called subsystems. A system-of-systems is a group of dissimilar systems that collaborate to achieve a mission purpose which none of them can perform alone. Building the right system and building the system right is the top priority for a product development team, and can be enabled through use of a system engineer. With the purpose of this goal, a system engineer has the responsibility to ensure that the development team makes decisions that ensure the stakeholders' needs are satisfied in a high quality, trustworthy, cost efficient, and schedule compliant manner throughout a system's entire life cycle.

Generally speaking, systems engineering is the high level view of a project. To achieve success in applying systems engineering, there are tasks that must be performed in a process. The first and foremost is stating the problem/understanding the problem. The problem should identify stakeholder and system boundaries, describe a mission statement, and provide a concept of operations or a description of the deficiency, etc. It is important to understand the problem by thinking about what must be done and how it must be accomplished [10].

The second task is investigating alternatives, especially for complex systems. Since no design can economically achieve "best" on all its figures of merit, investigating alternative designs must be redone while coping with the increasing amounts of design and analysis data [10]. For example, figures of merit should be computed in designing, model constructing, data analyzing, and prototype building processes. Alternative designs will reduce the risk and also help clarify the problem statement. Besides these, modeling the system is another task that has to be accomplished.

Systems engineers create many types of systems models, such as block diagrams, functional flow diagrams, etc., based on the best designs to help manage the systems development. Integration means bringing systems, people, and other interactions together. Launching the system, assessing performance, and re-evaluating are also very important tasks that are necessary for systems engineering.

Over the past few decades, several process models in systems engineering have become well known: “waterfall”, “Vee”, and “Spiral” [12]. Those system life cycle process models specify a series of steps to reflect systems engineering approach. The “Vee” diagram is the most famous and has been applied in many industries. It is no surprise that different “Vee” models have been seen, but they all derive from the same basic model.

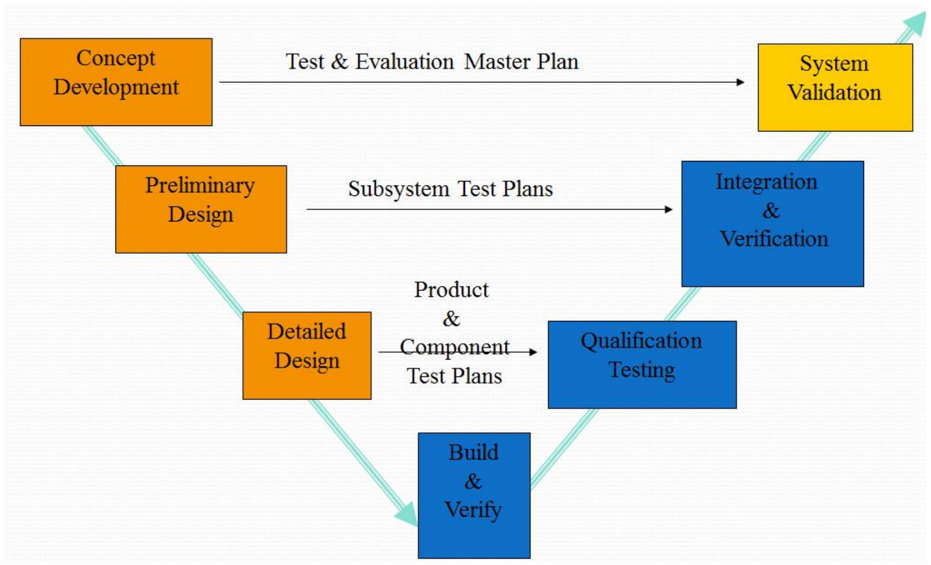


Figure 1.2. Systems engineering Vee diagram

The left side of the diagram demonstrates the decomposition and definition sequence; it resolves the architecture of a system and creates design details. The right side of the model illustrates the integration and verification sequence [12]. In it successively higher levels of subsystems are verified, with the sequence flowing up to the culminating system level. In the middle of the model, at each level of testing plan, requirements and specifications documents ensure that products, components, subsystems, and system meet all requirements and specifications.

To standardize these processes, ISO15288 was published by International Organization for Standardization. It is a common framework defining processes and terminology for describing life cycle of systems. ISO15288 can be applied to manage and perform stages through the full system lifecycle by selecting sets of process [13]. In ISO15288, four groups of systems engineering processes are organized: Agreement, Enterprise, Project, and Technical [14]. Each process contains a purpose, activities, and outcomes. Concept, development, and production are some examples of stages in the life cycle that have been described in ISO5288. A critical part of this international standard is that it will not conflict with any organization's policy or procedure. This is due to each life cycle having no definitive order for use. Systems differ in their purpose, applications, domain, compliancy, time, location, and size etc., but ISO15288 describes the processes that comprise the development of any man-made system in a repeatable life cycle.

1.2 Literature Review

In the past few years, systems engineering and PLM have become closely related. Systems engineering and PLM not only share many common characteristics, but also complement each other [15]. The product life cycle is covered through preliminary design, detail design, production or construction, product utilization, support phase-out, and disposal; those phases are based on understanding the application of systems engineering [16]. PLM requires management of the entire product process; it must meet the challenge of synchronizing disciplines involved in complex product systems during the production process. Systems engineering methodologies provide ways to synchronize disciplines during design, simulation, testing, verification, and validation based on multidisciplinary functions for an industrial company [15].

Currently, mass customization, small lot sizes, high variability of product types, and a changing product portfolio are characteristics of modern manufacturing systems during the life cycle [17]. A direct consequence of these characteristics is a more complex manufacturing system. This problem is especially serious in the aerospace and defense industry. According to a recent study of Government Accountability Office, growth in research and development costs and months' delay in delivering have been reported by defense acquisition programs [17]. In fact, due to complex manufacturing processes, most issues happening in production can be traced back to early architecture decisions, and those decisions directly or indirectly affect the efficiency of design. Figure 1.4 shows that 70 percent of cost is locked in the design concept and it is more expensive to make changes during later processes.

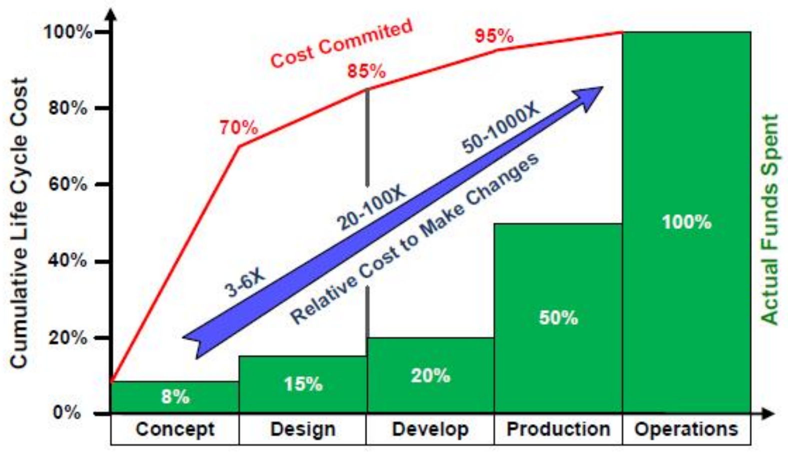


Figure 1.3. Life cycle cost commitment as a function of the system life cycle [15]

Systems engineering principles can alleviate manufacturing risks of serious challenges and issues. Involving systems engineering methodology is important to make adjustments in the manufacturing work flow process. Currently, many systems engineering methodologies have been applied to help make changes in manufacturing. One such method is called product and process development, integrated by Georgia Institute of Technology's aerospace system design laboratory [17]. This methodology comprises manufacturing process capability indices early in the design phase to ensure robust design concepts are being developed. In addition, "Manufacturing Systems Engineering" [18], "Factory Physics" [19], and "The Toyota Way: 14 Management Principles from the Worlds Greatest Manufacturer" [20] explore systems engineering methods to improve manufacturing process. Even though the traditional systems engineering methodology improves design process, many flaws still need to be fixed. The traditional document-centric, text-based report and drawing format in traditional systems engineering is not flexible. Under this high-flux design environment, it requires a higher level of methodology to assist systems engineers to do their job [21, 22].

One should not think of systems engineering without thinking of Model-based Systems Engineering (MBSE). MBSE is shifting the design approach from document-centric to model-based systems engineering by capturing information elements and relationships to support requirement, design, analysis, verification, and validation in life cycle. Compared to the traditional manufacturing process, applying MBSE will improve the communication among stakeholders, making it easier to manage systems in order to improve product quality, and improve the ability to learn the concepts of system. Currently, many MBSE methodologies are used by systems engineering communities: IBM Rational Harmony for Systems Engineering (Hoffmann 2011); INCOSE Object-Oriented Systems Engineering Method (INCOSE 2008) [13]; and Systematica Methodology for Pattern-Based Systems Engineering (ICTT System Sciences) [23–27].

As mentioned above, the general development of MBSE methodologies is helping systems engineers do a better job of adapting to the accelerated life cycles in product development. But, due to the increasing complexity of production and manufacturing processes and advancing in the computer-aided design technology, the traditional systems engineering tools no longer have enough capabilities to help systems engineers manage processes. The traditional systems engineering tools can be divided into three categories, which are system design tools, systems analysis tools, and systems control tools [28]. In each category, many methods and tools are involved to assist systems engineers in different processes. Most of the time, systems engineers use more than one system methods and tools to manage projects. For example, one of MBSE methodologies, Object-Oriented Systems Engineering Method (OOSEM), can be provided by SysML tools and associated with configuration management tools, performance modeling tools, and verification tools [29]. With the increased diversity of system tools, it is difficult for systems engineers to integrate tools and at the same time track the changes during processes.

MBSE methodology emphasis capturing elements and relationships in model and reusing then across multiple diagrams [13]. Most of the system tools do not provide a database for reusing legacy data in life cycle process.

Currently, researchers are trying to integrate PLM and systems engineering to overcome the weaknesses of traditional systems engineering tools and achieve the benefits of using PLM software; for instance, one company, called InterCax, have created a concept to build a bridge between MBSE and PLM. This concept called System Lifecycle Management (SLIM), which is deployed in Systems Modeling Language (SysML) environment, uses PLM software for specialization and configuration control [30]. This SLIM allows systems engineers to work directly in an SysML environment, in which the modeling language is most familiar to systems engineers. It also addresses the weaknesses of SysML. Another study conducted by Georgia Institution of Technology attempts to reduce cost and time by implementing a digitized systems engineering process into a PLM software [1]. InterCax [30] requires systems engineers to work in SysML environment instead of working in PLM environment; also, it does not use the full capabilities of PLM. On the other hand, the work by Georgia Institution of Technology does not include MBSE models, systems engineering implementation of Integrated Product and Process Design method in PLM software is used. The current work addresses the integration of PLM with systems engineering models to address the current limitations and advance the systems engineering practice. More specifically, we propose a new methodology for integrating Teamcenter PLM (an industry standard tool) with an advanced MBSE model, namely s*pattern [23–27]. This is meant to achieve the full potential of implementing "a re-usable and configurable" systems engineering model on a powerful PLM platform that supports a comprehensive set of tools and functionalities.

1.3 Thesis Objectives and Contributions

Implementing a MBSE model in Teamcenter PLM and then using the latters capabilities to improve systems engineering process is our approach to advance the systems engineering practice. In order to achieve this, a strong MBSE model and process that supports PLM systems engineering application and integration is needed; systematica methodology for PBSE has shown more advantages than other MBSE methods; PBSE has MBSE capabilities but simplifies the introduction of MBSE methods; it also reduces the recurring cost of modeling and shifting from “learn how to model to “learn the model. On the other hand, a strong PLM allowing integrating systems engineering model in its application tool is also required. Teamcenter provides a systems engineering application to integrate MBSE method in it.

In order to implement the merging of systems engineering with PLM in practice, the objectives of this research were as follows:

- Identify a targeted portion which is initially focused on Feature-Interactions-Functional Roles- Physical Systems trace of the General Production Pattern in S* Metamodel [31], and implement it into Teamcenter data schema.
- Generate a specialized system from using General Production Pattern in a specific application.

The long-term goal of this research is implementing the entire General Production Pattern in S*Metamodel into Siemens Teamcenter Systems Engineering data schema.

1.4 Thesis Outline

This thesis focuses on moving the key portion of General Production Pattern, which is a S*Pattern data structure compatible with the S*Metamodel, into a Siemens Teamcenter Systems Engineering data schema [21]. In Chapter 2, the details about the concept of MBSE and PBSE, the methodology of the S*model and the S*Metamodel are presented. In addition, the methodology of S*Patterns is presented. Furthermore, the data model configuring tool, Business Modeler Integrated Development Environment (BMIDE), is introduced. Last, the functionalities of Siemens Teamcenter software and systems engineering application are illustrated.

In Chapter 3 the procedure of creating and configuring S* mapping classes in BMIDE are presented. Bill of materials (BOM) views for Feature Framework, Interaction Framework, and System Environment are created in the Teamcenter systems engineering application environment. The relationships between different classes are also created. Teamcenter is merged with Microsoft Visio 2010 professional to generate diagrams for reviews. An oil filter S* Model specialized from General Production Pattern is applied to demonstrate the use of the General Production Pattern in Teamcenter platform.

Chapter 4 discusses the results of this thesis.

In Chapters 5 and 6 conclusions and future work are presented.

2. METHODOLOGY

2.1 MBSE/PBSE and S* Metamodel

ICTT System Sciences, a systems engineering company devoted to solving complex systems problems for enterprises, institutions and industries, has implemented Systematica systems engineering methodology [23] which is a specific MBSE methodology, through the use of S* Model. Leveraging the power of MBSE, a PBSE methodology, which is re-usable, configurable S* Models based on the use of S*Metamodel [31],is created. Re-usable, configurable S*Models are also called S* Patterns [23, 24, 27]. As mentioned earlier, the International Council of Systems Engineering defines MBSE a "formalized application of modeling to support system requirements, design, analysis, verification and validation activities in the system development life cycle" [32]. MBSE is a paradigm that emphasizes the common visual modeling principles' application and best practice involved in systems engineering activities. One of the visual modeling languages is SysML, which is in response to Unified Modeling Language (UML) for systems engineering [33]. The S* Model is an MBSE model which is based on the S* Metamodel. The practical S* Models are not limited in whatever modeling languages and tools are represented. Throughout the development of MBSE methodologies, the vast number of MBSE methodologies and system representation standards have proved that many elements are needed to build the "smallest model" framework of the S* Metamodel [24]. This is why S* Metamodel as the smallest set of information sufficient describes a system for systems engineering purpose, in any modeling language.

The Figure 2.1 depict key element of the S* Metamodel, where different colors represent different related classes [23]. In this research, the trace of Feature, Functional Interaction, Functional Role, and Design Component is being focused.

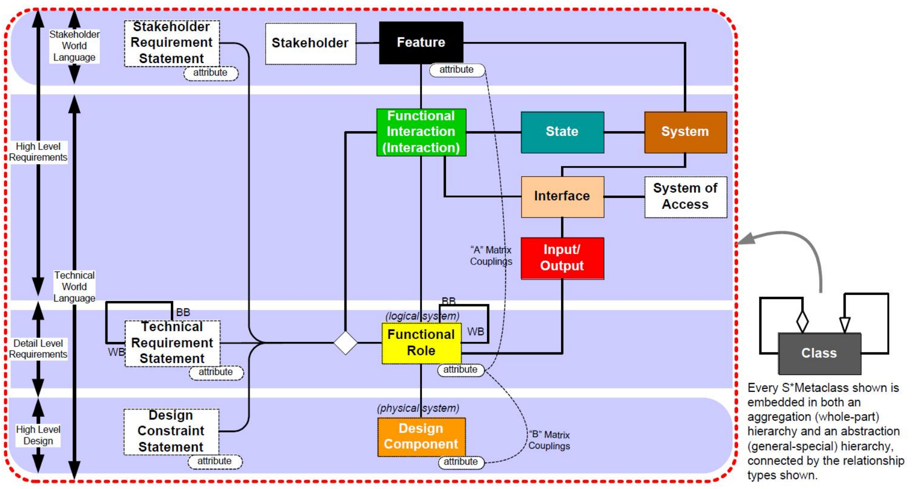

Stakeholder World Language
Stakeholder Requirement Statement
attribute
Stakeholder
Feature
attribute
High Level Requirements
Functional Interaction (Interaction)
State
System
Interface
System of Access
"A" Matrix Couplings
Input/ Output
Technical World Language
Detail Level Requirements
BB
WB
Technical Requirement Statement
attribute
(logical system)
Functional Role
BB
WB
attribute
High Level Design
Design Constraint Statement
attribute
(physical system)
Design Component
attribute
"B" Matrix Couplings
Class
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

Figure 2.1. Summary of some of the key portions of S* Metamodel [23]

Understanding the performance of each element in a system and interactions among elements is necessary to understand the whole system. As long as the increasing complexity of systems, stakeholders have different performance measures. It is necessary to understand the needs and performance measures of all stakeholders [28]. Features are packages of behavior or performance of a system that have stakeholder value [23]. S* Models are aimed at covering all the stakeholders not only just users or customers. Feature attributes are features' parameters that express stakeholder valuations in stakeholder language. For example, the Cruise Control Feature has feature attributes in fuel economy and speed variation. Because Features and Feature Attributes cover all stakeholders' value and interest, they affect all the design decisions, trade-offs, and optimization should be made in accordance. In the General Production Pattern Metamodel, selectable system features are described by the S* Metamodel. When the S* Patterns are used in specific product specialization and configuration, feature selection must obey one or few stakeholder values. Those selectable features contain: system delivery, compatibility, production capability, reliability and availability, operability, maintainability, configurability, securability, accountability, integrity, product containment, product protection, regulatory compliance, and health and safety. Those feature models can be categorized by different stakeholders, for instance, system management functions focus on operability, maintainability, configurability, securability, and accountability.

Learned from traditional historical views, lots of potential problems occurred due to ignoring interactions in the system. If go back to take a look at all the physical science, and mathematics, whether Newton's Law or Maxwell's equation, all the physical laws describe interactions [25]. Each engineering discipline (ME, EE, ChE, etc.) is built upon those laws [23, 25]. Since systems engineering is an interdisciplinary field, has to respect the physical laws and bring interaction to the front. In a system, there always exist interactions between components; an interaction means exchange of energy, force, mass or information [25] which leads to change of state. The Systematica calls that interaction a functional interaction, and each component plays a functional role in that interaction. MBSE helps make interactions explicit; this is the reason to emphasizes functional interaction as a fundamentally coordinating class relating other information in the S* Metamodel [25]. In the general production pattern S* Metamodel, the interaction class consists the following interaction actions: consume utility, control operation, coordinate production, deliver system, detect faults, maintain system, manage configuration data, manage electronic access, manage fault and maintenance data and alarms, manage maintenance safety procedures, manage physical access, manage safety procedures and interlocks, manage security data, monitor product quality, operate system, perform configuration procedures, perform maintenance procedures, protect, provide interface, remove scrap, remove system, secure system, stage material, configure system, account for system, and transport material. Those interactions support defined features in a system.

The functional role is also called logical system in S* Metamodel. Functional Roles are described by their behavior, and the role attributes are parameters of functional role which have technical valuations [34]. Operator, operator (level N+1), manufacturing management system, manufacturing management system (Level N+1), manufacturing system, manufacturing system (level N-1), combined managed system, managed subsystem, direct management system, material in process, occupant system, air in process, environment, manufacturing system of access, support and isolation system, support and isolation system (level N-1), utility service, utility service(level N-1), material service, and material service (level N-1) are included in the Logical Systems class in General Production Pattern.

The methodologies in systems engineering are concerned with both the engineering process and the information activities during the process [23]. Compared with traditional systems engineering, PBSE refers using S* Patterns concentrates on enhancing information involvement and relationships passing through the systems' process. In general, PBSE is built on MBSE models in patterns and uses a powerful MBSE Metamodel to describe systemic phenomena. There are many advantages to applying the PBSE approach, such as reducing the cost and time that shifting from the "learn how to model" to the "learn the model", such as being compatible with multiple modeling language standards, generating configured systems from models rapidly, etc.

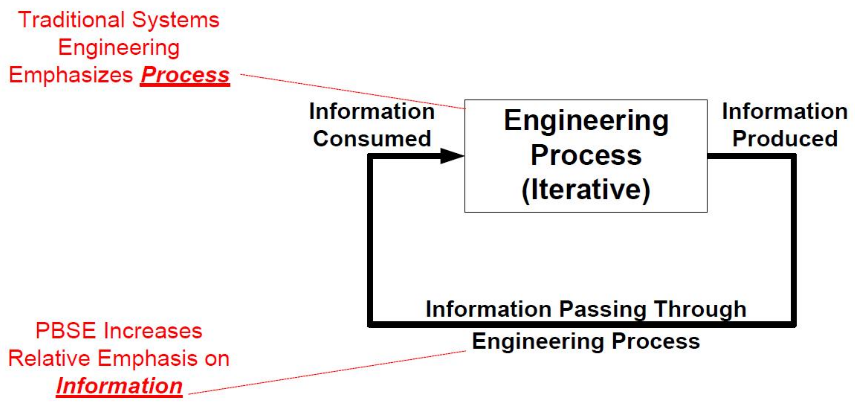

Traditional Systems Engineering Emphasizes ***Process***
Information Consumed
Engineering Process (Iterative)
Information Produced
Information Passing Through Engineering Process
PBSE Increases Relative Emphasis on ***Information***

Figure 2.2. The engineering process consumes and products information [23]

S* Patterns are re-usable, configurable S*Models [23,24,26], based on Systematica methodology for PBSE. An S*Pattern may be used and re-used across different system product lines, system families or systems configurations [23]. Once an S* Pattern has been applied in a specific enterprise or product line, it is very easy to quickly generate an S* Model from pattern instead of creating a new model in a new project.

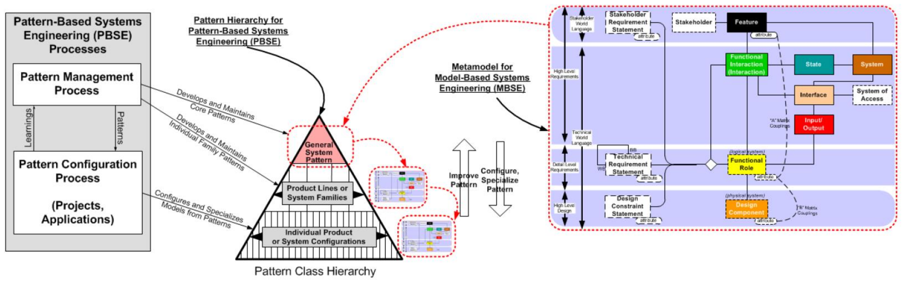

Pattern-Based Systems Engineering (PBSE) Processes
Pattern Management Process
Learnings
Patterns
Pattern Configuration Process
(Projects, Applications)
Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
Develops and Maintains Core Patterns
Develops and Maintains Individual Family Patterns
Configures and Specializes Models from Patterns
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
Pattern Class Hierarchy
Improve Pattern
Configure, Specialize Pattern
Metamodel for Model-Based Systems Engineering (MBSE)
Stakeholder World Language
High Level Requirements
Technical World Language
Detail Level Requirements
High Level Design
Stakeholder Requirement Statement
attribute
Stakeholder
Feature
Functional Interaction (Interaction)
State
System
Interface
System of Access
Input/ Output
"A" Matrix Couplings
BB
WB
Technical Requirement Statement
(logical system)
Functional Role
(physical system)
Design Constraint Statement
Design Component
"B" Matrix Couplings

Figure 2.3. S* Patterns are re-usable, configurable S* Models [23]

As depicted in the above figure shows, PBSE involves two processes involved: (1) The Pattern Management Process in a general system pattern, product lines, or system families' levels, (2) The Pattern Configuration Process in an individual production level [26]. The Pattern Management Process generates the underlying family model and updates the model based on project discovery and learning at the same time [26]. The Pattern Configuration Process makes configurations from upper level pattern use on specific projects. The S* Metamodel in the figure above is a Metamodel to create the General Production Pattern. This is the S* pattern have implemented in the Teamcenter platform, as described in the next chapter.

2.2 Business Modeler Integrated Devolvement Environment

BMIDE stands for Business Modeler Integrated Development Environment, it is a tool used for configuring the data model of the Teamcenter installation. Using the BMIDE function is allowed to configure the data model with new business objects, classes, and properties. BMIDE interface contain two perspectives; standard perspectives and advanced perspectives. In this research, the standard perspective is used due to its simplification and contains all the views as needed.

Using BMIDE allows creating data model objects, this included business objects, classes, properties, constants, and document management objects, lists of values, options, and rules. The specific objects created in this research will be described later.

2.3 PLM Software: Teamcenter

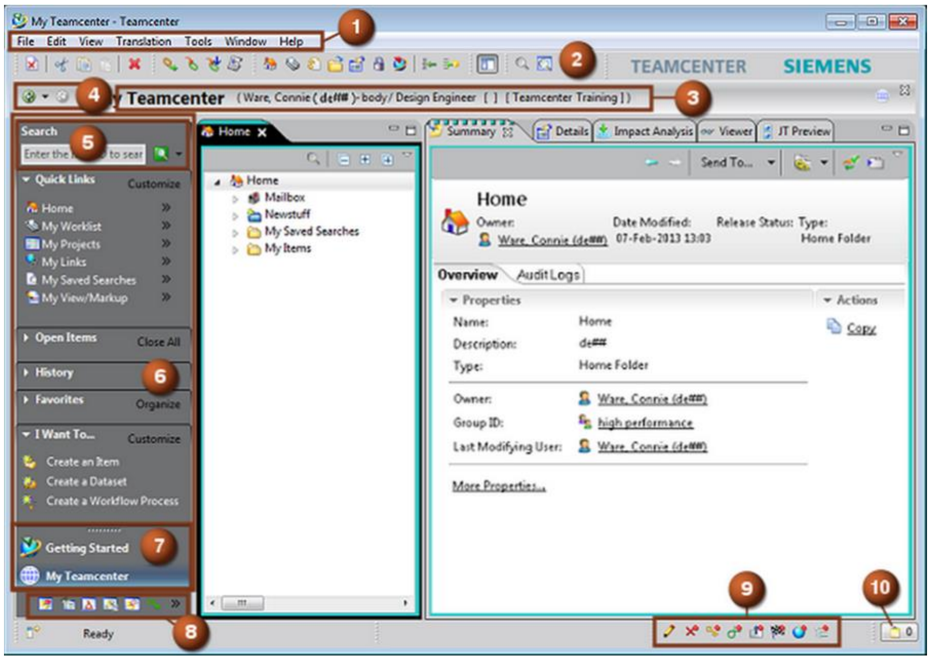
Under economic globalization pressure and highly competitive markets, PLM is necessary lead companies to develop and deliver better products. PLM systems help those companies make smarter decisions by providing decision makers with the right information.

Siemens, as one of the pioneer multinational conglomerate companies in the world, has involved in automation, energy, healthcare and mobility. Due to the structure of the company, Siemens realized it needed to keep enhancing its competitiveness like all the large international companies, is a very tough job to finish. In this case, Siemens aimed to future develop at the PLM software market. Siemens spent a lot of money to complete the acquisition of UGS, a computer software company that specializes in Product Lifecycle Management software in both 2D and 3D areas. Currently, Siemens have a lot of products in PLM software portfolio, for instance, Teamcenter, Active Integration, NX, Solid Edge, Fibersim, Syncrofit, Seat Design Environment, Femap, LMS, QPE, and Tecomatix. Those PLM products have covered diverse technologies, including PDM, CAD (Computer-aided design), CAM (Computer-aided manufacturing), CAE (Computer-aided engineering), FEA (Finite element analysis), digital manufacturing, and MOM (Manufacturing operations management) etc.

Teamcenter is the most widely used PLM software system in the world [35]. It helps companies deliver complex products to the market by connecting people with products and process in order to enhance productivity and integrate global operations. Basically, this is Siemens PLM software's collaborative product data management solution. Teamcenter provides a more manageable, more productive, collaborative closer and stronger control environment for the manufacturing industry, this capability of Teamcenter simplify and speed up implementation process, increase productivity, enhance corporation, and expand the range of the whole product lifecycle process control. Its overall unified architecture can provide a complete end to end PLM portfolio to user.

Teamcenter customers are distributed at many areas, especially in automotive, aerospace and defense, high-tech electronics and machinery. The major benefit of using Teamcenter can be divided by few aspects. First of all, Productivity is the most important advantage in using Teamcenter, Teamcenter is able to establish a single source of product and process because of this, all the team members can find needed information everywhere and all the time through accessing this common resource. The most direct impact is saving lots of searching time for all individual users. The second is Teamwork; the Teamcenter facilitate collaboration to enables global teams to communicate easily and visually, contact suppliers earlier, and improve the change process to let decision makers make right decisions faster. Teamcenter is a product lifecycle management tool, so manageability is absolutely one of the benefits. Compared to other PLM software systems in the market, Teamcenter is the only system that offers solutions from product planning all the way to retirement. Its end to end solution can help users manage changes acrossing lifecycle. As mentioned earlier, Teamcenter is the most widely used PLM software in the world, which means there are higher probabilities to achieve collaboration between companies by using the same PLM software. In addition, communication between users, development and innovation in Teamcenter itself will be easier to accomplish.

Teamcenter has two tiers: A Rich client tier and a thin client tier. The Thin client is web-based and without the application of computer terminals in the system. The Rich client is installed in the user's machinery. Usually, the Rich client is used by authors and administrators who have access to manage design, create data, and maintain process. The Thin client users are consumers and suppliers who are only needed to view data. In this research, as an administrator, Teamcenter Rich client interface are being used.



Home
Owner: Ware, Connie (de##)
Date Modified: 07-Feb-2013 13:03
Release Status:
Type: Home Folder

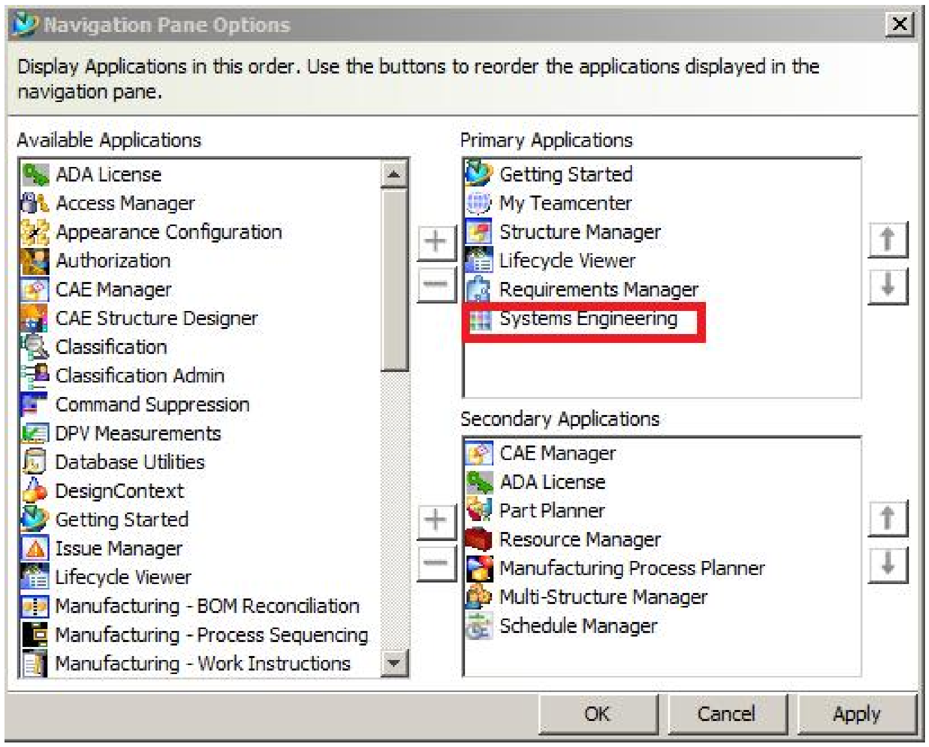
Name: Home
Description: de##
Type: Home Folder
Owner: Ware, Connie (de##)
Group ID: high performance
Last Modifying User: Ware, Connie (de##)

Figure 2.4. Teamcenter rich client interface [36]

Siemens PLM Corporation provided a detailed list of Teamcenter capability. It included design and simulation management, document and content management, BOM management, PLM process execution, requirement management, service life-cycle management, manufacturing management, supplier integration, product cost management, environment compliance and product sustainability, and Systems Engineering. The emphasis of this thesis is using the systems engineering application in Teamcenter to implement and accomplish a PBSE model and establish its platform.

Teamcenter is the first PLM solution to integrate systems engineering within an entire product lifecycle. It provides a close loop systems engineering environment. The systems engineering environment employs systems engineering methodology to allow an engineer to establish systems requirement, then define and validate all component and subsystems in the contact of the entire system's lifecycle. The benefit is that products meet customers' value satisfaction and understand the entire impact of design decisions in the early stages of the lifecycle.

In order to access systems engineering application module, in the Teamcenter interface, navigation pane options allow the users to select systems engineering applications in primary or secondary application window.


Navigation Pane Options

Display Applications in this order. Use the buttons to reorder the applications displayed in the navigation pane.

Available Applications
- ADA License
- Access Manager
- Appearance Configuration
- Authorization
- CAE Manager
- CAE Structure Designer
- Classification
- Classification Admin
- Command Suppression
- DPV Measurements
- Database Utilities
- DesignContext
- Getting Started
- Issue Manager
- Lifecycle Viewer
- Manufacturing - BOM Reconciliation
- Manufacturing - Process Sequencing
- Manufacturing - Work Instructions

Primary Applications
- Getting Started
- My Teamcenter
- Structure Manager
- Lifecycle Viewer
- Requirements Manager
- Systems Engineering

Secondary Applications
- CAE Manager
- ADA License
- Part Planner
- Resource Manager
- Manufacturing Process Planner
- Multi-Structure Manager
- Schedule Manager

Figure 2.5. Teamcenter systems engineering in navigation pane [36]

After selecting the systems engineering application in the Primary Application, the systems engineering icon is shown in Figure 2.6 which is the primary application tab .

In addition, the most important reason for us to choose Teamcenter is because Teamcenter systems engineering is the most relevant application to Systematica methodology, models, and patterns. This application allows a system engineer to view and manage physical, logical, functional, and requirement statement hierarchies and trace relationships between them.

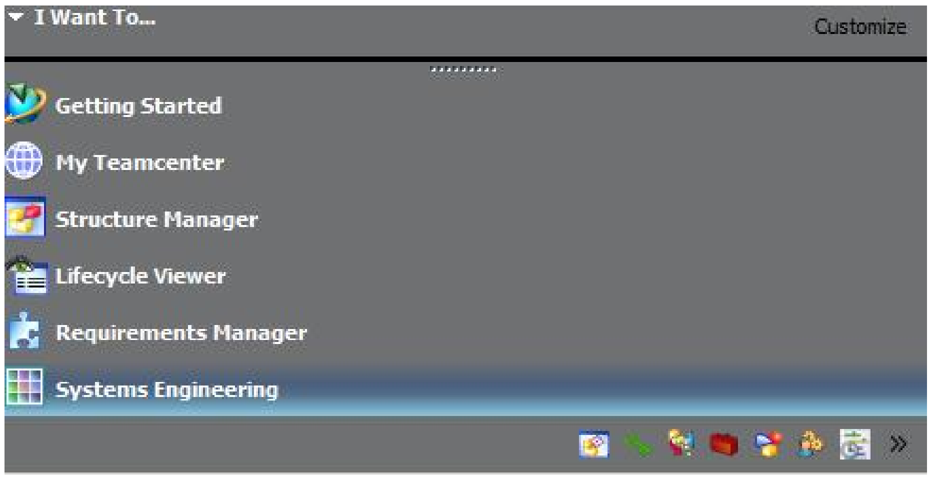
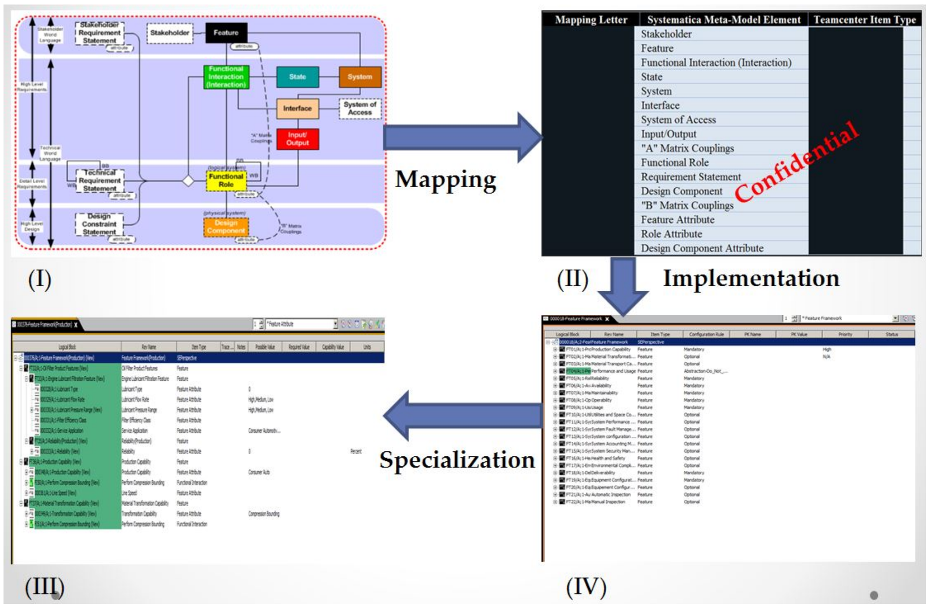


Figure 2.6. Teamcenter primary applications of systems engineering [36]

3. IMPLEMENTATION AND DEMONSTRATION

3.1 Research Approach

The methodology followed in this research consists of three steps: mapping; implementation; and specialization. This is illustrated in Figure 3.1. The first step was the mapping process from Systematica Metamodel to Teamcenter Schema in BMIDE. The second step was the implementation process in Teamcenter interface. The last step was the specialization and configuration by using the S*Pattern in oil filter end seal compression manufacturing process.

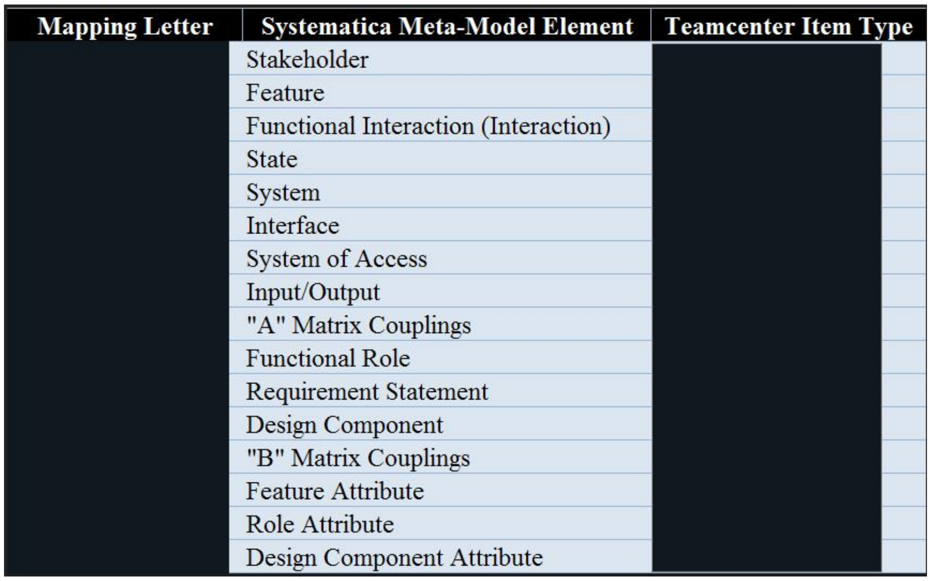


Mapping Letter	Systematica Meta-Model Element	Teamcenter Item Type
[redacted]	Stakeholder	[redacted]
[redacted]	Feature	[redacted]
[redacted]	Functional Interaction (Interaction)	[redacted]
[redacted]	State	[redacted]
[redacted]	System	[redacted]
[redacted]	Interface	[redacted]
[redacted]	System of Access	[redacted]
[redacted]	Input/Output	[redacted]
[redacted]	"A" Matrix Couplings	[redacted]
[redacted]	Functional Role	[redacted]
[redacted]	Requirement Statement	[redacted]
[redacted]	Design Component	[redacted]
[redacted]	"B" Matrix Couplings	[redacted]
[redacted]	Feature Attribute	[redacted]
[redacted]	Role Attribute	[redacted]
[redacted]	Design Component Attribute	[redacted]

Figure 3.1. Research process approach (I) The S* Pattern compatible with S* Metamodel (II) Mapping documental of S* Metalmodel to Teamcenter schema (columns are hidden due to confidentiality) (III) Sample implementation of S* Metamodel in Teamcenter interface (IV) Configured S* Pattern of oil filter end seal compression research.

3.2 Mapping in BMIDE

Teamcenter was chosen in this research because of its capabilities. Teamcenter systems engineering is the most relevant application to Systematica models and patterns. This application is allowed a system engineer's view to manage physical, logical, functional, and requirement statement hierarchies and trace relationship between them. Before doing schema configuration in Teamcenter systems engineering, it is necessary to use BMIDE interface to define data model objects (business object, classes, properties etc.). Those data model objects are based on S* Metamodel elements mapping to Teamcenter. The first version of the mapping document [37] was summarized by the following graph; due to the confidential agreement with the industrial sponsor of this research, some information has been removed/protected.



Mapping Letter	Systematica Meta-Model Element	Teamcenter Item Type
[redacted]	Stakeholder	[redacted]
[redacted]	Feature	[redacted]
[redacted]	Functional Interaction (Interaction)	[redacted]
[redacted]	State	[redacted]
[redacted]	System	[redacted]
[redacted]	Interface	[redacted]
[redacted]	System of Access	[redacted]
[redacted]	Input/Output	[redacted]
[redacted]	"A" Matrix Couplings	[redacted]
[redacted]	Functional Role	[redacted]
[redacted]	Requirement Statement	[redacted]
[redacted]	Design Component	[redacted]
[redacted]	"B" Matrix Couplings	[redacted]
[redacted]	Feature Attribute	[redacted]
[redacted]	Role Attribute	[redacted]
[redacted]	Design Component Attribute	[redacted]

Figure 3.2. Mapping documental of S* Metalmodel to Teamcenter schema (columns are hidden due to confidentiality)

In this mapping document, the main design choices include: mapping most systematica classes to specialized classes in Teamcenter Logical Block item type, mapping most systematica relationships to Teamcenter structural relationships, mapping Systematica classes of different attributes (Feature Attribute, Role Attribute, and

Physical System Attribute) to specialized item and attaching requirement statement content into Requirement Attribute Table.

Business objects are fundamental objects to represent product parts, documents, change process, and so on. Item, Item Revision, Dataset, Folder and Form are used in business objects frequently. A basic structure of an "Item" object consists of item master, item revision, views and other object forms, an "Item Revisions" uses to manage a specific revision of an item. The "Item" object was used in this research to represent systems engineering logical blocks, processes, requirements and similar concepts. As shown in the figure below, the "Fnd0SEBlock" object was attached under "Item" to represent systems engineering logical blocks. Logical blocks, which were created to define abstract physical architectures for implementing system functions, represent solution components. In this research, the logical block was the most appropriate object to represent S* Metamodel items.

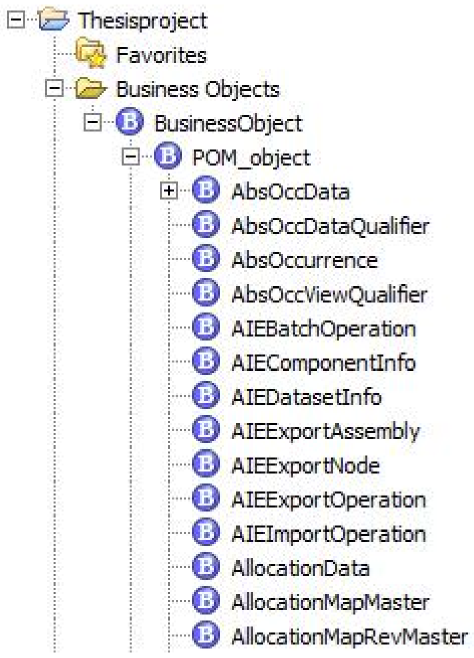


Figure 3.3. Business objects in BMIDE view

Once the mapping document that contains specialized item types, specialized relationship types, and the mapping methods were prepared for Systematica Schema

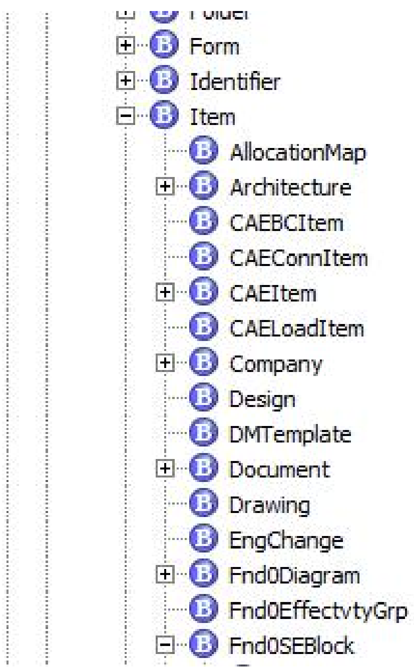
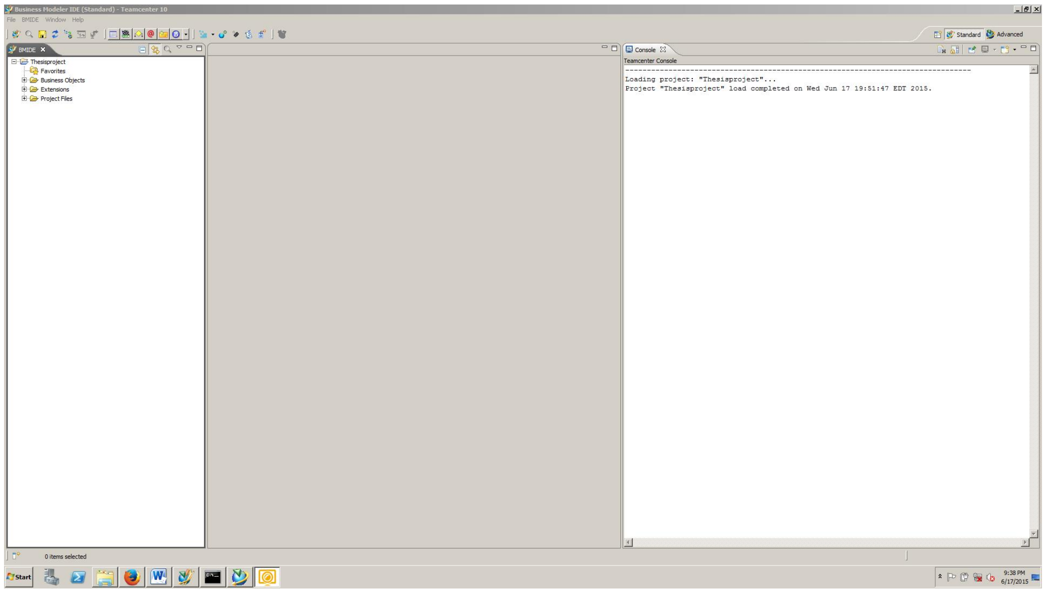


Figure 3.4. The Item business object in BMIDE view

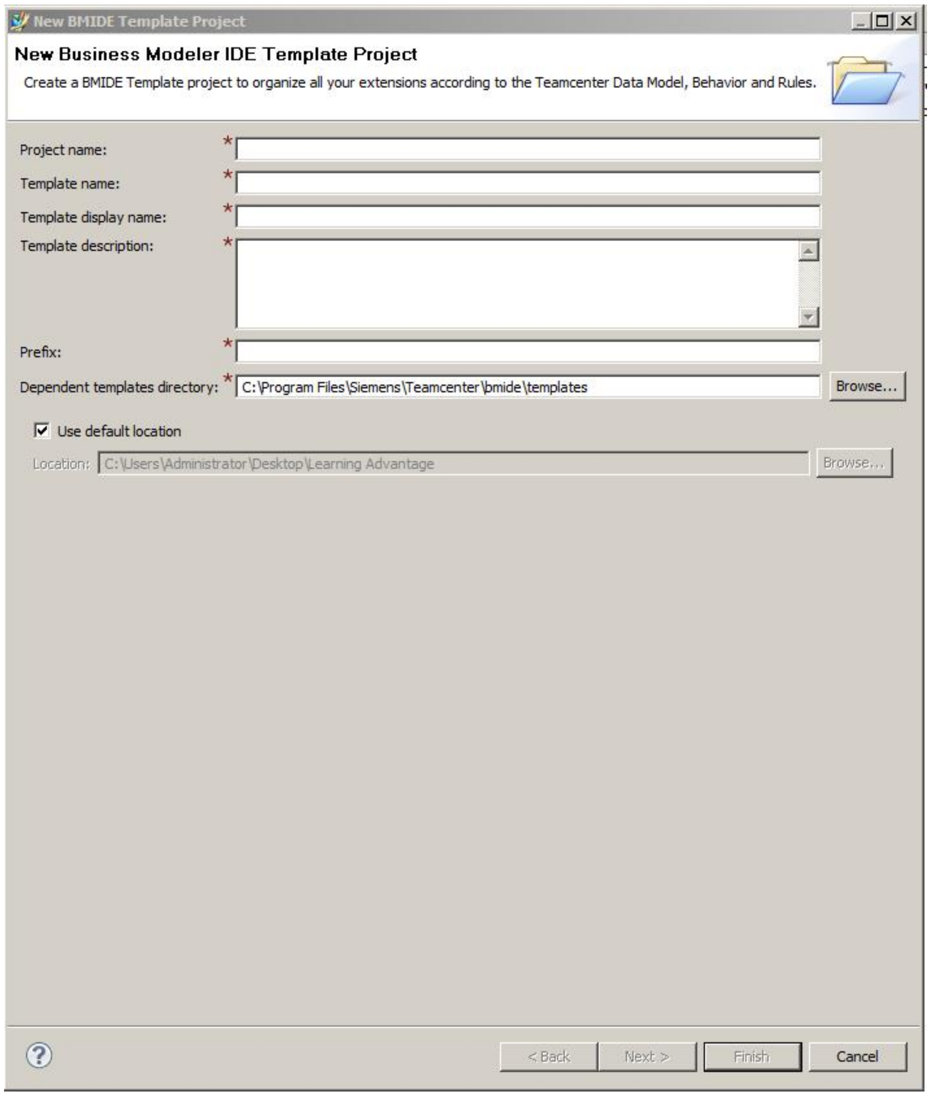
configuration, creating a new BMIDE project template was the first step to configure Teamcenter schema.



```
--------------------------------------------------------------------------------
Loading project: "Thesisproject"...
Project "Thesisproject" load completed on Wed Jun 17 19:51:47 EDT 2015.
```

Figure 3.5. Business Modeler IDE interface

In the BMIDE interface, Selecting File, New, and New BMIDE Template Project, it allows users to create a BMIDE Template Project to organize all extensions according to the Teamcenter Data Model, Behavior and Rules. The BMIDE Template Project provides a template environment for users to organize XML files in folders instead of coding XML files, and packaging template for deployment.


New BMIDE Template Project

New Business Modeler IDE Template Project

Create a BMIDE Template project to organize all your extensions according to the Teamcenter Data Model, Behavior and Rules.

Project name: *

Template name: *

Template display name: *

Template description: *

Prefix: *

Dependent templates directory: * C:\Program Files\Siemens\Teamcenter\bmide\templates

[✓] Use default location

Location: C:\Users\Administrator\Desktop\Learning Advantage

Figure 3.6. New Business Modeler IDE template project creation window

The project name is chosen by users. Template name and template display name are default to the project name. Template description will appear in Teamcenter environment manager for reviewing and understanding. Prefix is a unique naming to distinguish with other projects.

In the next step, BMIDE also allows users to select a dependent template, choose a language from a list in Locals Selector, write a code in Code Generation Information if needed, make setting in the Build Configuration Information, and make setting in the Service Bindings Configuration Information. A new BMIDE template will be generated once all the information is finished.

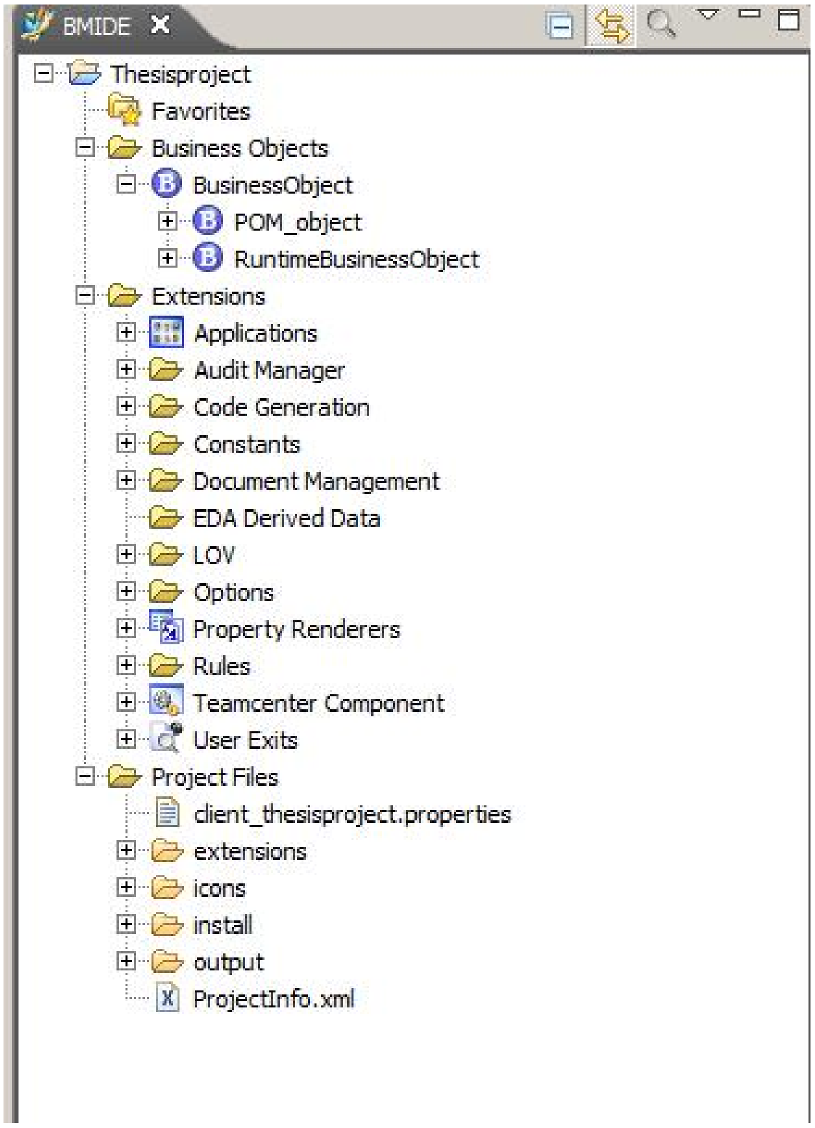


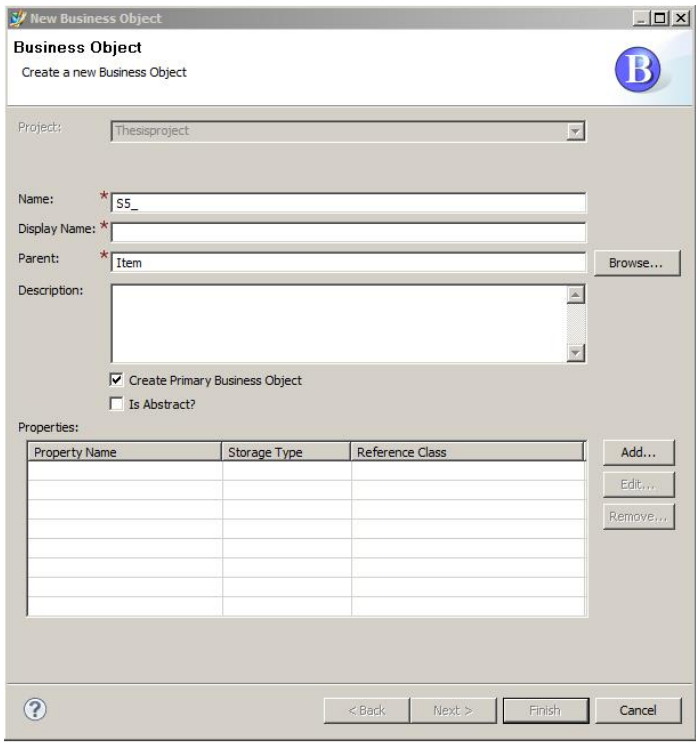
Figure 3.7. View of “Thesisproject” template project

In this research, this template was called "Thesisproject" with a certain prefix had been created. As mentioned earlier in the mapping document, most of the specialized item types will be constructed under a specific item type. That specific item type will be used as a "Father" COTS object for specialized item types.

In order to better control specialized items without affecting Teamcenter COTs items, two higher level object groups were created based on different characteristics of objects: one was created to manage and organize systems engineering specialized items and another one was created to represent requirement statements and needs. By creating higher level object groups, it is easier to add common property or value for all specialized items.

In Teamcenter, relationships are also defined in business objects. "ImanRelation" is the object to manage and organize relationships between business objects. One specialized relationship type called "trace link" was created to represent relationships in hierarchies. For example, features in systems engineering model were created based on stakeholders needs, A "Need" object is the source of a "Feature" object.

In BMIDE interface, in order to create a new business object, it is necessary to understand the upper level object, which is also called "Father" object, for that new created object. Understanding upper level object is not only for creating hierarchies, but also hesitating properties and values. For example, in this research, most of the specialized objects were created under a certain item type; the two higher level object groups were also created under that specific item type; all the properties and values existed in COTs business object automatically comply with that two higher level object groups. In addition, it is easy to show the relationship between different levels. After figuring out the relationship between different levels, it is ready to create a new business object under the "Father" object. The figure below showed the window of creating a new business object.


New Business Object

Business Object
Create a new Business Object

Project: Thesisproject

Name: * S5_
Display Name: *
Parent: * Item
Description:

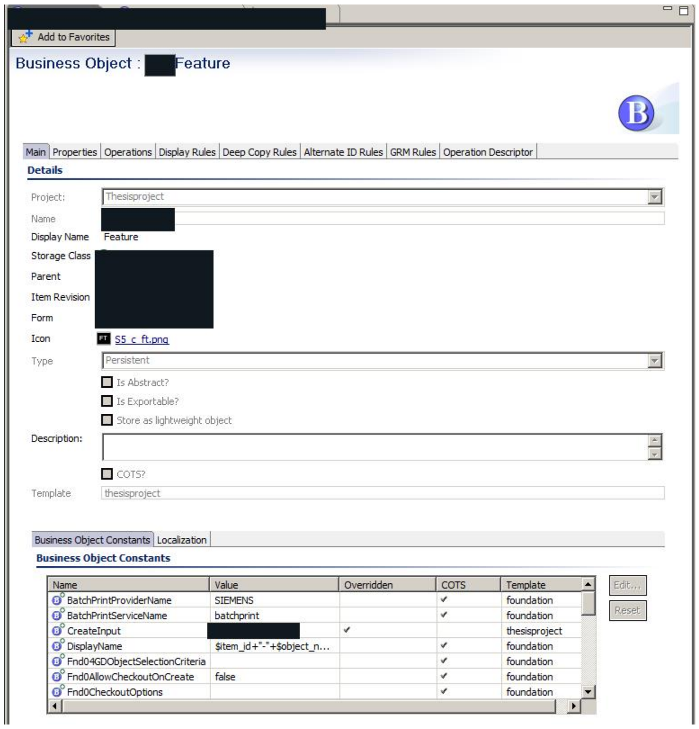
[✓] Create Primary Business Object
[ ] Is Abstract?

Properties:

Property Name	Storage Type	Reference Class

Figure 3.8. New business object creating window

After creating a new business object, a new business class that contains a small letter “c” showed in BMIDE view. The two figures below showed editor views of sample specialized items which are “Feature” and “FeatureRevision”. In the object editor view window, all the characteristics of the new business object are demonstrated. From the two figures, feature editor window and feature revision editor window look similar. Project name, display name, parent class, item revision, and object icon appeared in both main tabs. All the properties details were shown in property interface which allow users to create new properties.



[redacted]

Add to Favorites

Business Object : [redacted]Feature

Main | Properties | Operations | Display Rules | Deep Copy Rules | Alternate ID Rules | GRM Rules | Operation Descriptor

Details

Project:	Thesisproject
Name	[redacted]
Display Name	Feature
Storage Class	[redacted]
Parent	[redacted]
Item Revision	[redacted]
Form	[redacted]
Icon	S5_c_ft.png
Type	Persistent
	☐ Is Abstract?
	☐ Is Exportable?
	☐ Store as lightweight object
Description:	
	☐ COTS?
Template	thesisproject

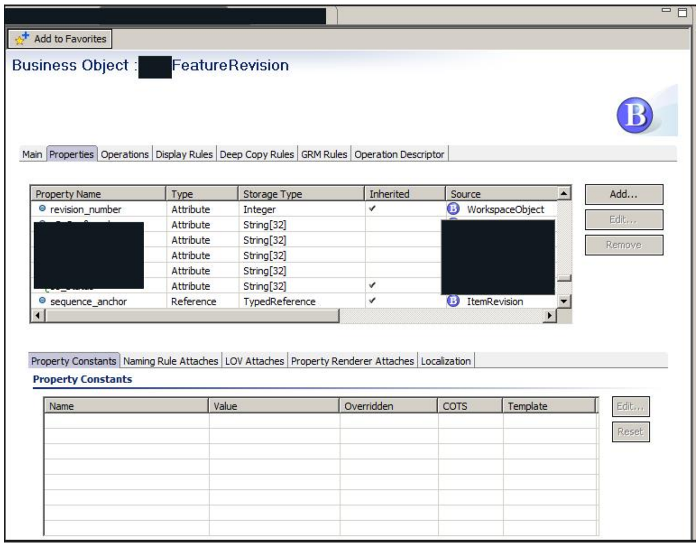
Business Object Constants | Localization

Business Object Constants

Name	Value	Overridden	COTS	Template
BatchPrintProviderName	SIEMENS		✓	foundation
BatchPrintServiceName	batchprint		✓	foundation
CreateInput	[redacted]	✓		thesisproject
DisplayName	$item_id+"-"+$object_n...		✓	foundation
Fnd04GDObjectSelectionCriteria			✓	foundation
Fnd0AllowCheckoutOnCreate	false		✓	foundation
Fnd0CheckoutOptions			✓	foundation

Edit... Reset

Figure 3.9. Sample business object Feature view (contents are hidden due to confidentiality)



[redacted]

Add to Favorites

Business Object : [redacted]FeatureRevision

Main | Properties | Operations | Display Rules | Deep Copy Rules | GRM Rules | Operation Descriptor

Property Name	Type	Storage Type	Inherited	Source
revision_number	Attribute	Integer	✓	WorkspaceObject
[redacted]	Attribute	String[32]		[redacted]
[redacted]	Attribute	String[32]		[redacted]
[redacted]	Attribute	String[32]		[redacted]
[redacted]	Attribute	String[32]		[redacted]
[redacted]	Attribute	String[32]	✓	[redacted]
sequence_anchor	Reference	TypedReference	✓	ItemRevision

Add... Edit... Remove

Property Constants | Naming Rule Attaches | LOV Attaches | Property Renderer Attaches | Localization

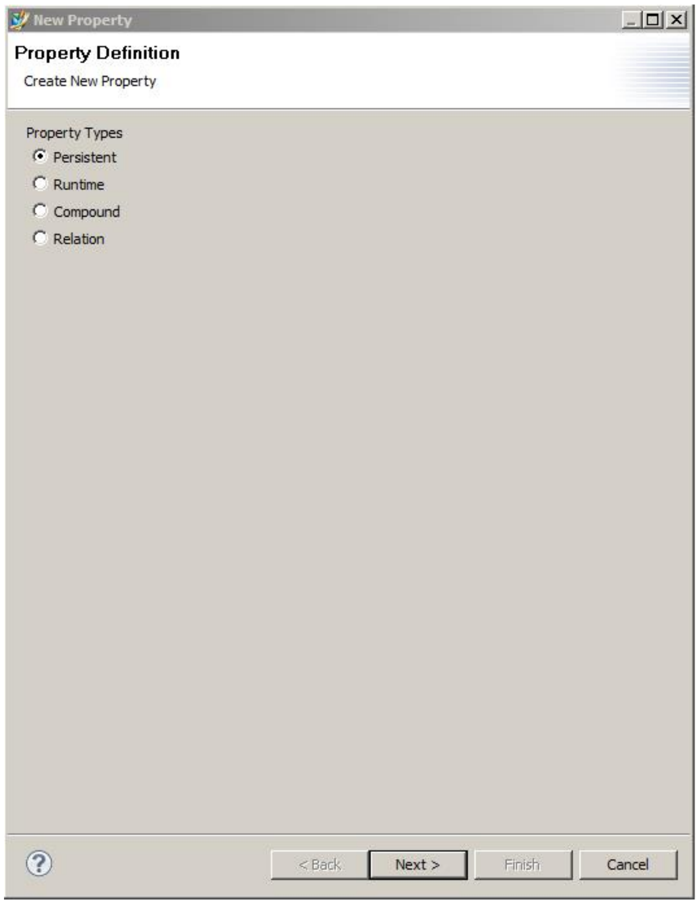
Property Constants

Name	Value	Overridden	COTS	Template

Edit... Reset

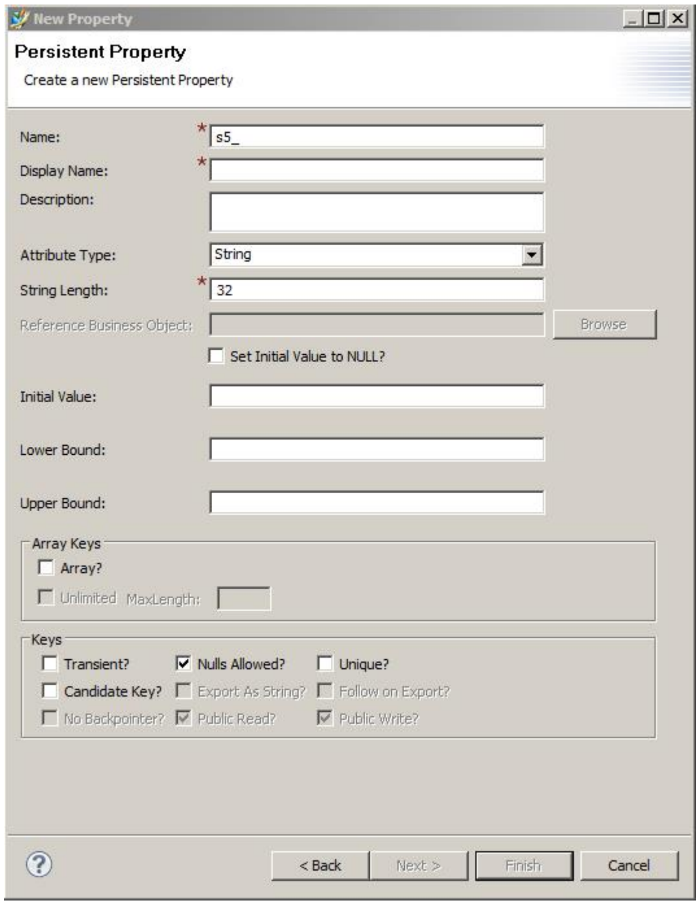
Figure 3.10. Sample business object FeatureRevision view (contents are hidden due to confidentiality)

The “ItemRevision” manages the specific revision instance of an item. In order to have properties show in Teamcenter BOM view, all the properties should be added in “ItemRevision” property window. In the “ItemRevision” window, the add button was on the right side of the property table which enables users to create new properties. A property window showed up and contained a few property types by clicking the add button. In this research, persistent properties were selected in property types to use in Systematica Metamodel elements. The figures showed the procedure to create a new persistent property in BMIDE.



Property Types
[✓ Persistent]
Runtime
Compound
Relation

Figure 3.11. Creating a new property in a business object



New Property

Persistent Property

Create a new Persistent Property

Name: * s5_

Display Name: *

Description:

Attribute Type: String

String Length: * 32

Reference Business Object: Browse

☐ Set Initial Value to NULL?

Initial Value:

Lower Bound:

Upper Bound:

Array Keys

☐ Array?

☐ Unlimited MaxLength:

Keys

☐ Transient? ☑ Nulls Allowed? ☐ Unique?

☐ Candidate Key? ☐ Export As String? ☐ Follow on Export?

☐ No Backpointer? ☑ Public Read? ☑ Public Write?

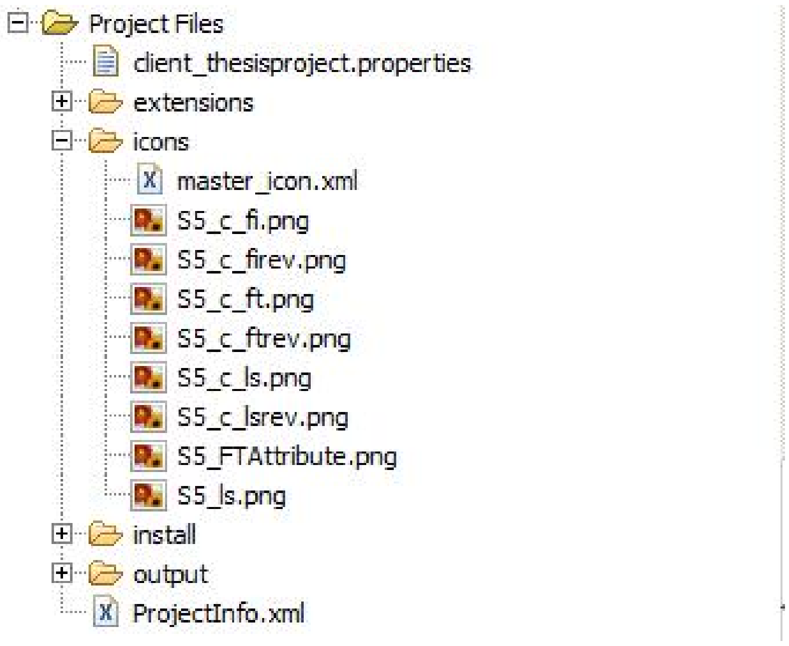
< Back | Next > | Finish | Cancel

Figure 3.12. Creating a new persistent property in a business object

The specialized items, modeled class display name, properties created in each object, properties' display names, and attribute types of each property were summarized in the research.

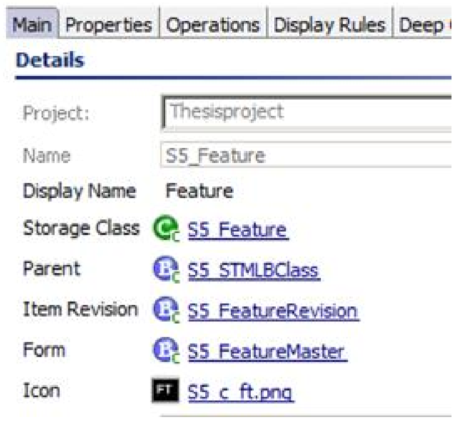
Since many specialized items were created in BMIDE, providing a visual distinction of each item is necessary. In this case, adding icons for different specialized items enable diversity appearance in Teamcenter. In the view of project template, the icons folder is underneath the "Project Files". Administrators or users are able to insert and store defined customer icons' photos in this folder. Going back to specific business object editor views enable to change icons by selecting defined customer icons figure from icons folder instead of using default icons. The Figures 3.14 and 3.15

were examples of a changed feature icon and a changed feature revision icon. In this research, a black color background and an acronym was used to represent feature and feature revision; a green color background and an acronym was created to represent Interaction and Interaction revision; a yellow color background and an acronym represented logical system and logical system revision.



```
Project Files
    client_thesisproject.properties
    extensions
    icons
        master_icon.xml
        S5_c_fi.png
        S5_c_firev.png
        S5_c_ft.png
        S5_c_ftrev.png
        S5_c_ls.png
        S5_c_lsrev.png
        S5_FTAttribute.png
        S5_ls.png
    install
    output
    ProjectInfo.xml
```

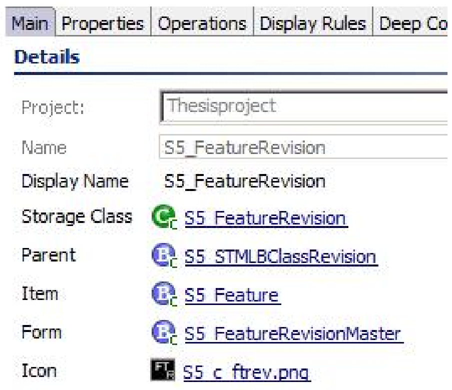
Figure 3.13. Icons creating in project files



Details

Project:	Thesisproject
Name	S5_Feature
Display Name	Feature
Storage Class	S5_Feature
Parent	S5_STMLBClass
Item Revision	S5_FeatureRevision
Form	S5_FeatureMaster
Icon	S5_c_ft.png

Figure 3.14. Changed icon in business object “Feature” view



Main | Properties | Operations | Display Rules | Deep Co

Details

Project:	Thesisproject
Name	S5_FeatureRevision
Display Name	S5_FeatureRevision
Storage Class	S5_FeatureRevision
Parent	S5_STMLBClassRevision
Item	S5_Feature
Form	S5_FeatureRevisionMaster
Icon	S5_c_ftrev.png

Figure 3.15. Changed icon in business object “FeatureRevision” view

An icon is a good way to distinguish different objects. In addition, naming rule is another capability to organize those diversity business objects in Teamcenter view. Naming rules consist of a rule patter and a counter to define the data entry format for a business object property. Underneath the “Extensions” folder, “Naming Rule” folder enables users to create a new naming rule.

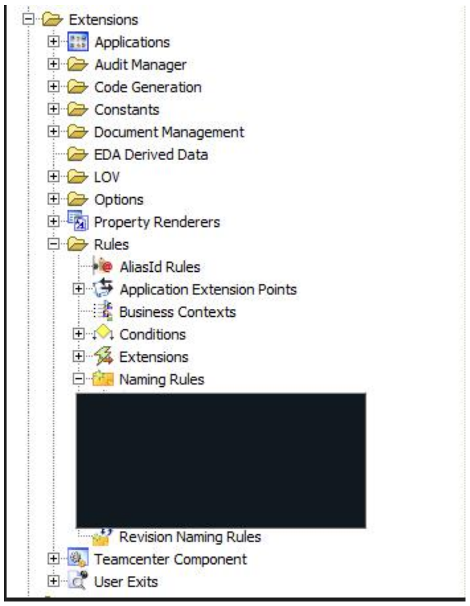
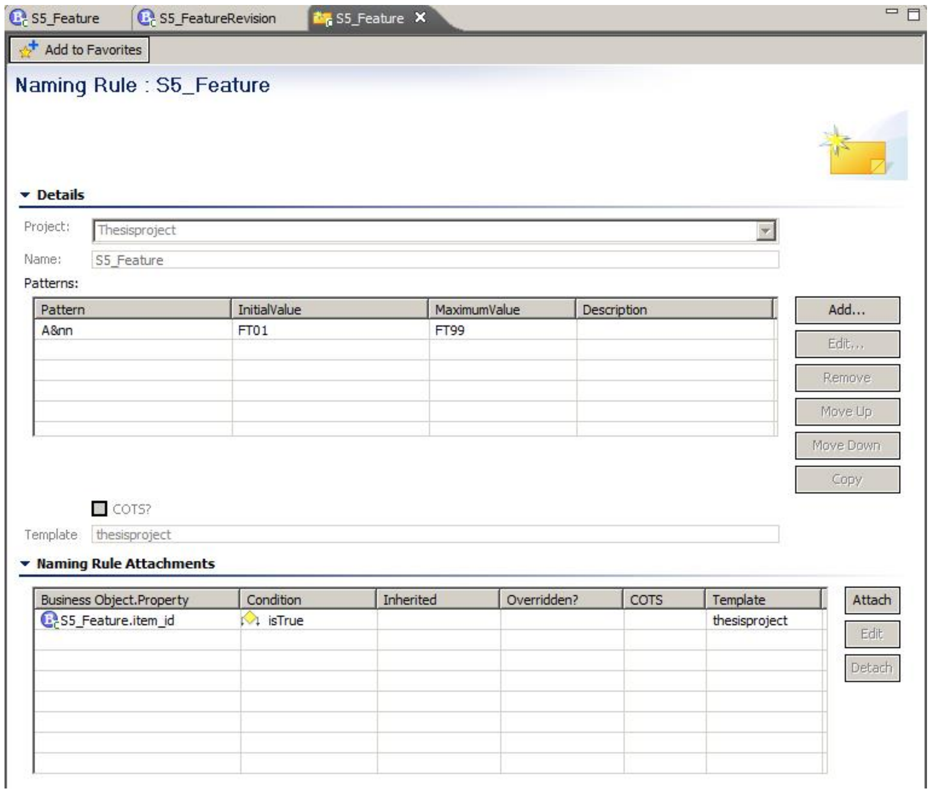


Figure 3.16. Naming rule created in rules folder (contents are hidden due to confidentiality)

In the naming rule window, pattern, initial value, maximum value, and description are able to be added in a “add naming rule pattern” window. Once a naming rule is created, it is necessary to attach it in the correspondent business object property. An example below shows a feature naming rule created in BMIDE.



Naming Rule : S5_Feature

Details

Project: Thesisproject

Name: S5_Feature

Patterns:

Pattern	InitialValue	MaximumValue	Description
A&nn	FT01	FT99	

COTS?

Template thesisproject

Naming Rule Attachments

Business Object.Property	Condition	Inherited	Overridden?	COTS	Template
S5_Feature.item_id	isTrue				thesisproject

Figure 3.17. The view of sample naming rule in business object feature

The summary of Icon name within specialized objects and details about naming rules, pattern, initial value, and maximum value were create in research. In order to provide a convenient environment for end users, Lists of values (LOVs) is a concrete capability for end users to pick a list of defined values which are displayed in the Teamcenter data entry box. BMIDE enables users to create three different types of LOVs which are Batch LOV, Classis LOV, and Dynamic LOV. In this research, a classic LOV was created due to its string type. The figures are details creating procedures of a classic LOVs in BMIDE.

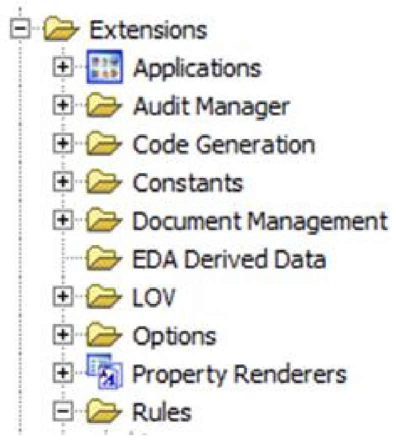
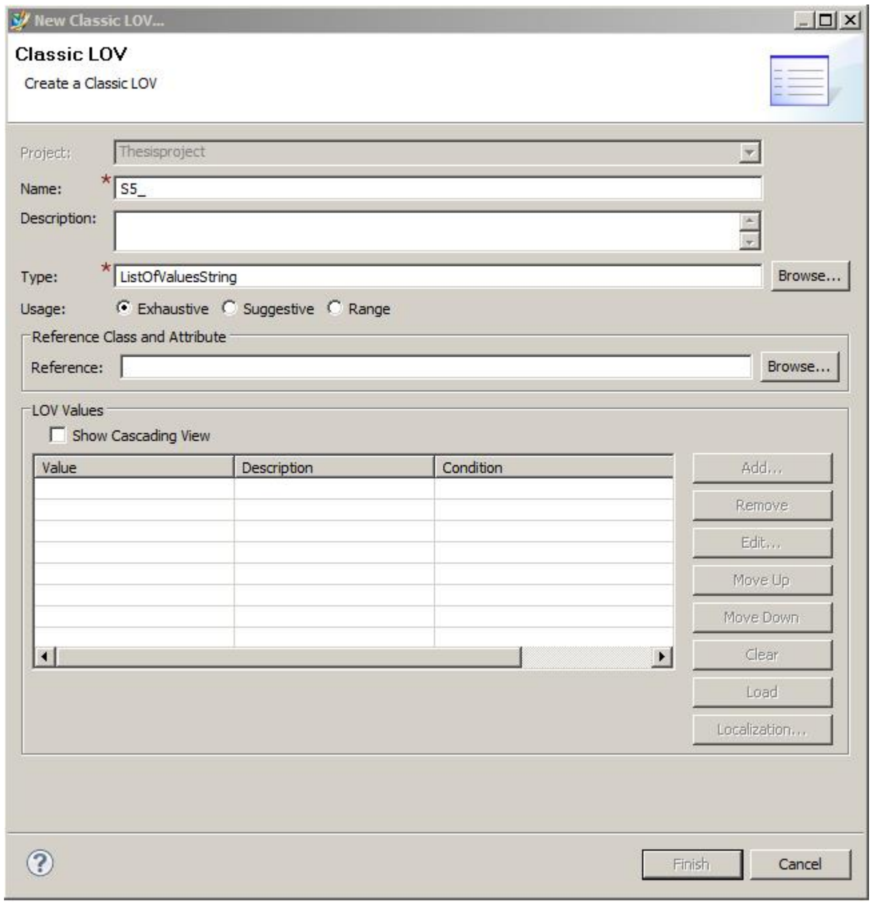


Figure 3.18. List of values folder in BMIDE view


New Classic LOV...

Classic LOV

Create a Classic LOV

Project: Thesisproject

Name: * S5_

Description:

Type: * ListOfValuesString

Usage: [✓ Exhaustive] Suggestive Range

Reference Class and Attribute

Reference:

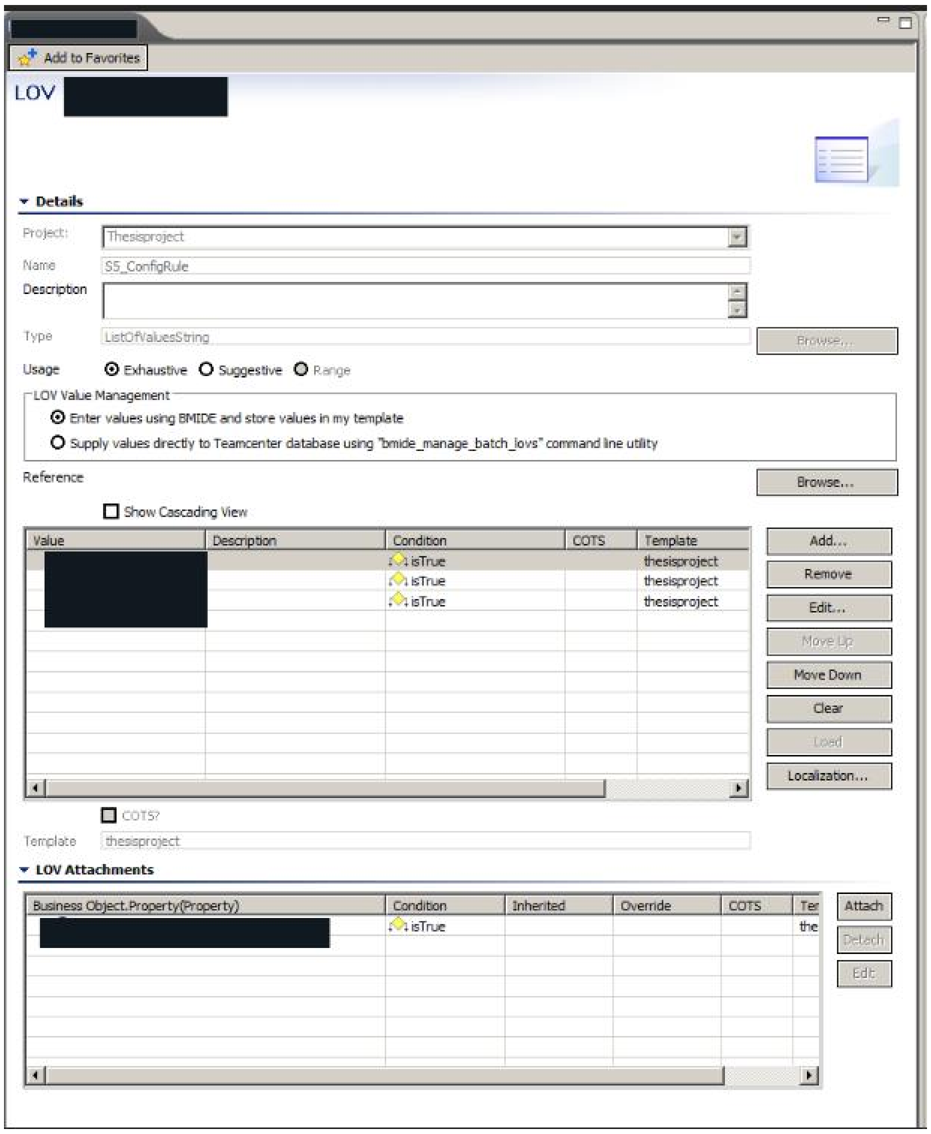
LOV Values

Show Cascading View

Value	Description	Condition

Figure 3.19. Creating a new classic list of Values window

The figure below is an example of a created custom LOV. The table in the middle of the dialog box was the place to create customer defined values which will appear in a data entry box for picking. Once finished building a picking list, it is necessary to attach these LOVs into a business object property.


[redacted]

LOV [redacted]

Details

Project: Thesisproject

Name: S5_ConfigRule

Description:

Type: ListOfValuesString

Usage: [✓ Exhaustive] Suggestive Range

LOV Value Management

[✓ Enter values using BMIDE and store values in my template]

Supply values directly to Teamcenter database using "bmide_manage_batch_lovs" command line utility

Reference

Show Cascading View

Value	Description	Condition	COTS	Template
[redacted]		isTrue		thesisproject
		isTrue		thesisproject
		isTrue		thesisproject

COTS?

Template: thesisproject

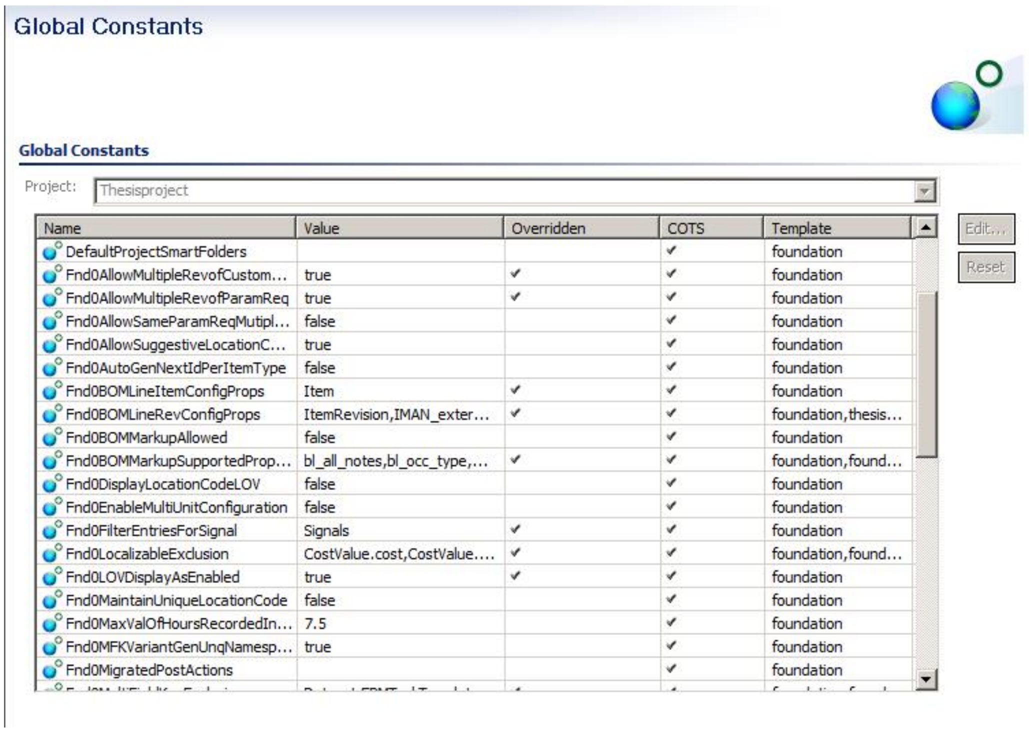
LOV Attachments

Business Object.Property(Property)	Condition	Inherited	Override	COTS	Ter
[redacted]	isTrue				the

Figure 3.20. LOVs view of a sample property (contents are hidden due to confidentiality)

After created the project template, constructing business objects, adding properties, and defining icons, list of values, and naming rules. The last step to finish BMIDE was to implement custom properties which appear in Teamcenter BOM columns (the explanation of BOM view columns will be illustrated later). There are a few requirements that have to be done to add custom properties in BOM columns in order to display in Teamcenter Structure Manager or system engineering application, such as: customer business objects must be created under the "ItemRevision" objects and customs properties must be added to those customer business objects.

The first step to add custom properties in BOM columns was enabling Global Constants Editor. On the menu bar, choose Global Constants Editor in BMIDE Editors, Global Constants provides consistent definitions which have either default values or custom values used throughout the system. The constant was selected in "Fnd0BOMLineRevCongifProps". From the name of this constant, it is easy to tell that this constant is used for adding properties from item revision types



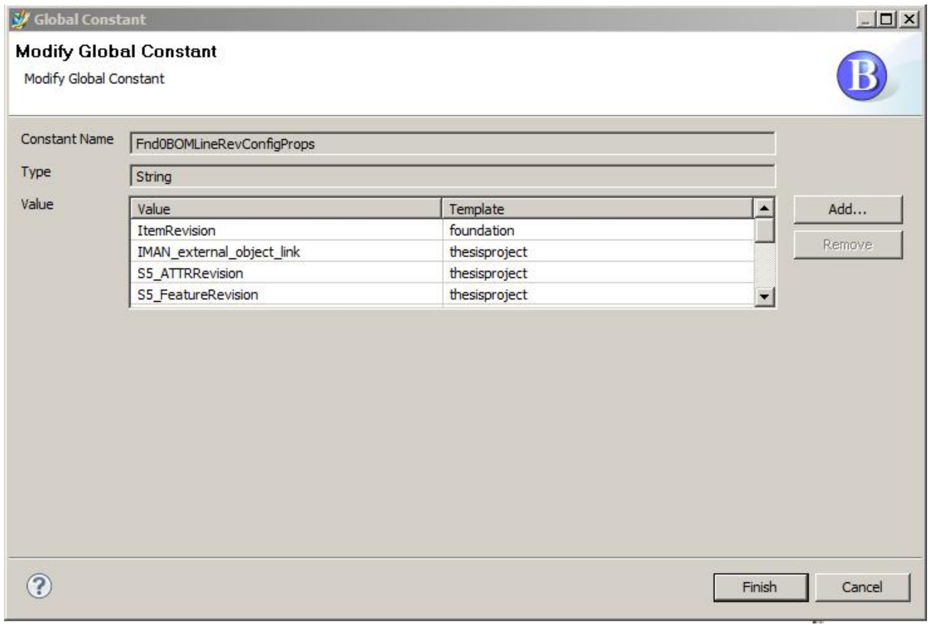
Global Constants

Project: Thesisproject

Name	Value	Overridden	COTS	Template
DefaultProjectSmartFolders			✓	foundation
Fnd0AllowMultipleRevofCustom...	true	✓	✓	foundation
Fnd0AllowMultipleRevofParamReq	true	✓	✓	foundation
Fnd0AllowSameParamReqMutipl...	false		✓	foundation
Fnd0AllowSuggestiveLocationC...	true		✓	foundation
Fnd0AutoGenNextIdPerItemType	false		✓	foundation
Fnd0BOMLineItemConfigProps	Item	✓	✓	foundation
Fnd0BOMLineRevConfigProps	ItemRevision,IMAN_exter...	✓	✓	foundation,thesis...
Fnd0BOMMarkupAllowed	false		✓	foundation
Fnd0BOMMarkupSupportedProp...	bl_all_notes,bl_occ_type,...	✓	✓	foundation,found...
Fnd0DisplayLocationCodeLOV	false		✓	foundation
Fnd0EnableMultiUnitConfiguration	false		✓	foundation
Fnd0FilterEntriesForSignal	Signals	✓	✓	foundation
Fnd0LocalizableExclusion	CostValue.cost,CostValue....	✓	✓	foundation,found...
Fnd0LOVDisplayAsEnabled	true	✓	✓	foundation
Fnd0MaintainUniqueLocationCode	false		✓	foundation
Fnd0MaxValOfHoursRecordedIn...	7.5		✓	foundation
Fnd0MFKVariantGenUnqNamesp...	true		✓	foundation
Fnd0MigratedPostActions			✓	foundation

Figure 3.21. The global constant window

When this constant had been selected and edited, a dialog box “Modify Global Constant” showed on the screen to enable adding those customer objects revision, which contains custom properties, by click “add button”.



Modify Global Constant

Modify Global Constant

Constant Name: Fnd0BOMLineRevConfigProps

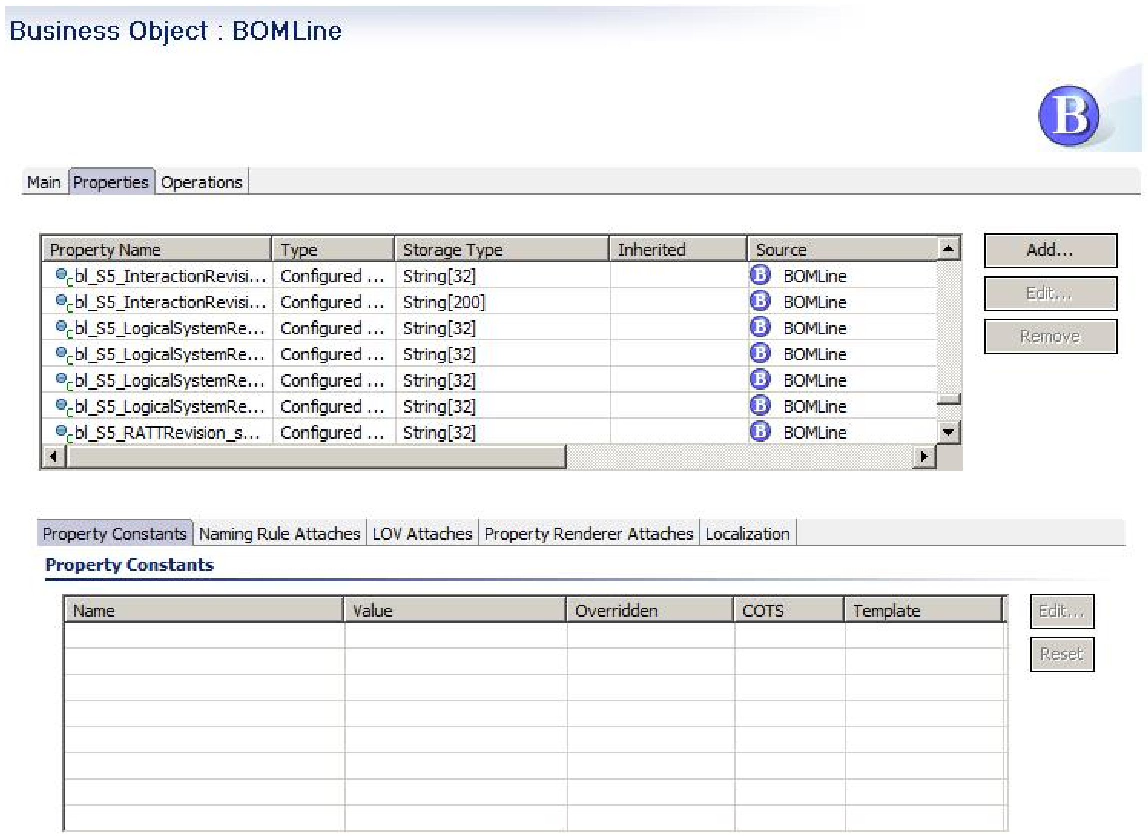
Type: String

Value:

Value	Template
ItemRevision	foundation
IMAN_external_object_link	thesisproject
S5_ATTRRevision	thesisproject
S5_FeatureRevision	thesisproject

Figure 3.22. Added business object revisions in modify global constant window

In the BMIDE training document, it mentioned “When you create a new constant, you must also add the code on the server to return the constant’s value to the caller, so the caller can branch the business logic based on the returned value [38]”. In this case, reload data had to be done before using BOMLine. In the BOMLine dialog box, new property names appeared in BOM columns starting with bi allow users to edit their display name for using in BOM view in Teamcenter.


Business Object : BOMLine

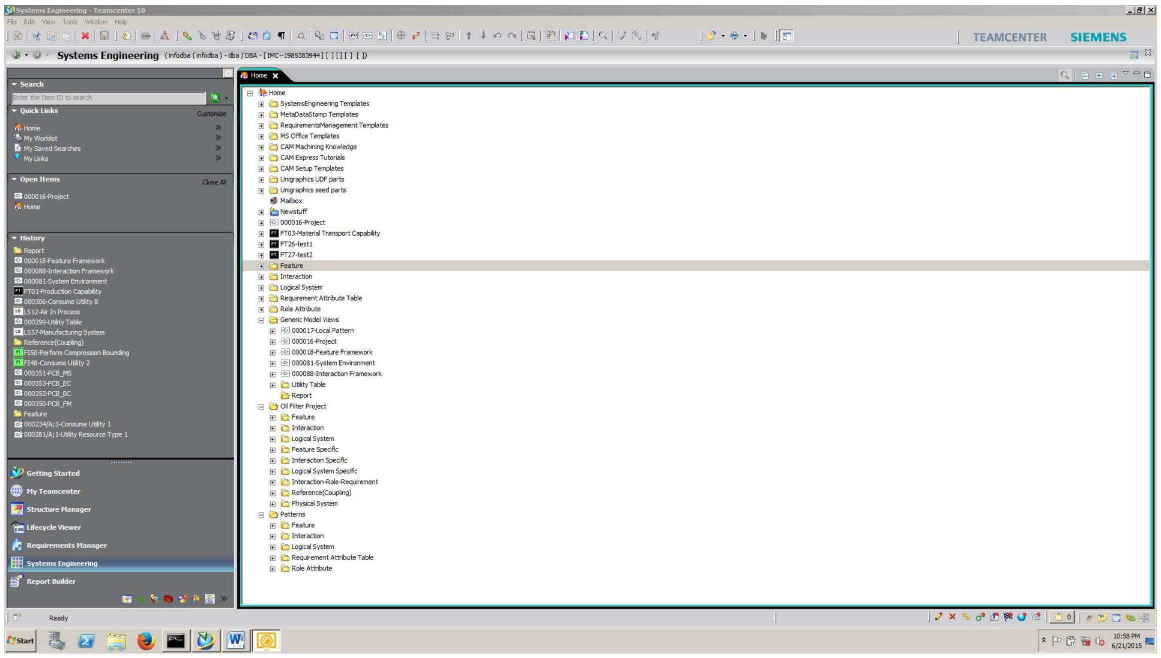
Property Name	Type	Storage Type	Inherited	Source
bl_S5_InteractionRevisi...	Configured ...	String[32]		BOMLine
bl_S5_InteractionRevisi...	Configured ...	String[200]		BOMLine
bl_S5_LogicalSystemRe...	Configured ...	String[32]		BOMLine
bl_S5_LogicalSystemRe...	Configured ...	String[32]		BOMLine
bl_S5_LogicalSystemRe...	Configured ...	String[32]		BOMLine
bl_S5_LogicalSystemRe...	Configured ...	String[32]		BOMLine
bl_S5_RATTRevision_s...	Configured ...	String[32]		BOMLine

Name	Value	Overridden	COTS	Template

Figure 3.23. Modified properties in “BOMLine” view

3.3 Implementation of General Production Pattern

The objective of this research was to implement the trace of Feature-Interaction-Functional Roles-Design Component from General Production Pattern [39] based on S* Metamodel into Teamcenter Systems Engineering. In the Teamcenter Systems Engineering interface (Figure 3.24), several folders were created to store different specialized items for better organization. The steps to create a folder in Teamcenter rich client interface included New, Meum, and Folder.



Systems Engineering (infodba (infodba) - dba / DBA - [IMC--1985383944] [ ] [] [ ] [ ])

```
Open Items
000016-Project
Home

History
Report
000018-Feature Framework
000088-Interaction Framework
000081-System Environment
FT01-Production Capability
000306-Consume Utility 8
LS12-Air In Process
000299-Utility Table
LS37-Manufacturing System
Reference(Coupling)
FI50-Perform Compression Bounding
FI48-Consume Utility 2
000351-PCB_MS
000353-PCB_EC
000352-PCB_BC
000350-PCB_FM
Feature
000234/A;3-Consume Utility 1
000281/A;1-Utility Resource Type 1
```

```
Home
  SystemsEngineering Templates
  MetaDataStamp Templates
  RequirementsManagement Templates
  MS Office Templates
  CAM Machining Knowledge
  CAM Express Tutorials
  CAM Setup Templates
  Unigraphics UDF parts
  Unigraphics seed parts
  Mailbox
  Newstuff
  000016-Project
  FT03-Material Transport Capability
  FT26-test1
  FT27-test2
  Feature
  Interaction
  Logical System
  Requirement Attribute Table
  Role Attribute
  Generic Model Views
    000017-Local Pattern
    000016-Project
    000018-Feature Framework
    000081-System Environment
    000088-Interaction Framework
    Utility Table
    Report
  Oil Filter Project
    Feature
    Interaction
    Logical System
    Feature Specific
    Interaction Specific
    Logical System Specific
    Interaction-Role-Requirement
    Reference(Coupling)
    Physical System
  Patterns
    Feature
    Interaction
    Logical System
    Requirement Attribute Table
    Role Attribute
```

Figure 3.24. Teamcenter systems engineering interface

Currently, Feature, Interaction, Logical Systems, Requirement Attribute Table (include Attribute Table Row and Requirement Relationship), and Role Attribute folders that contain correspondent items under a Patterns folder were created. In addition, Generic Model Views folder had been created to collect BOM structure diagrams.

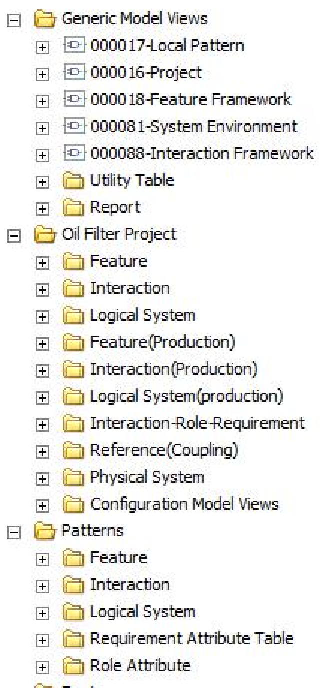
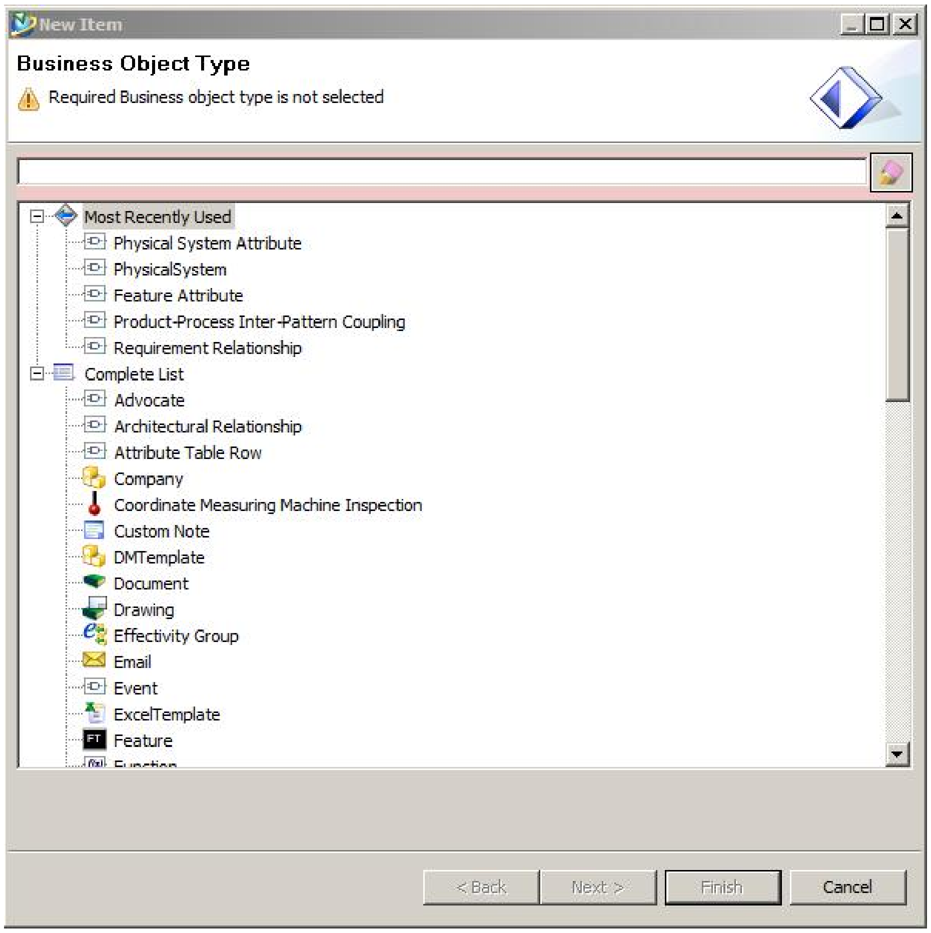

Generic Model Views
000017-Local Pattern
000016-Project
000018-Feature Framework
000081-System Environment
000088-Interaction Framework
Utility Table
Report
Oil Filter Project
Feature
Interaction
Logical System
Feature(Production)
Interaction(Production)
Logical System(production)
Interaction-Role-Requirement
Reference(Coupling)
Physical System
Configuration Model Views
Patterns
Feature
Interaction
Logical System
Requirement Attribute Table
Role Attribute

Figure 3.25. Created folders for project data management

An Item is a structure of related objects that represents products, parts, components, or systems engineering logical blocks. Items or item revisions as the fundamental data objects are used to manage information in Teamcenter. Items are able to contain other data objects which include items and folders. In this research, we mapped all specialized S* Metamodel elements into items in Teamcenter platform. In this case, creating either an S* Metamodel element or an item in Teamcenter are similar. When clicked "Item" in the "New" tab, the following dialog box had been shown. In this dialog box, the users allowed to pick a default or customer defined item type. The mapping in BMIDE enabled to select those specialized S* Metamodel elements as customer defined items in this window.


New Item

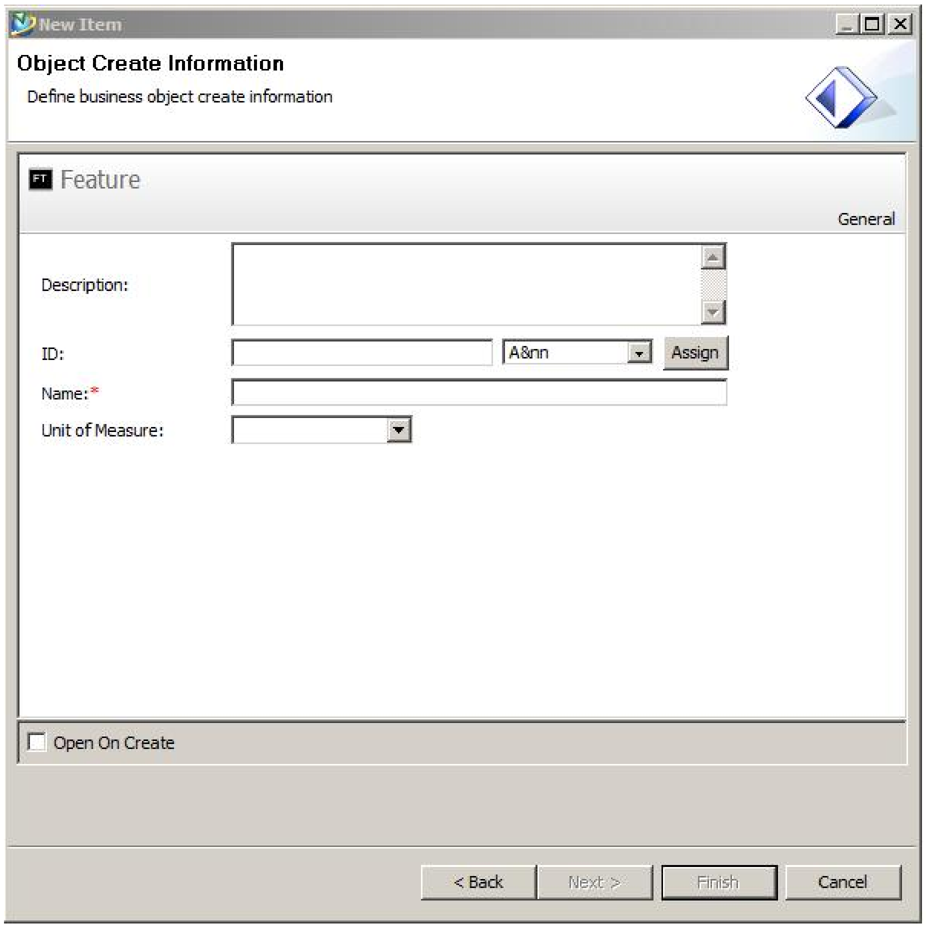
Business Object Type

Required Business object type is not selected

- Most Recently Used
  - Physical System Attribute
  - PhysicalSystem
  - Feature Attribute
  - Product-Process Inter-Pattern Coupling
  - Requirement Relationship
- Complete List
  - Advocate
  - Architectural Relationship
  - Attribute Table Row
  - Company
  - Coordinate Measuring Machine Inspection
  - Custom Note
  - DMTemplate
  - Document
  - Drawing
  - Effectivity Group
  - Email
  - Event
  - ExcelTemplate
  - Feature

Figure 3.26. Creating a new business object window

This is an example of creating an S* Metamodel element "Feature" in general production pattern feature folder. Once created this feature item in Teamcenter interface, defined icon and naming rule in BMIDE are automatically showed in the dialog box. In addition, the name and description of this item could be defined by users.


New Item

Object Create Information

Define business object create information

Feature

General

Description:	
ID:	A&nn
Name:*	
Unit of Measure:	

☐ Open On Create

Figure 3.27. Creating a sample Systematica class in new business object information defining window

A set of features, interactions, and logical systems included in S* Metamodel elements were created in folders, these features, interactions, and logical systems were summarized by general production pattern. In this case, features were general descriptions of stakeholders needs in a production, and interactions and logical systems were summarized based on features. Because of successful configuration in BMIDE, Features, Interactions, and Logical Systems all carried special icons and sequence IDs.

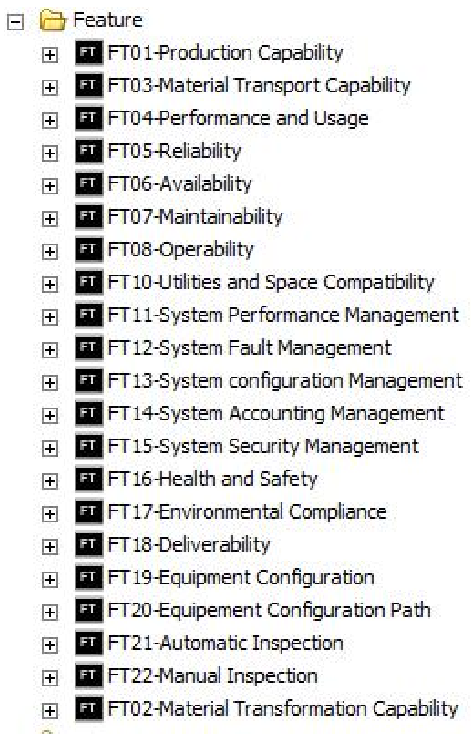

Feature
FT01-Production Capability
FT03-Material Transport Capability
FT04-Performance and Usage
FT05-Reliability
FT06-Availability
FT07-Maintainability
FT08-Operability
FT10-Utilities and Space Compatibility
FT11-System Performance Management
FT12-System Fault Management
FT13-System configuration Management
FT14-System Accounting Management
FT15-System Security Management
FT16-Health and Safety
FT17-Environmental Compliance
FT18-Deliverability
FT19-Equipment Configuration
FT20-Equipement Configuration Path
FT21-Automatic Inspection
FT22-Manual Inspection
FT02-Material Transformation Capability

Figure 3.28. Created feature items in feature folder

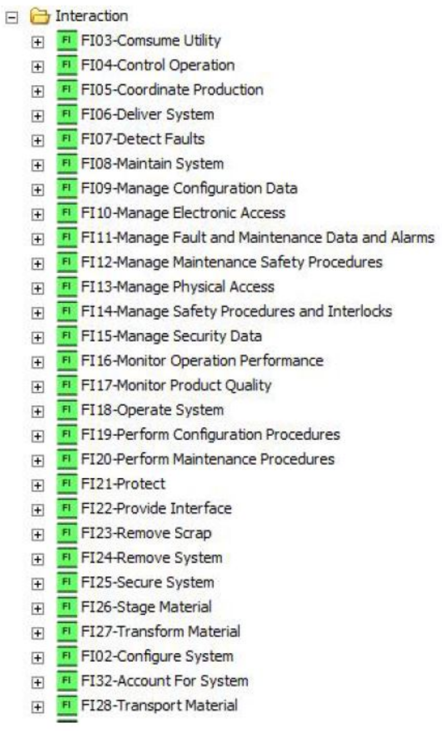

Interaction
FI03-Comsume Utility
FI04-Control Operation
FI05-Coordinate Production
FI06-Deliver System
FI07-Detect Faults
FI08-Maintain System
FI09-Manage Configuration Data
FI10-Manage Electronic Access
FI11-Manage Fault and Maintenance Data and Alarms
FI12-Manage Maintenance Safety Procedures
FI13-Manage Physical Access
FI14-Manage Safety Procedures and Interlocks
FI15-Manage Security Data
FI16-Monitor Operation Performance
FI17-Monitor Product Quality
FI18-Operate System
FI19-Perform Configuration Procedures
FI20-Perform Maintenance Procedures
FI21-Protect
FI22-Provide Interface
FI23-Remove Scrap
FI24-Remove System
FI25-Secure System
FI26-Stage Material
FI27-Transform Material
FI02-Configure System
FI32-Account For System
FI28-Transport Material

Figure 3.29. Created functional interaction items in functional interaction folder

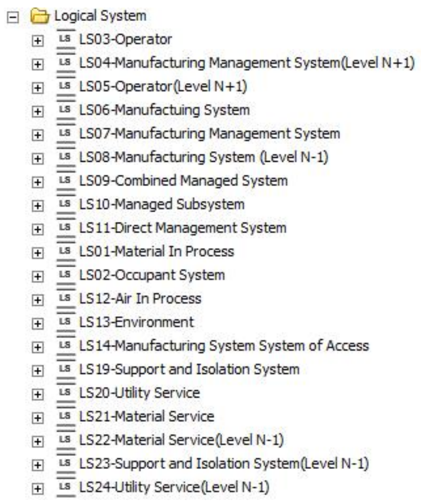
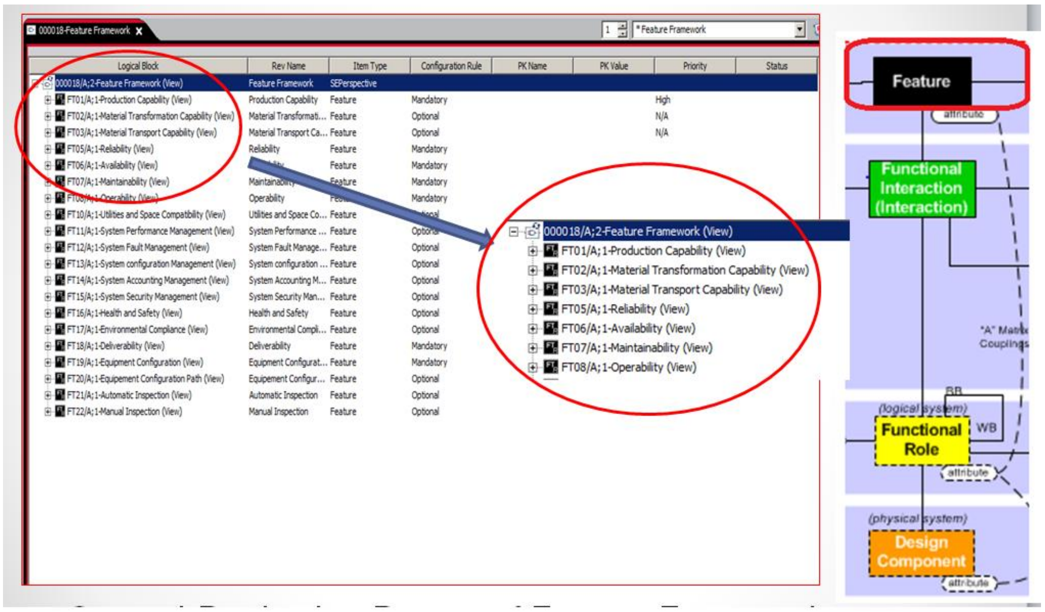

Logical System
LS03-Operator
LS04-Manufacturing Management System(Level N+1)
LS05-Operator(Level N+1)
LS06-Manufactuing System
LS07-Manufacturing Management System
LS08-Manufacturing System (Level N-1)
LS09-Combined Managed System
LS10-Managed Subsystem
LS11-Direct Management System
LS01-Material In Process
LS02-Occupant System
LS12-Air In Process
LS13-Environment
LS14-Manufacturing System System of Access
LS19-Support and Isolation System
LS20-Utility Service
LS21-Material Service
LS22-Material Service(Level N-1)
LS23-Support and Isolation System(Level N-1)
LS24-Utility Service(Level N-1)

Figure 3.30. Created logical system items in logical systems folder

In this general production pattern, several “SEPerspective” items were created to represent logical models in S* Metamodel which store and manage a set of correspondent logical blocks. In the Teamcenter Systems Engineering application, many views were included, for example: logical block view, function view, and requirement view. A created logical model can be displayed in a Logical Block view automatically or a Structure Manager to create, view, and modify a BOM view. The advantages of using Logical Block views are: performing design solution alternatives, building logical decompositions, and building diagram logical decompositions.

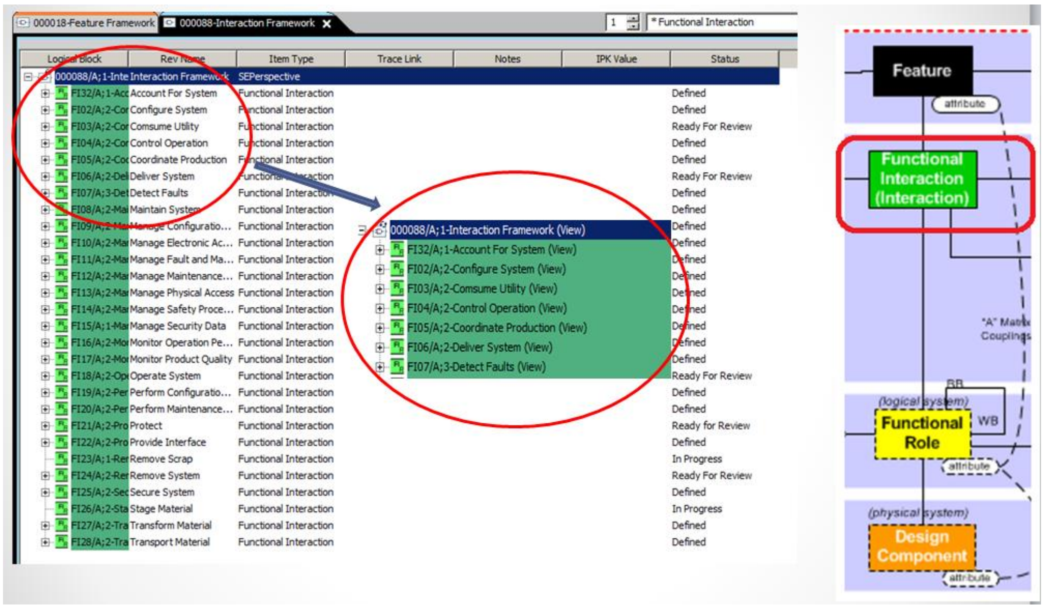
The following graphs are “Feature Framework” contains all the “Feature” logical blocks, “Interaction Framework” contains all the “Functional Interactions”, “logical blocks”, and “Systems Environment” contains all the “Logical System” logical blocks. Each view represents a class in Systematica Metamodel.



Logical Block	Rev Name	Item Type	Configuration Rule	PK Name	PK Value	Priority	Status
000018/A;2-Feature Framework (View)	Feature Framework	SEPerspective					
FT01/A;1-Production Capability (View)	Production Capability	Feature	Mandatory			High	
FT02/A;1-Material Transformation Capability (View)	Material Transformati...	Feature	Optional			N/A	
FT03/A;1-Material Transport Capability (View)	Material Transport Ca...	Feature	Optional			N/A	
FT05/A;1-Reliability (View)	Reliability	Feature	Mandatory				
FT06/A;1-Availability (View)	Availability	Feature	Mandatory				
FT07/A;1-Maintainability (View)	Maintainability	Feature	Mandatory				
FT08/A;1-Operability (View)	Operability	Feature	Mandatory				
FT10/A;1-Utilities and Space Compatibility (View)	Utilities and Space Co...	Feature	Optional				
FT11/A;1-System Performance Management (View)	System Performance ...	Feature	Optional				
FT12/A;1-System Fault Management (View)	System Fault Manage...	Feature	Optional				
FT13/A;1-System configuration Management (View)	System configuration ...	Feature	Optional				
FT14/A;1-System Accounting Management (View)	System Accounting M...	Feature	Optional				
FT15/A;1-System Security Management (View)	System Security Man...	Feature	Optional				
FT16/A;1-Health and Safety (View)	Health and Safety	Feature	Optional				
FT17/A;1-Environmental Compliance (View)	Environmental Compli...	Feature	Optional				
FT18/A;1-Deliverability (View)	Deliverability	Feature	Mandatory				
FT19/A;1-Equipment Configuration (View)	Equipment Configurat...	Feature	Mandatory				
FT20/A;1-Equipement Configuration Path (View)	Equipement Configur...	Feature	Optional				
FT21/A;1-Automatic Inspection (View)	Automatic Inspection	Feature	Optional				
FT22/A;1-Manual Inspection (View)	Manual Inspection	Feature	Optional				

000018/A;2-Feature Framework (View)
- FT01/A;1-Production Capability (View)
- FT02/A;1-Material Transformation Capability (View)
- FT03/A;1-Material Transport Capability (View)
- FT05/A;1-Reliability (View)
- FT06/A;1-Availability (View)
- FT07/A;1-Maintainability (View)
- FT08/A;1-Operability (View)

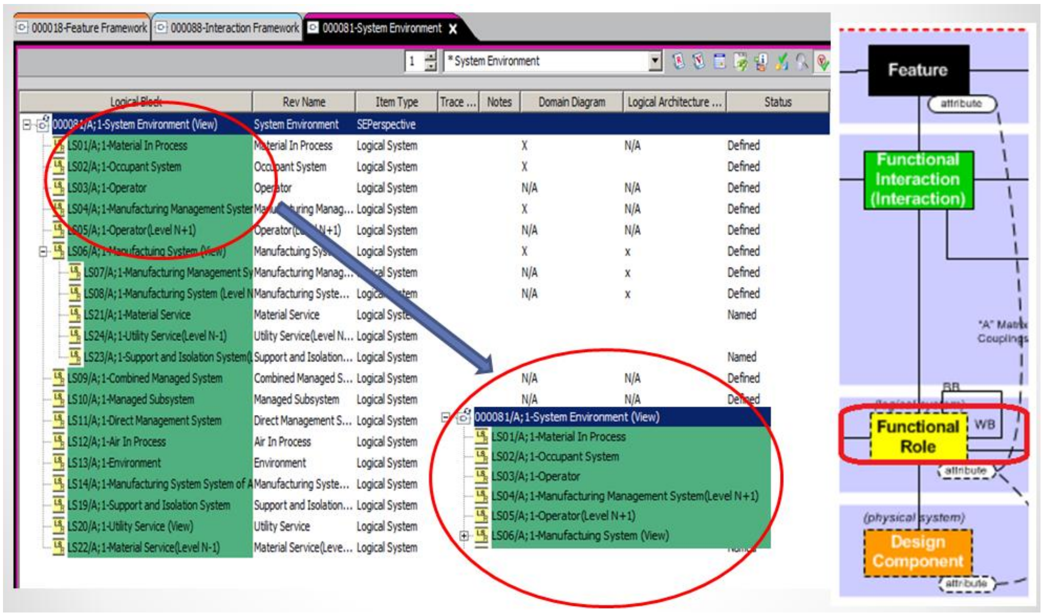
Figure 3.31. Feature framework in BOM view



Logical Block	Rev Name	Item Type	Trace Link	Notes	IPK Value	Status
000088/A;1-Inte	Interaction Framework	SEPerspective				
FI32/A;1-Acc	Account For System	Functional Interaction				Defined
FI02/A;2-Cor	Configure System	Functional Interaction				Defined
FI03/A;2-Cor	Comsume Utility	Functional Interaction				Ready For Review
FI04/A;2-Cor	Control Operation	Functional Interaction				Defined
FI05/A;2-Coc	Coordinate Production	Functional Interaction				Defined
FI06/A;2-Del	Deliver System	Functional Interaction				Ready For Review
FI07/A;3-Det	Detect Faults	Functional Interaction				Defined
FI08/A;2-Mai	Maintain System	Functional Interaction				Defined
FI09/A;2-Mai	Manage Configuratio...	Functional Interaction				Defined
FI10/A;2-Mai	Manage Electronic Ac...	Functional Interaction				Defined
FI11/A;2-Mai	Manage Fault and Ma...	Functional Interaction				Defined
FI12/A;2-Mai	Manage Maintenance...	Functional Interaction				Defined
FI13/A;2-Mai	Manage Physical Access	Functional Interaction				Defined
FI14/A;2-Mai	Manage Safety Proce...	Functional Interaction				Defined
FI15/A;1-Mai	Manage Security Data	Functional Interaction				Defined
FI16/A;2-Moi	Monitor Operation Pe...	Functional Interaction				Defined
FI17/A;2-Moi	Monitor Product Quality	Functional Interaction				Defined
FI18/A;2-Op	Operate System	Functional Interaction				Ready For Review
FI19/A;2-Per	Perform Configuratio...	Functional Interaction				Defined
FI20/A;2-Per	Perform Maintenance...	Functional Interaction				Defined
FI21/A;2-Pro	Protect	Functional Interaction				Ready for Review
FI22/A;2-Pro	Provide Interface	Functional Interaction				Defined
FI23/A;1-Rer	Remove Scrap	Functional Interaction				In Progress
FI24/A;2-Rer	Remove System	Functional Interaction				Ready For Review
FI25/A;2-Sec	Secure System	Functional Interaction				Defined
FI26/A;2-Sta	Stage Material	Functional Interaction				In Progress
FI27/A;2-Tra	Transform Material	Functional Interaction				Defined
FI28/A;2-Tra	Transport Material	Functional Interaction				Defined

000088/A;1-Interaction Framework (View)
- FI32/A;1-Account For System (View)
- FI02/A;2-Configure System (View)
- FI03/A;2-Comsume Utility (View)
- FI04/A;2-Control Operation (View)
- FI05/A;2-Coordinate Production (View)
- FI06/A;2-Deliver System (View)
- FI07/A;3-Detect Faults (View)

Figure 3.32. Interaction framework in BOM view

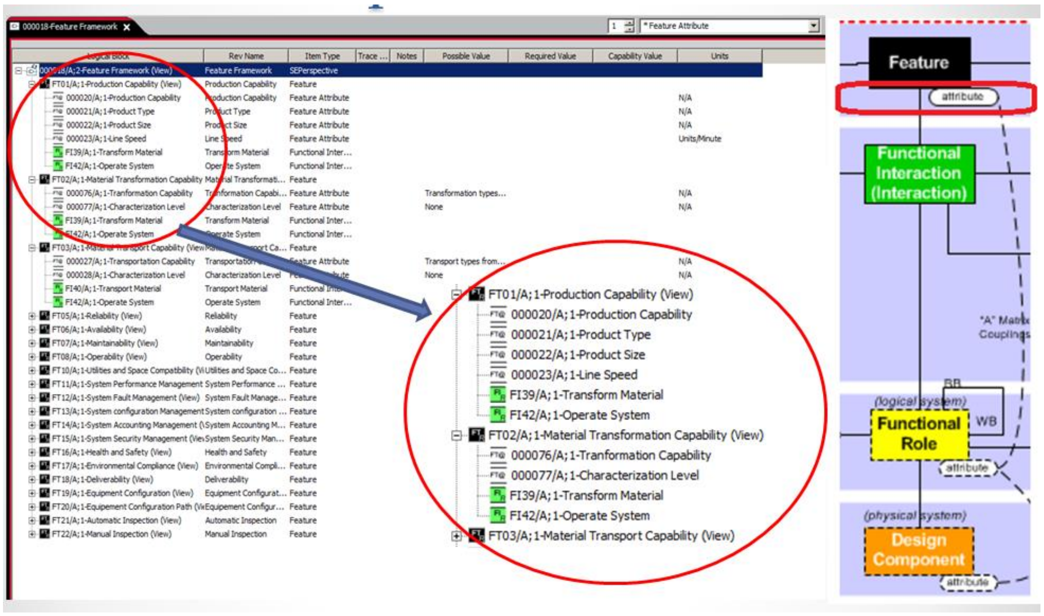


Logical Block	Rev Name	Item Type	Trace ...	Notes	Domain Diagram	Logical Architecture ...	Status
000081/A;1-System Environment (View)	System Environment	SEPerspective					
LS01/A;1-Material In Process	Material In Process	Logical System			X	N/A	Defined
LS02/A;1-Occupant System	Occupant System	Logical System			X		Defined
LS03/A;1-Operator	Operator	Logical System			N/A	N/A	Defined
LS04/A;1-Manufacturing Management Syster	Manufacturing Manag...	Logical System			X	N/A	Defined
LS05/A;1-Operator(Level N+1)	Operator(Level N+1)	Logical System			N/A	N/A	Defined
LS06/A;1-Manufactuing System (View)	Manufactuing Sys...	Logical System			X	x	Defined
LS07/A;1-Manufacturing Management Sy	Manufacturing Manag...	Logical System			N/A	x	Defined
LS08/A;1-Manufacturing System (Level N	Manufacturing Syste...	Logical System			N/A	x	Defined
LS21/A;1-Material Service	Material Service	Logical System					Named
LS24/A;1-Utility Service(Level N-1)	Utility Service(Level N...	Logical System					
LS23/A;1-Support and Isolation System(	Support and Isolation...	Logical System					Named
LS09/A;1-Combined Managed System	Combined Managed S...	Logical System			N/A	N/A	Defined
LS10/A;1-Managed Subsystem	Managed Subsystem	Logical System			N/A	N/A	Defined
LS11/A;1-Direct Management System	Direct Management S...	Logical System					
LS12/A;1-Air In Process	Air In Process	Logical System					
LS13/A;1-Environment	Environment	Logical System					
LS14/A;1-Manufacturing System System of A	Manufacturing Syste...	Logical System					
LS19/A;1-Support and Isolation System	Support and Isolation...	Logical System					
LS20/A;1-Utility Service (View)	Utility Service	Logical System					
LS22/A;1-Material Service(Level N-1)	Material Service(Leve...	Logical System					

000081/A;1-System Environment (View)
- LS01/A;1-Material In Process
- LS02/A;1-Occupant System
- LS03/A;1-Operator
- LS04/A;1-Manufacturing Management System(Level N+1)
- LS05/A;1-Operator(Level N+1)
- LS06/A;1-Manufactuing System (View)

Figure 3.33. System environment in BOM view

The column configuration box helps to select a set of saved column configuration list based on different views, and those columns represent customer properties of correspondent objects which defined by administrator in BMIDE. Currently, those columns are used to show both the value of relationships and the property value of classes in the BOM view by switching table display control, which is an explicit and simple way to present. In addition, Teamcenter used tree structures to build relationships between different hierarchies, by expanding or collapsing nodes to view an appropriate data in tree structures. These tree structures shows an explicit up-down structure and made easier for users to trace the changes during processes.


000018-Feature Framework

1 *Feature Attribute

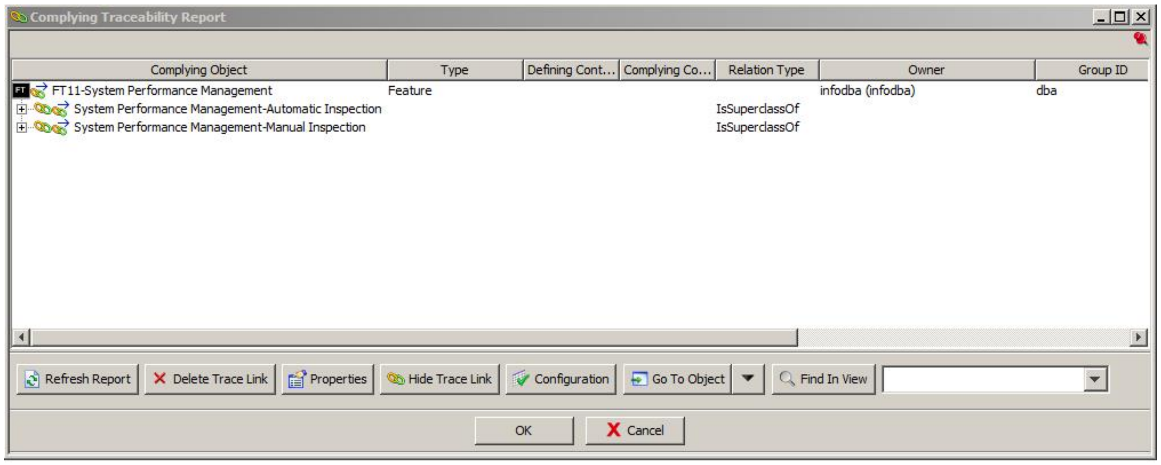
Logical Block	Rev Name	Item Type	Trace ...	Notes	Possible Value	Required Value	Capability Value	Units
000018/A;2-Feature Framework (View)	Feature Framework	SEPerspective						
FT01/A;1-Production Capability (View)	Production Capability	Feature						
000020/A;1-Production Capability	Production Capability	Feature Attribute						N/A
000021/A;1-Product Type	Product Type	Feature Attribute						N/A
000022/A;1-Product Size	Product Size	Feature Attribute						N/A
000023/A;1-Line Speed	Line Speed	Feature Attribute						Units/Minute
FI39/A;1-Transform Material	Transform Material	Functional Inter...						
FI42/A;1-Operate System	Operate System	Functional Inter...						
FT02/A;1-Material Transformation Capability	Material Transformati...	Feature						
000076/A;1-Tranformation Capability	Tranformation Capabi...	Feature Attribute			Transformation types...			N/A
000077/A;1-Characterization Level	Characterization Level	Feature Attribute			None			N/A
FI39/A;1-Transform Material	Transform Material	Functional Inter...						
FI42/A;1-Operate System	Operate System	Functional Inter...						
FT03/A;1-Material Transport Capability (View	Material Transport Ca...	Feature						
000027/A;1-Transportation Capability	Transportation C...	Feature Attribute			Transport types from...			N/A
000028/A;1-Characterization Level	Characterization Level	Feature Attribute			None			N/A
FI40/A;1-Transport Material	Transport Material	Functional Inter...						
FI42/A;1-Operate System	Operate System	Functional Inter...						
FT05/A;1-Reliability (View)	Reliability	Feature						
FT06/A;1-Availability (View)	Availability	Feature						
FT07/A;1-Maintainability (View)	Maintainability	Feature						
FT08/A;1-Operability (View)	Operability	Feature						
FT10/A;1-Utilities and Space Compatibility (Vi	Utilities and Space Co...	Feature						
FT11/A;1-System Performance Management	System Performance ...	Feature						
FT12/A;1-System Fault Management (View)	System Fault Manage...	Feature						
FT13/A;1-System configuration Management	System configuration ...	Feature						
FT14/A;1-System Accounting Management (\	System Accounting M...	Feature						
FT15/A;1-System Security Management (Viev	System Security Man...	Feature						
FT16/A;1-Health and Safety (View)	Health and Safety	Feature						
FT17/A;1-Environmental Compliance (View)	Environmental Compli...	Feature						
FT18/A;1-Deliverability (View)	Deliverability	Feature						
FT19/A;1-Equipment Configuration (View)	Equipment Configurat...	Feature						
FT20/A;1-Equipement Configuration Path (Vi	Equipement Configur...	Feature						
FT21/A;1-Automatic Inspection (View)	Automatic Inspection	Feature						
FT22/A;1-Manual Inspection (View)	Manual Inspection	Feature						

FT01/A;1-Production Capability (View)
000020/A;1-Production Capability
000021/A;1-Product Type
000022/A;1-Product Size
000023/A;1-Line Speed
FI39/A;1-Transform Material
FI42/A;1-Operate System
FT02/A;1-Material Transformation Capability (View)
000076/A;1-Tranformation Capability
000077/A;1-Characterization Level
FI39/A;1-Transform Material
FI42/A;1-Operate System
FT03/A;1-Material Transport Capability (View)

Figure 3.34. Feature attributes and values in properties

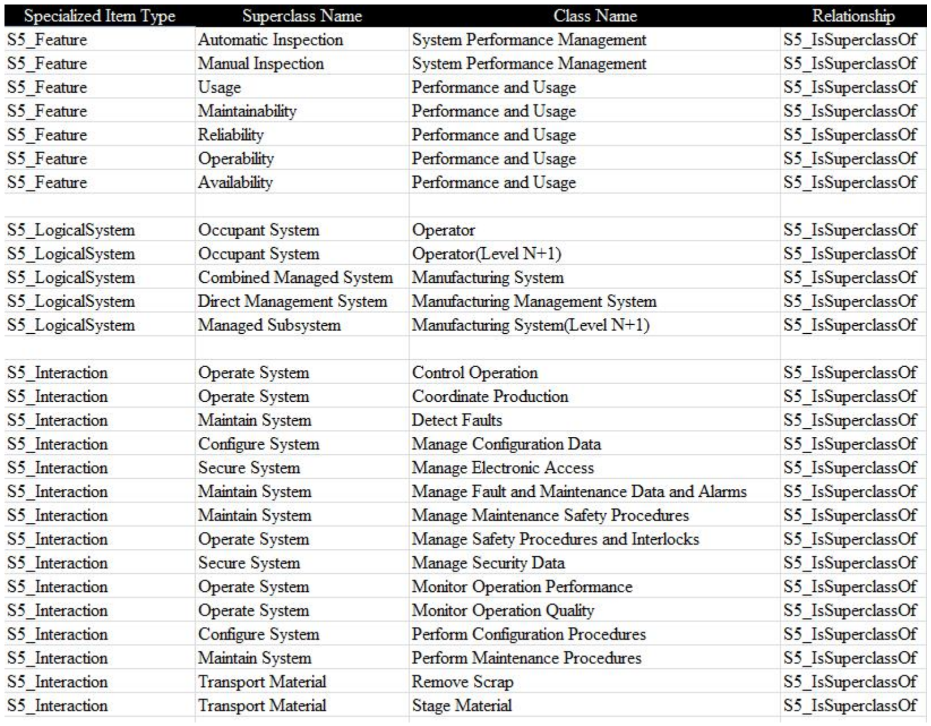
The feature attributes were directly created in the lower-level of features, in the BOM view, by accessing column configuration box to show related feature attributes properties. In those three BOM views, there were some invisible relationships connected by trace links among the same hierarchy. Some items are considered as superclass of other classes, the latter is a special case of the former. For example, if vehicle is considered as a superclass and then cars are classes. The mapping in the

BMIDE created a trace link relationship called “IsSuperclassOf” to define this relationship between class and superclass in Teamcenter systems engineering interface. Trace links provide treatabilities between structure elements, and traceability defines one object is precedent than another object. The figure below is an example of trace link report generated from feature BOM view.



Complying Object	Type	Defining Cont...	Complying Co...	Relation Type	Owner	Group ID
FT11-System Performance Management	Feature				infodba (infodba)	dba
System Performance Management-Automatic Inspection				IsSuperclassOf		
System Performance Management-Manual Inspection				IsSuperclassOf		

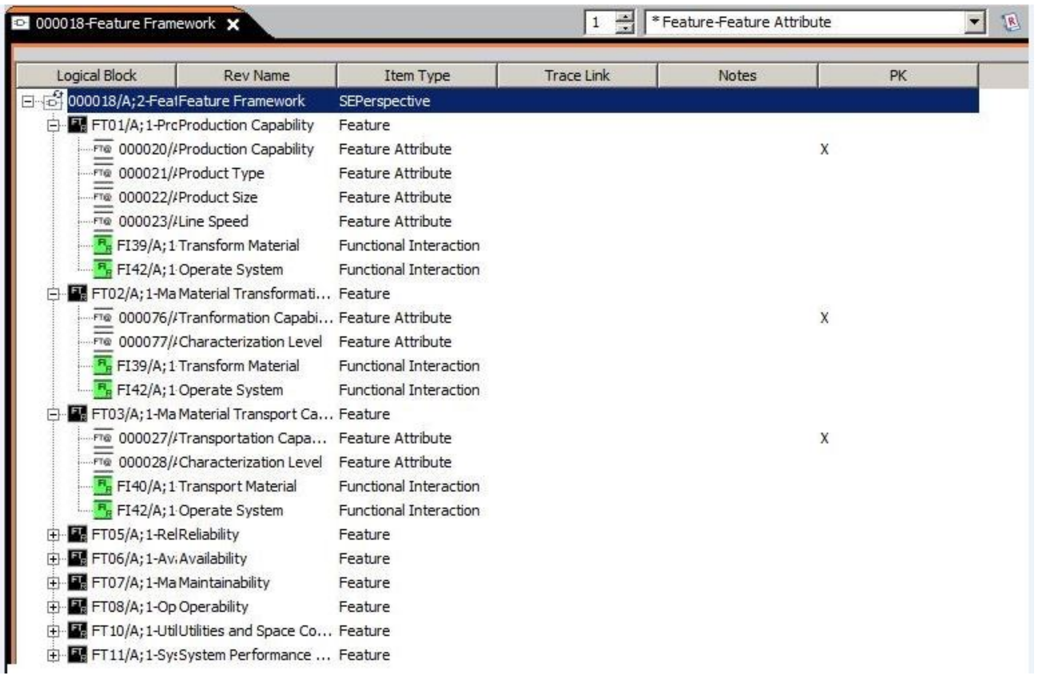
Figure 3.35. A sample trace link report



Specialized Item Type	Superclass Name	Class Name	Relationship
S5_Feature	Automatic Inspection	System Performance Management	S5_IsSuperclassOf
S5_Feature	Manual Inspection	System Performance Management	S5_IsSuperclassOf
S5_Feature	Usage	Performance and Usage	S5_IsSuperclassOf
S5_Feature	Maintainability	Performance and Usage	S5_IsSuperclassOf
S5_Feature	Reliability	Performance and Usage	S5_IsSuperclassOf
S5_Feature	Operability	Performance and Usage	S5_IsSuperclassOf
S5_Feature	Availability	Performance and Usage	S5_IsSuperclassOf
S5_LogicalSystem	Occupant System	Operator	S5_IsSuperclassOf
S5_LogicalSystem	Occupant System	Operator(Level N+1)	S5_IsSuperclassOf
S5_LogicalSystem	Combined Managed System	Manufacturing System	S5_IsSuperclassOf
S5_LogicalSystem	Direct Management System	Manufacturing Management System	S5_IsSuperclassOf
S5_LogicalSystem	Managed Subsystem	Manufacturing System(Level N+1)	S5_IsSuperclassOf
S5_Interaction	Operate System	Control Operation	S5_IsSuperclassOf
S5_Interaction	Operate System	Coordinate Production	S5_IsSuperclassOf
S5_Interaction	Maintain System	Detect Faults	S5_IsSuperclassOf
S5_Interaction	Configure System	Manage Configuration Data	S5_IsSuperclassOf
S5_Interaction	Secure System	Manage Electronic Access	S5_IsSuperclassOf
S5_Interaction	Maintain System	Manage Fault and Maintenance Data and Alarms	S5_IsSuperclassOf
S5_Interaction	Maintain System	Manage Maintenance Safety Procedures	S5_IsSuperclassOf
S5_Interaction	Operate System	Manage Safety Procedures and Interlocks	S5_IsSuperclassOf
S5_Interaction	Secure System	Manage Security Data	S5_IsSuperclassOf
S5_Interaction	Operate System	Monitor Operation Performance	S5_IsSuperclassOf
S5_Interaction	Operate System	Monitor Operation Quality	S5_IsSuperclassOf
S5_Interaction	Configure System	Perform Configuration Procedures	S5_IsSuperclassOf
S5_Interaction	Maintain System	Perform Maintenance Procedures	S5_IsSuperclassOf
S5_Interaction	Transport Material	Remove Scrap	S5_IsSuperclassOf
S5_Interaction	Transport Material	Stage Material	S5_IsSuperclassOf

Figure 3.36. The overall class and superclass summary

From the S* Metamodel, each class has relationships with other class. The relationship between feature and feature attribute, feature and interaction, interaction and logical system were created in BOM view. A primary key is a way to designate a unique identification of each record in the table. In the relationship between the two tables, the primary key has been used in one table to refer a specific record from another table. In this research, relationships between S* Metamodel classes were represented by creating primary keys in BOM views to show inter-connections. The trace from Feature to Physical Systems is a bidirectional relationship. This trace allows systems engineers to select appropriate features based on defined stakeholder needs or requirements, and then all the way to find involved physical systems by using primary keys. Moreover, using the primary key to build connect between different classes, it is easy to trace the relationship from component design to features. For example, most of commercial industries started a project with product designs. In this case, this trace helps engineers to figure out whether their product designs meet requirements from stakeholders or whether consistent or conflict with other engineering departments.

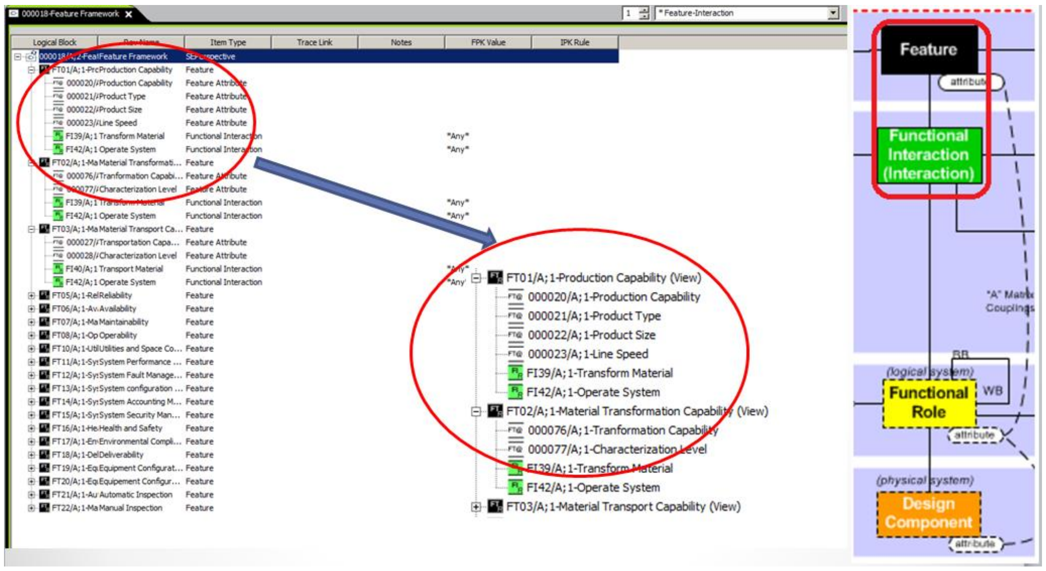


000018-Feature Framework

1 | *Feature-Feature Attribute

Logical Block	Rev Name	Item Type	Trace Link	Notes	PK
000018/A;2-Feat	Feature Framework	SEPerspective			
FT01/A;1-Prc	Production Capability	Feature			
000020/	Production Capability	Feature Attribute			X
000021/	Product Type	Feature Attribute			
000022/	Product Size	Feature Attribute			
000023/	Line Speed	Feature Attribute			
FI39/A;1	Transform Material	Functional Interaction			
FI42/A;1	Operate System	Functional Interaction			
FT02/A;1-Ma	Material Transformati…	Feature			
000076/	Tranformation Capabi…	Feature Attribute			X
000077/	Characterization Level	Feature Attribute			
FI39/A;1	Transform Material	Functional Interaction			
FI42/A;1	Operate System	Functional Interaction			
FT03/A;1-Ma	Material Transport Ca…	Feature			
000027/	Transportation Capa…	Feature Attribute			X
000028/	Characterization Level	Feature Attribute			
FI40/A;1	Transport Material	Functional Interaction			
FI42/A;1	Operate System	Functional Interaction			
FT05/A;1-Rel	Reliability	Feature			
FT06/A;1-Av	Availability	Feature			
FT07/A;1-Ma	Maintainability	Feature			
FT08/A;1-Op	Operability	Feature			
FT10/A;1-Util	Utilities and Space Co…	Feature			
FT11/A;1-Sy	System Performance …	Feature			

Figure 3.37. Primary key in feature and feature attribute

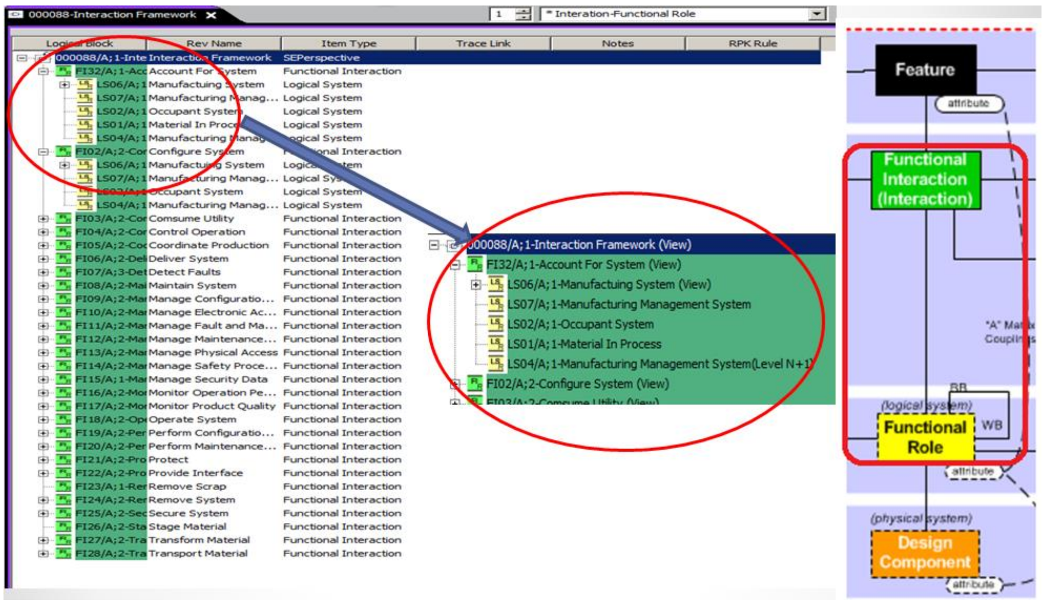


```
000018-Feature Framework                                            1   * Feature-Interaction

Logical Block            Rev Name                       Item Type               Trace Link   Notes   FPK Value   IPK Rule
000018/A;2-Fea Feature Framework                        SEPerspective
  FT01/A;1-Prc Production Capability                    Feature
    000020/A Production Capability                      Feature Attribute
    000021/A Product Type                               Feature Attribute
    000022/A Product Size                               Feature Attribute
    000023/A Line Speed                                 Feature Attribute
    FI39/A;1 Transform Material                         Functional Interaction                       *Any*
    FI42/A;1 Operate System                             Functional Interaction                       *Any*
  FT02/A;1-Ma Material Transformati...                  Feature
    000076/A Tranformation Capabi...                    Feature Attribute
    000077/A Characterization Level                     Feature Attribute
    FI39/A;1 Transform Material                         Functional Interaction                       *Any*
    FI42/A;1 Operate System                             Functional Interaction                       *Any*
  FT03/A;1-Ma Material Transport Ca...                  Feature
    000027/A Transportation Capa...                     Feature Attribute
    000028/A Characterization Level                     Feature Attribute
    FI40/A;1 Transport Material                         Functional Interaction                       *Any*
    FI42/A;1 Operate System                             Functional Interaction                       *Any*
  FT05/A;1-Rel Reliability                              Feature
  FT06/A;1-Av Availability                              Feature
  FT07/A;1-Ma Maintainability                           Feature
  FT08/A;1-Op Operability                               Feature
  FT10/A;1-Util Utilities and Space Co...               Feature
  FT11/A;1-Sy System Performance ...                    Feature
  FT12/A;1-Sy System Fault Manage...                    Feature
  FT13/A;1-Sy System configuration ...                  Feature
  FT14/A;1-Sy System Accounting M...                    Feature
  FT15/A;1-Sy System Security Man...                    Feature
  FT16/A;1-He Health and Safety                         Feature
  FT17/A;1-En Environmental Compli...                   Feature
  FT18/A;1-Del Deliverability                           Feature
  FT19/A;1-Eq Equipment Configurat...                   Feature
  FT20/A;1-Eq Equipement Configur...                    Feature
  FT21/A;1-Au Automatic Inspection                      Feature
  FT22/A;1-Ma Manual Inspection                         Feature
```

```
FT01/A;1-Production Capability (View)
  000020/A;1-Production Capability
  000021/A;1-Product Type
  000022/A;1-Product Size
  000023/A;1-Line Speed
  FI39/A;1-Transform Material
  FI42/A;1-Operate System
FT02/A;1-Material Transformation Capability (View)
  000076/A;1-Tranformation Capability
  000077/A;1-Characterization Level
  FI39/A;1-Transform Material
  FI42/A;1-Operate System
FT03/A;1-Material Transport Capability (View)
```

Figure 3.38. The connection of feature and Interaction represented by feature primary key value and interaction primary key rule

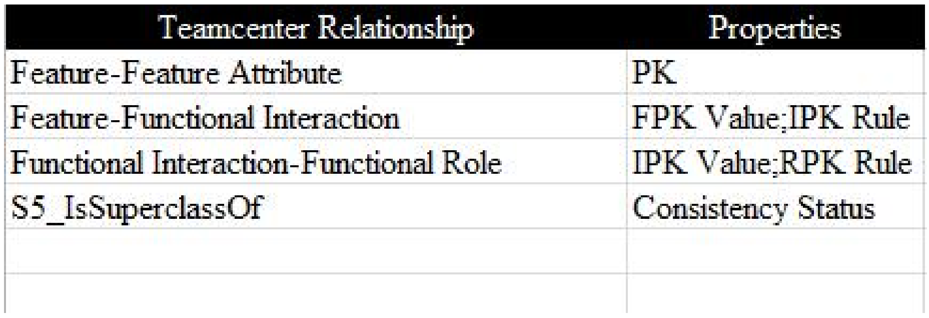


```
000088-Interaction Framework                                        1   * Interation-Functional Role

Logical Block      Rev Name                      Item Type               Trace Link   Notes   RPK Rule
000088/A;1-Inte    Interaction Framework         SEPerspective
  FI32/A;1-Acc     Account For System            Functional Interaction
    LS06/A;1       Manufactuing System           Logical System
    LS07/A;1       Manufacturing Manag...        Logical System
    LS02/A;1       Occupant System               Logical System
    LS01/A;1       Material In Process           Logical System
    LS04/A;1       Manufacturing Manag...        Logical System
  FI02/A;2-Cor     Configure System              Functional Interaction
    LS06/A;1       Manufactuing System           Logical System
    LS07/A;1       Manufacturing Manag...        Logical System
    LS02/A;1       Occupant System               Logical System
    LS04/A;1       Manufacturing Manag...        Logical System
  FI03/A;2-Cor     Comsume Utility               Functional Interaction
  FI04/A;2-Cor     Control Operation             Functional Interaction
  FI05/A;2-Coc     Coordinate Production         Functional Interaction
  FI06/A;2-Del     Deliver System                Functional Interaction
  FI07/A;3-Det     Detect Faults                 Functional Interaction
  FI08/A;2-Mai     Maintain System               Functional Interaction
  FI09/A;2-Mar     Manage Configuratio...        Functional Interaction
  FI10/A;2-Mar     Manage Electronic Ac...       Functional Interaction
  FI11/A;2-Mar     Manage Fault and Ma...        Functional Interaction
  FI12/A;2-Mar     Manage Maintenance...         Functional Interaction
  FI13/A;2-Mar     Manage Physical Access        Functional Interaction
  FI14/A;2-Mar     Manage Safety Proce...        Functional Interaction
  FI15/A;1-Mar     Manage Security Data          Functional Interaction
  FI16/A;2-Mor     Monitor Operation Pe...       Functional Interaction
  FI17/A;2-Mor     Monitor Product Quality       Functional Interaction
  FI18/A;2-Op      Operate System                Functional Interaction
  FI19/A;2-Per     Perform Configuratio...       Functional Interaction
  FI20/A;2-Per     Perform Maintenance...        Functional Interaction
  FI21/A;2-Pro     Protect                       Functional Interaction
  FI22/A;2-Pro     Provide Interface             Functional Interaction
  FI23/A;1-Rer     Remove Scrap                  Functional Interaction
  FI24/A;2-Rer     Remove System                 Functional Interaction
  FI25/A;2-Sec     Secure System                 Functional Interaction
  FI26/A;2-Sta     Stage Material                Functional Interaction
  FI27/A;2-Tra     Transform Material            Functional Interaction
  FI28/A;2-Tra     Transport Material            Functional Interaction
```

```
000088/A;1-Interaction Framework (View)
  FI32/A;1-Account For System (View)
    LS06/A;1-Manufactuing System (View)
    LS07/A;1-Manufacturing Management System
    LS02/A;1-Occupant System
    LS01/A;1-Material In Process
    LS04/A;1-Manufacturing Management System(Level N+1)
  FI02/A;2-Configure System (View)
  FI03/A;2-Comsume Utility (View)
```

Figure 3.39. The connection of interaction and logical systems represented by role primary key rule



Teamcenter Relationship	Properties
Feature-Feature Attribute	PK
Feature-Functional Interaction	FPK Value;IPK Rule
Functional Interaction-Functional Role	IPK Value;RPK Rule
S5_IsSuperclassOf	Consistency Status

Figure 3.40. The summaries of Teamcenter relationships and correlative primary keys

Teamcenter also have ability to configure the Microsoft office Visio diagram integration. Sometimes, building blocks help users to illustrate hierarchical relationships of elements easier. For example, the following feature overview and interaction overview graphs used block diagram to show the traceability between same hierarchical classes. Since the live integration of Teamcenter and Microsoft Visio, diagrams can be generated from Teamcenter based on using appropriate diagrams templates.

The Feature overview diagram shows all the features, feature attributes, and relationships between features' superclass and classes. The Interaction overview diagram shows the Interactions in general production and relationships between interactions superclass and classes.

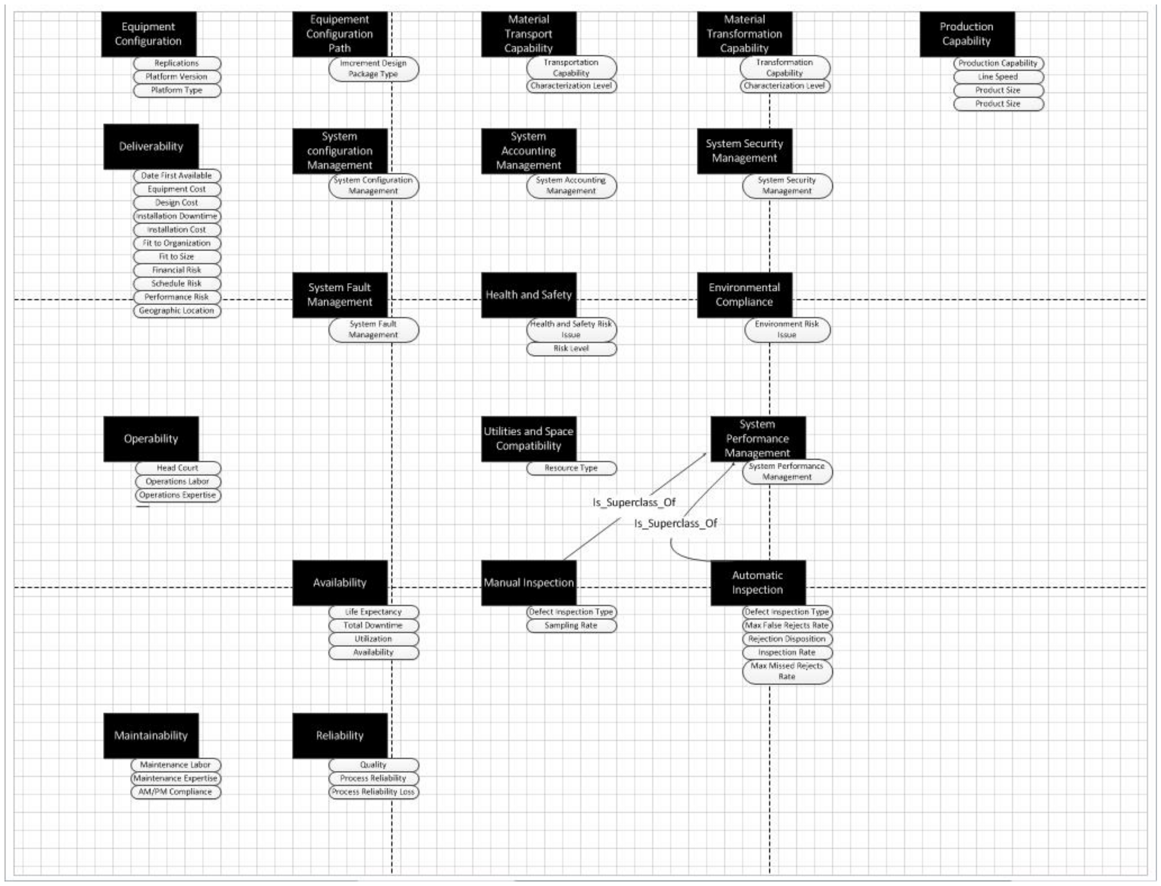


Figure 3.41. Feature overview diagram in Microsoft Visio

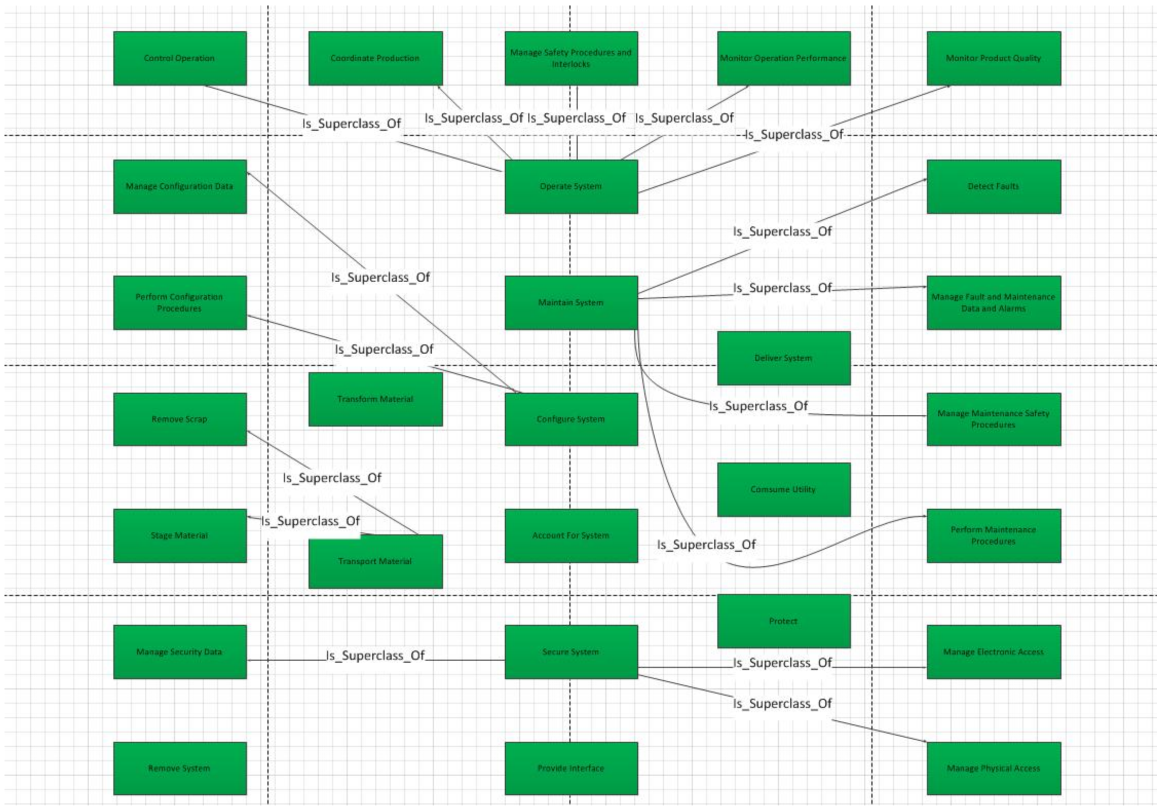


Figure 3.42. Interaction overview diagram in Microsoft Visio

3.4 Configuration System: Oil Filter End Seal Compression

Oil filters are designed to remove containment in an engine oil, transmission oil, lubricating oil, or hydraulic oil. The major use of oil filter is in automotive combustion engine. In this case, manufacturing of the oil filter selected to illustrate general production pattern can be specialized.

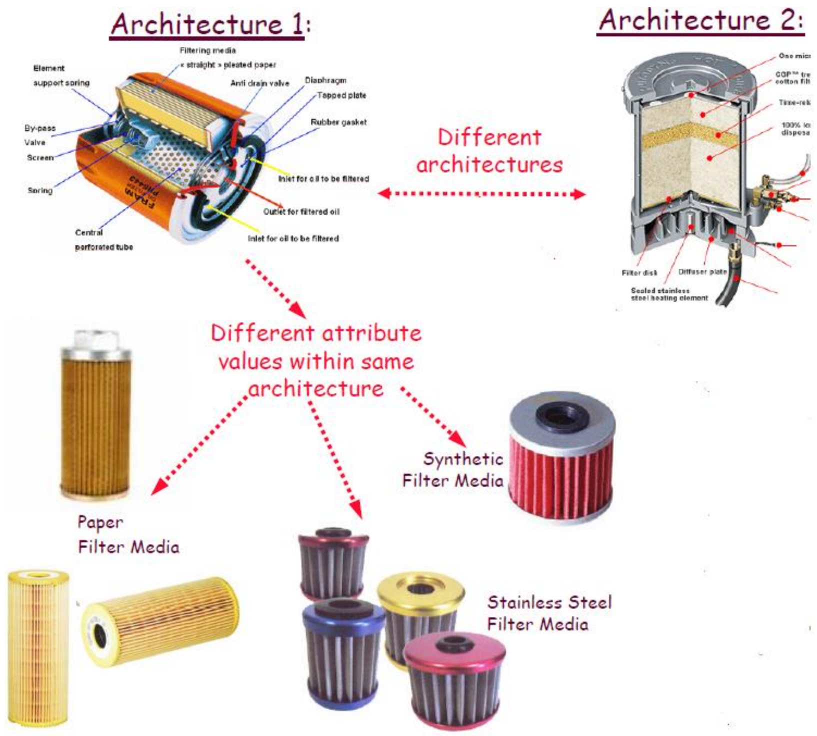
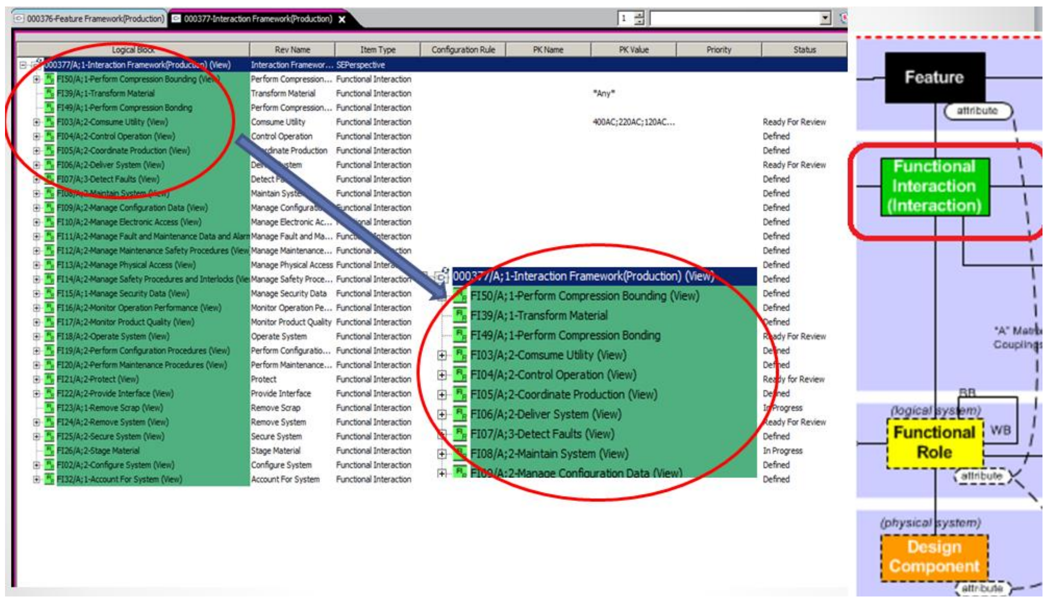


Figure 3.43. Oil filter physical architectures [40]

The end seal bonding compression production selected to demonstrate this oil filter specialization research. In an enterprise case, usually, systems engineers focus on interactions with a starting point. specialized from the S* Pattern, the interaction to produce this end seal compression bonding should be classified into “Transform Material”, in this research, a new interaction called “Perform Compression Bonding” was created to represent the interaction of which components are bonded using compression forces.



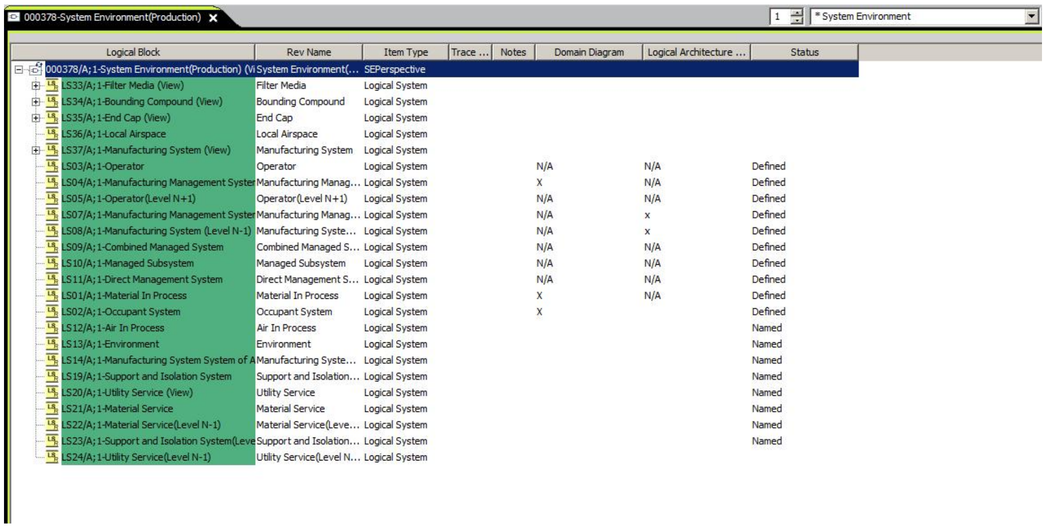
000376-Feature Framework(Production) | 000377-Interaction Framework(Production)

Logical Block	Rev Name	Item Type	Configuration Rule	PK Name	PK Value	Priority	Status
000377/A;1-Interaction Framework(Production) (View)	Interaction Framewor...	SEPerspective					
FI50/A;1-Perform Compression Bounding (View)	Perform Compression...	Functional Interaction					
FI39/A;1-Transform Material	Transform Material	Functional Interaction			"Any"		
FI49/A;1-Perform Compression Bonding	Perform Compression...	Functional Interaction					
FI03/A;2-Comsume Utility (View)	Comsume Utility	Functional Interaction			400AC;220AC;120AC...		Ready For Review
FI04/A;2-Control Operation (View)	Control Operation	Functional Interaction					Defined
FI05/A;2-Coordinate Production (View)	Coordinate Production	Functional Interaction					Defined
FI06/A;2-Deliver System (View)	Deliver System	Functional Interaction					Ready For Review
FI07/A;3-Detect Faults (View)	Detect Faults	Functional Interaction					Defined
FI08/A;2-Maintain System (View)	Maintain System	Functional Interaction					Defined
FI09/A;2-Manage Configuration Data (View)	Manage Configurat...	Functional Interaction					Defined
FI10/A;2-Manage Electronic Access (View)	Manage Electronic Ac...	Functional Interaction					Defined
FI11/A;2-Manage Fault and Maintenance Data and Alarm	Manage Fault and Ma...	Functional Interaction					Defined
FI12/A;2-Manage Maintenance Safety Procedures (View	Manage Maintenance...	Functional Interaction					Defined
FI13/A;2-Manage Physical Access (View)	Manage Physical Access	Functional Interaction					Defined
FI14/A;2-Manage Safety Procedures and Interlocks (Vie	Manage Safety Proce...	Functional Interaction					Defined
FI15/A;1-Manage Security Data (View)	Manage Security Data	Functional Interaction					Defined
FI16/A;2-Monitor Operation Performance (View)	Monitor Operation Pe...	Functional Interaction					Defined
FI17/A;2-Monitor Product Quality (View)	Monitor Product Quality	Functional Interaction					Defined
FI18/A;2-Operate System (View)	Operate System	Functional Interaction					Ready For Review
FI19/A;2-Perform Configuration Procedures (View)	Perform Configuratio...	Functional Interaction					Defined
FI20/A;2-Perform Maintenance Procedures (View)	Perform Maintenance...	Functional Interaction					Defined
FI21/A;2-Protect (View)	Protect	Functional Interaction					Ready for Review
FI22/A;2-Provide Interface (View)	Provide Interface	Functional Interaction					Defined
FI23/A;1-Remove Scrap (View)	Remove Scrap	Functional Interaction					In Progress
FI24/A;2-Remove System (View)	Remove System	Functional Interaction					Ready For Review
FI25/A;2-Secure System (View)	Secure System	Functional Interaction					Defined
FI26/A;2-Stage Material	Stage Material	Functional Interaction					In Progress
FI02/A;2-Configure System (View)	Configure System	Functional Interaction					Defined
FI32/A;1-Account For System (View)	Account For System	Functional Interaction					Defined

000377/A;1-Interaction Framework(Production) (View)
- FI50/A;1-Perform Compression Bounding (View)
- FI39/A;1-Transform Material
- FI49/A;1-Perform Compression Bonding
- FI03/A;2-Comsume Utility (View)
- FI04/A;2-Control Operation (View)
- FI05/A;2-Coordinate Production (View)
- FI06/A;2-Deliver System (View)
- FI07/A;3-Detect Faults (View)
- FI08/A;2-Maintain System (View)
- FI09/A;2-Manage Configuration Data (View)

Figure 3.44. Specialized functional interaction framework

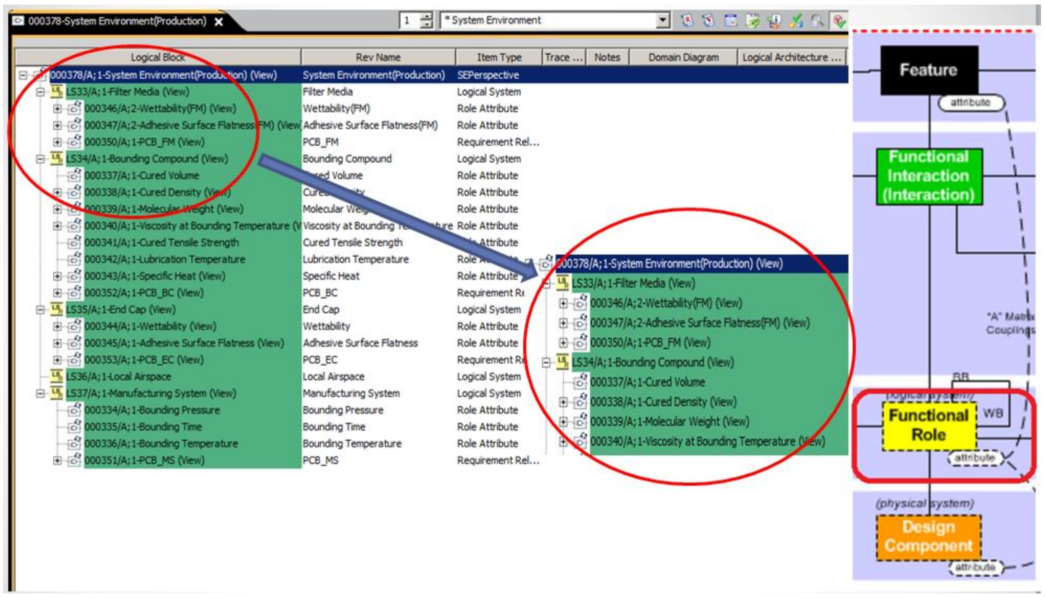
Once the interaction was defined, the logical systems were modeled quickly. Filter media, bonding compound, and end cap are necessary parts for production. Local airspace, manufacturing system and other manufacturing logical systems hesitated from general production pattern directly. Oil filter compression not only has to consider production feature, interactions and logical systems, but also need to consider manufacturing systems. Functional role attributes created and attached to logical systems, those role attributes were considered as technical valuation of production logical systems has been shown in the figure of System Environment (Production).



000378-System Environment(Production)　1　* System Environment

Logical Block	Rev Name	Item Type	Trace ...	Notes	Domain Diagram	Logical Architecture ...	Status
000378/A;1-System Environment(Production) (Vi	System Environment(...	SEPerspective					
LS33/A;1-Filter Media (View)	Filter Media	Logical System					
LS34/A;1-Bounding Compound (View)	Bounding Compound	Logical System					
LS35/A;1-End Cap (View)	End Cap	Logical System					
LS36/A;1-Local Airspace	Local Airspace	Logical System					
LS37/A;1-Manufacturing System (View)	Manufacturing System	Logical System					
LS03/A;1-Operator	Operator	Logical System			N/A	N/A	Defined
LS04/A;1-Manufacturing Management Syster	Manufacturing Manag...	Logical System			X	N/A	Defined
LS05/A;1-Operator(Level N+1)	Operator(Level N+1)	Logical System			N/A	N/A	Defined
LS07/A;1-Manufacturing Management Syster	Manufacturing Manag...	Logical System			N/A	x	Defined
LS08/A;1-Manufacturing System (Level N-1)	Manufacturing Syste...	Logical System			N/A	x	Defined
LS09/A;1-Combined Managed System	Combined Managed S...	Logical System			N/A	N/A	Defined
LS10/A;1-Managed Subsystem	Managed Subsystem	Logical System			N/A	N/A	Defined
LS11/A;1-Direct Management System	Direct Management S...	Logical System			N/A	N/A	Defined
LS01/A;1-Material In Process	Material In Process	Logical System			X	N/A	Defined
LS02/A;1-Occupant System	Occupant System	Logical System			X		Defined
LS12/A;1-Air In Process	Air In Process	Logical System					Named
LS13/A;1-Environment	Environment	Logical System					Named
LS14/A;1-Manufacturing System System of A	Manufacturing Syste...	Logical System					Named
LS19/A;1-Support and Isolation System	Support and Isolation...	Logical System					Named
LS20/A;1-Utility Service (View)	Utility Service	Logical System					Named
LS21/A;1-Material Service	Material Service	Logical System					Named
LS22/A;1-Material Service(Level N-1)	Material Service(Leve...	Logical System					Named
LS23/A;1-Support and Isolation System(Leve	Support and Isolation...	Logical System					Named
LS24/A;1-Utility Service(Level N-1)	Utility Service(Level N...	Logical System					

Figure 3.45. Specialized system environment



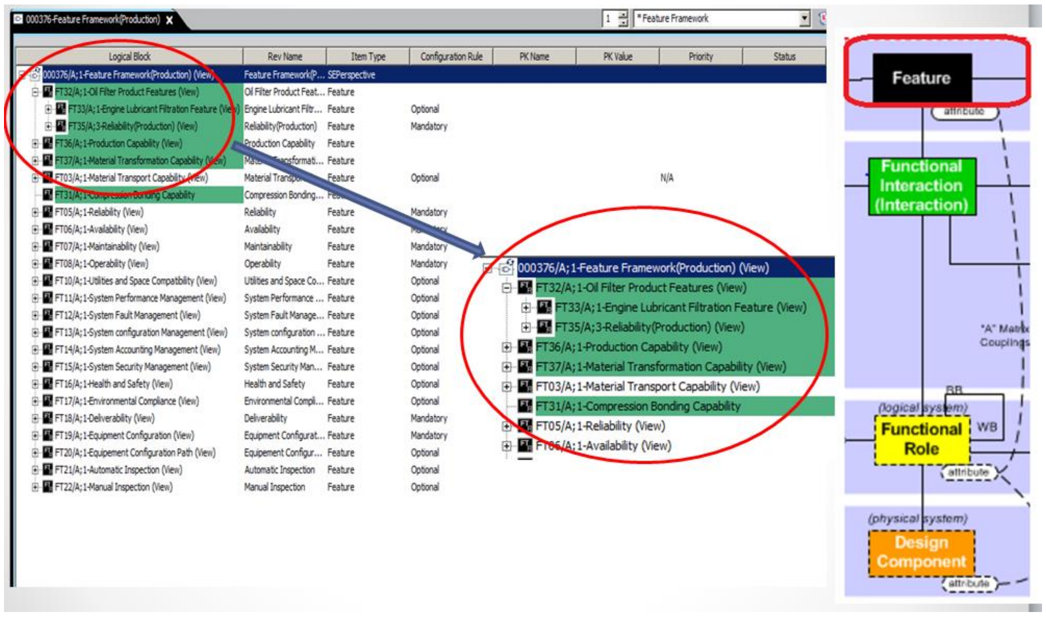
000378-System Environment(Production)　1　* System Environment

Logical Block	Rev Name	Item Type	Trace ...	Notes	Domain Diagram	Logical Architecture ...
000378/A;1-System Environment(Production) (View)	System Environment(Production)	SEPerspective				
LS33/A;1-Filter Media (View)	Filter Media	Logical System				
000346/A;2-Wettability(FM) (View)	Wettability(FM)	Role Attribute				
000347/A;2-Adhesive Surface Flatness(FM) (View	Adhesive Surface Flatness(FM)	Role Attribute				
000350/A;1-PCB_FM (View)	PCB_FM	Requirement Rel...				
LS34/A;1-Bounding Compound (View)	Bounding Compound	Logical System				
000337/A;1-Cured Volume	ed Volume	Role Attribute				
000338/A;1-Cured Density (View)	Cure	Role Attribute				
000339/A;1-Molecular Weight (View)	Molecular Weig	Role Attribute				
000340/A;1-Viscosity at Bounding Temperature (V	Viscosity at Bounding Te ature	Role Attribute				
000341/A;1-Cured Tensile Strength	Cured Tensile Strength	Attribute				
000342/A;1-Lubrication Temperature	Lubrication Temperature	Role				
000343/A;1-Specific Heat (View)	Specific Heat	Role Attribute				
000352/A;1-PCB_BC (View)	PCB_BC	Requirement Re				
LS35/A;1-End Cap (View)	End Cap	Logical System				
000344/A;1-Wettability (View)	Wettability	Role Attribute				
000345/A;1-Adhesive Surface Flatness (View)	Adhesive Surface Flatness	Role Attribute				
000353/A;1-PCB_EC (View)	PCB_EC	Requirement Re				
LS36/A;1-Local Airspace	Local Airspace	Logical System				
LS37/A;1-Manufacturing System (View)	Manufacturing System	Logical System				
000334/A;1-Bounding Pressure	Bounding Pressure	Role Attribute				
000335/A;1-Bounding Time	Bounding Time	Role Attribute				
000336/A;1-Bounding Temperature	Bounding Temperature	Role Attribute				
000351/A;1-PCB_MS (View)	PCB_MS	Requirement Rel...				

000378/A;1-System Environment(Production) (View)
LS33/A;1-Filter Media (View)
000346/A;2-Wettability(FM) (View)
000347/A;2-Adhesive Surface Flatness(FM) (View)
000350/A;1-PCB_FM (View)
LS34/A;1-Bounding Compound (View)
000337/A;1-Cured Volume
000338/A;1-Cured Density (View)
000339/A;1-Molecular Weight (View)
000340/A;1-Viscosity at Bounding Temperature (View)

Figure 3.46. Specialized role attributes

Based on considering stakeholders needs and understanding of interactions and logical systems, production and manufacturing features were created. Material transformation capability, production capability features and other general production features for manufacturing should be populated from the S* Pattern directly. And a specialized oil filter production feature that contains engine lubricant filtration feature and reliability (production) features were created. Feature attribute based on stakeholders valuations under each feature were also created.

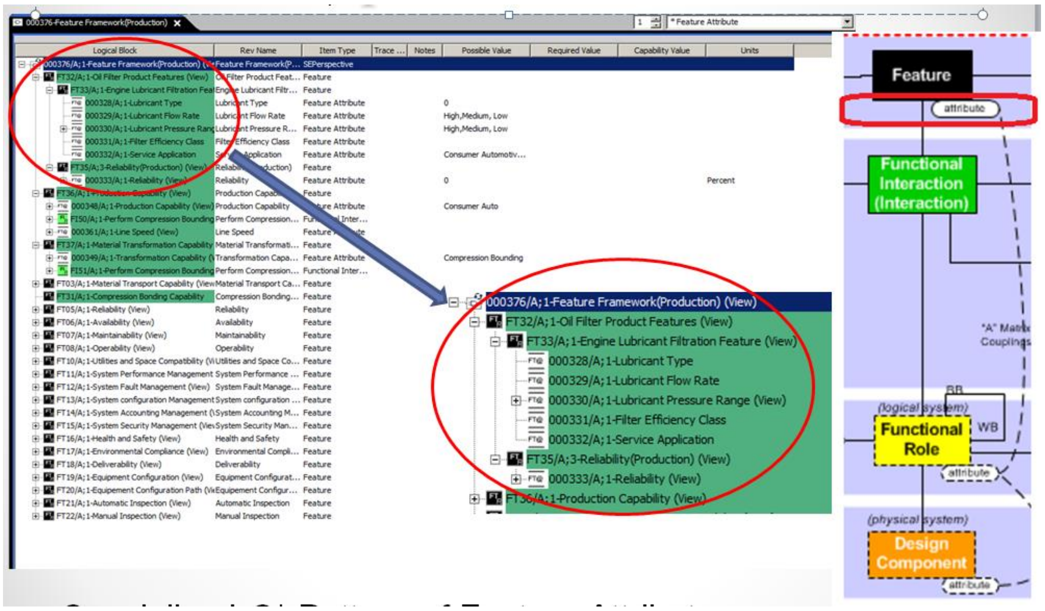


Logical Block	Rev Name	Item Type	Configuration Rule	PK Name	PK Value	Priority	Status
000376/A;1-Feature Framework(Production) (View)	Feature Framework(P...	SEPerspective					
FT32/A;1-Oil Filter Product Features (View)	Oil Filter Product Feat...	Feature					
FT33/A;1-Engine Lubricant Filtration Feature (View)	Engine Lubricant Filtr...	Feature	Optional				
FT35/A;3-Reliability(Production) (View)	Reliability(Production)	Feature	Mandatory				
FT36/A;1-Production Capability (View)	Production Capability	Feature					
FT37/A;1-Material Transformation Capability (View)	Material Transformati...	Feature					
FT03/A;1-Material Transport Capability (View)	Material Transpo	Feature	Optional			N/A	
FT31/A;1-Compression Bonding Capability	Compression Bonding...	Feature					
FT05/A;1-Reliability (View)	Reliability	Feature	Mandatory				
FT06/A;1-Availability (View)	Availability	Feature	Mandatory				
FT07/A;1-Maintainability (View)	Maintainability	Feature	Mandatory				
FT08/A;1-Operability (View)	Operability	Feature	Mandatory				
FT10/A;1-Utilities and Space Compatibility (View)	Utilities and Space Co...	Feature	Optional				
FT11/A;1-System Performance Management (View)	System Performance ...	Feature	Optional				
FT12/A;1-System Fault Management (View)	System Fault Manage...	Feature	Optional				
FT13/A;1-System configuration Management (View)	System configuration ...	Feature	Optional				
FT14/A;1-System Accounting Management (View)	System Accounting M...	Feature	Optional				
FT15/A;1-System Security Management (View)	System Security Man...	Feature	Optional				
FT16/A;1-Health and Safety (View)	Health and Safety	Feature	Optional				
FT17/A;1-Environmental Compliance (View)	Environmental Compli...	Feature	Optional				
FT18/A;1-Deliverability (View)	Deliverability	Feature	Mandatory				
FT19/A;1-Equipment Configuration (View)	Equipment Configurat...	Feature	Mandatory				
FT20/A;1-Equipement Configuration Path (View)	Equipement Configur...	Feature	Optional				
FT21/A;1-Automatic Inspection (View)	Automatic Inspection	Feature	Optional				
FT22/A;1-Manual Inspection (View)	Manual Inspection	Feature	Optional				

```
000376/A;1-Feature Framework(Production) (View)
  FT32/A;1-Oil Filter Product Features (View)
    FT33/A;1-Engine Lubricant Filtration Feature (View)
    FT35/A;3-Reliability(Production) (View)
  FT36/A;1-Production Capability (View)
  FT37/A;1-Material Transformation Capability (View)
  FT03/A;1-Material Transport Capability (View)
  FT31/A;1-Compression Bonding Capability
  FT05/A;1-Reliability (View)
  FT06/A;1-Availability (View)
```

Figure 3.47. Specialized feature framework

The Physical Systems were not created in the general production pattern due to depending on specific projects; it is difficult to summarize physical system in a general production pattern. Physical systems in this research can be defined obviously, which includes: end seal adhesive, accordion filtration component, and filter cap component. Based on understanding of each physical system, physical system attributes were also created. The CAD drawing created in NX or other CAD tools can be linked with physical system to accomplish integration by using Teamcenter environment.

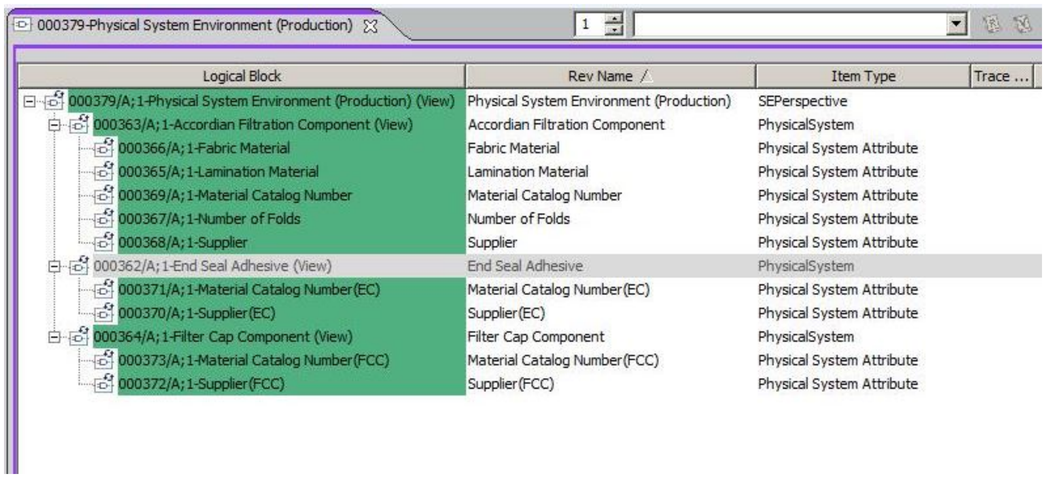


000376-Feature Framework(Production)

Logical Block	Rev Name	Item Type	Trace ...	Notes	Possible Value	Required Value	Capability Value	Units
000376/A;1-Feature Framework(Production) (Vi	Feature Framework(P...	SEPerspective						
FT32/A;1-Oil Filter Product Features (View)	Oil Filter Product Feat...	Feature						
FT33/A;1-Engine Lubricant Filtration Feat	Engine Lubricant Filtr...	Feature						
000328/A;1-Lubricant Type	Lubricant Type	Feature Attribute			0			
000329/A;1-Lubricant Flow Rate	Lubricant Flow Rate	Feature Attribute			High,Medium, Low			
000330/A;1-Lubricant Pressure Rang	Lubricant Pressure R...	Feature Attribute			High,Medium, Low			
000331/A;1-Filter Efficiency Class	Filter Efficiency Class	Feature Attribute						
000332/A;1-Service Application	Service Application	Feature Attribute			Consumer Automotiv...			
FT35/A;3-Reliability(Production) (View)	Reliability(Production)	Feature						
000333/A;1-Reliability (View)	Reliability	Feature Attribute			0			Percent
FT36/A;1-Production Capability (View)	Production Capab...	Feature						
000348/A;1-Production Capability (View)	Production Capability	Feature Attribute			Consumer Auto			
FI50/A;1-Perform Compression Bounding	Perform Compression...	Functional Inter...						
000361/A;1-Line Speed (View)	Line Speed	Feature Attribute						
FT37/A;1-Material Transformation Capability	Material Transformati...	Feature						
000349/A;1-Transformation Capability (V	Transformation Capa...	Feature Attribute			Compression Bounding			
FI51/A;1-Perform Compression Bounding	Perform Compression...	Functional Inter...						
FT03/A;1-Material Transport Capability (View	Material Transport Ca...	Feature						
FT31/A;1-Compression Bonding Capability	Compression Bonding...	Feature						
FT05/A;1-Reliability (View)	Reliability	Feature						
FT06/A;1-Availability (View)	Availability	Feature						
FT07/A;1-Maintainability (View)	Maintainability	Feature						
FT08/A;1-Operability (View)	Operability	Feature						
FT10/A;1-Utilities and Space Compatibility (Vi	Utilities and Space Co...	Feature						
FT11/A;1-System Performance Management	System Performance ...	Feature						
FT12/A;1-System Fault Management (View)	System Fault Manage...	Feature						
FT13/A;1-System configuration Management	System configuration ...	Feature						
FT14/A;1-System Accounting Management (\	System Accounting M...	Feature						
FT15/A;1-System Security Management (View	System Security Man...	Feature						
FT16/A;1-Health and Safety (View)	Health and Safety	Feature						
FT17/A;1-Environmental Compliance (View)	Environmental Compli...	Feature						
FT18/A;1-Deliverability (View)	Deliverability	Feature						
FT19/A;1-Equipment Configuration (View)	Equipment Configurat...	Feature						
FT20/A;1-Equipement Configuration Path (Vi	Equipement Configur...	Feature						
FT21/A;1-Automatic Inspection (View)	Automatic Inspection	Feature						
FT22/A;1-Manual Inspection (View)	Manual Inspection	Feature						

1
Feature Attribute
000376/A;1-Feature Framework(Production) (View)
FT32/A;1-Oil Filter Product Features (View)
FT33/A;1-Engine Lubricant Filtration Feature (View)
000328/A;1-Lubricant Type
000329/A;1-Lubricant Flow Rate
000330/A;1-Lubricant Pressure Range (View)
000331/A;1-Filter Efficiency Class
000332/A;1-Service Application
FT35/A;3-Reliability(Production) (View)
000333/A;1-Reliability (View)
FT36/A;1-Production Capability (View)
Feature
attribute
Functional Interaction (Interaction)
"A" Matrix Couplings
BB
(logical system)
Functional Role
WB
attribute
(physical system)
Design Component
attribute

Figure 3.48. Specialized feature attributes



000379-Physical System Environment (Production)

Logical Block	Rev Name	Item Type	Trace ...
000379/A;1-Physical System Environment (Production) (View)	Physical System Environment (Production)	SEPerspective	
000363/A;1-Accordian Filtration Component (View)	Accordian Filtration Component	PhysicalSystem	
000366/A;1-Fabric Material	Fabric Material	Physical System Attribute	
000365/A;1-Lamination Material	Lamination Material	Physical System Attribute	
000369/A;1-Material Catalog Number	Material Catalog Number	Physical System Attribute	
000367/A;1-Number of Folds	Number of Folds	Physical System Attribute	
000368/A;1-Supplier	Supplier	Physical System Attribute	
000362/A;1-End Seal Adhesive (View)	End Seal Adhesive	PhysicalSystem	
000371/A;1-Material Catalog Number(EC)	Material Catalog Number(EC)	Physical System Attribute	
000370/A;1-Supplier(EC)	Supplier(EC)	Physical System Attribute	
000364/A;1-Filter Cap Component (View)	Filter Cap Component	PhysicalSystem	
000373/A;1-Material Catalog Number(FCC)	Material Catalog Number(FCC)	Physical System Attribute	
000372/A;1-Supplier(FCC)	Supplier(FCC)	Physical System Attribute	

Figure 3.49. Specialized physical systems and physical systems attributes

Once the different views representing the S* Metamodel classes were created, the connections between these classes also needed to be constructed. As mentioned in the general production pattern, primary key is the unique identification connection between classes. In this specialized oil filter end seal compression model, the primary keys were still needed to help trace relationship from features to physical systems.

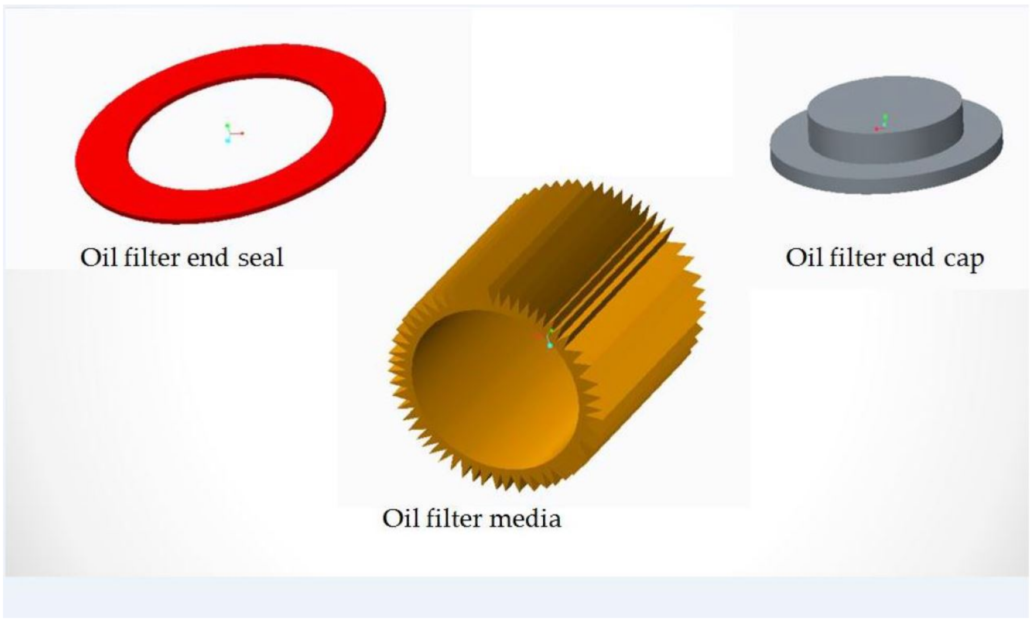
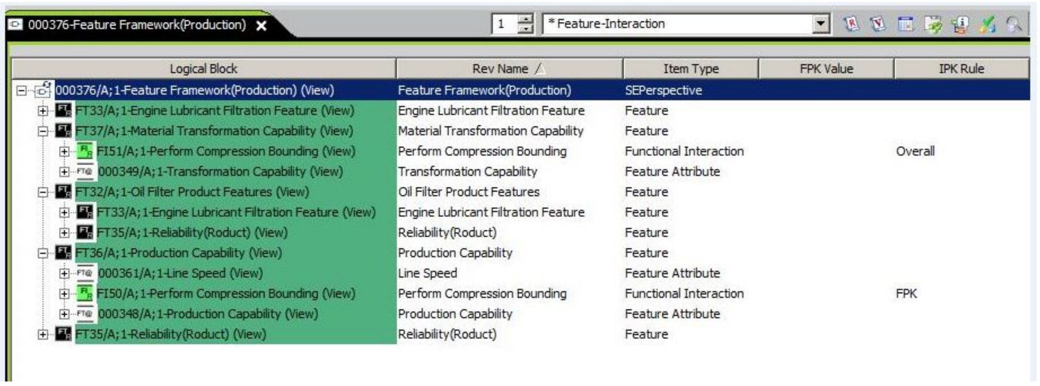


Figure 3.50. CAD drawings for physical systems

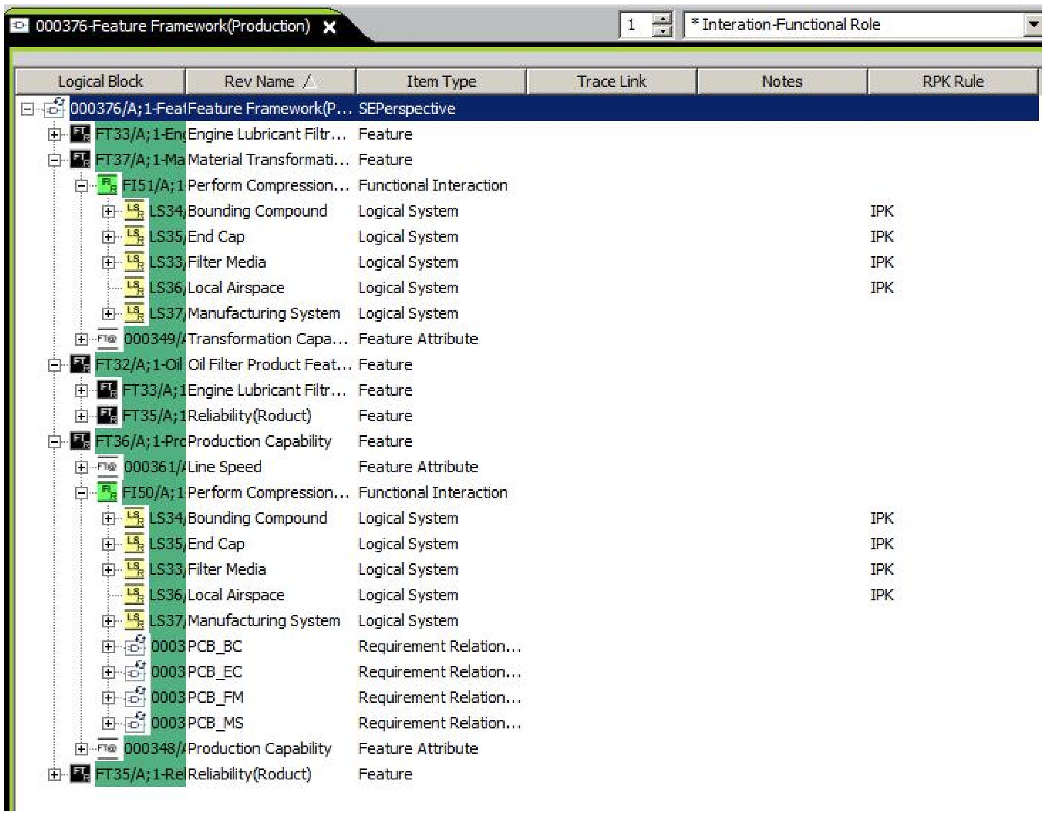
The figures below show the relationship constructed in BOM view of features and interaction, interactions and logical systems. Moreover, physical systems should be allocated with logical systems.



Logical Block	Rev Name	Item Type	FPK Value	IPK Rule
000376/A;1-Feature Framework(Production) (View)	Feature Framework(Production)	SEPerspective		
FT33/A;1-Engine Lubricant Filtration Feature (View)	Engine Lubricant Filtration Feature	Feature		
FT37/A;1-Material Transformation Capability (View)	Material Transformation Capability	Feature		
FI51/A;1-Perform Compression Bounding (View)	Perform Compression Bounding	Functional Interaction		Overall
000349/A;1-Transformation Capability (View)	Transformation Capability	Feature Attribute		
FT32/A;1-Oil Filter Product Features (View)	Oil Filter Product Features	Feature		
FT33/A;1-Engine Lubricant Filtration Feature (View)	Engine Lubricant Filtration Feature	Feature		
FT35/A;1-Reliability(Roduct) (View)	Reliability(Roduct)	Feature		
FT36/A;1-Production Capability (View)	Production Capability	Feature		
000361/A;1-Line Speed (View)	Line Speed	Feature Attribute		
FI50/A;1-Perform Compression Bounding (View)	Perform Compression Bounding	Functional Interaction		FPK
000348/A;1-Production Capability (View)	Production Capability	Feature Attribute		
FT35/A;1-Reliability(Roduct) (View)	Reliability(Roduct)	Feature		

Figure 3.51. Specialized connections between feature and interaction

In this specialization, a Matrix Coupling concept was introduced. Attribute Couplings represent how the value of these attributes with respect to each other if changes occurred. Matrix A Coupling describes how Feature Attribute change related to the



000376-Feature Framework(Production)

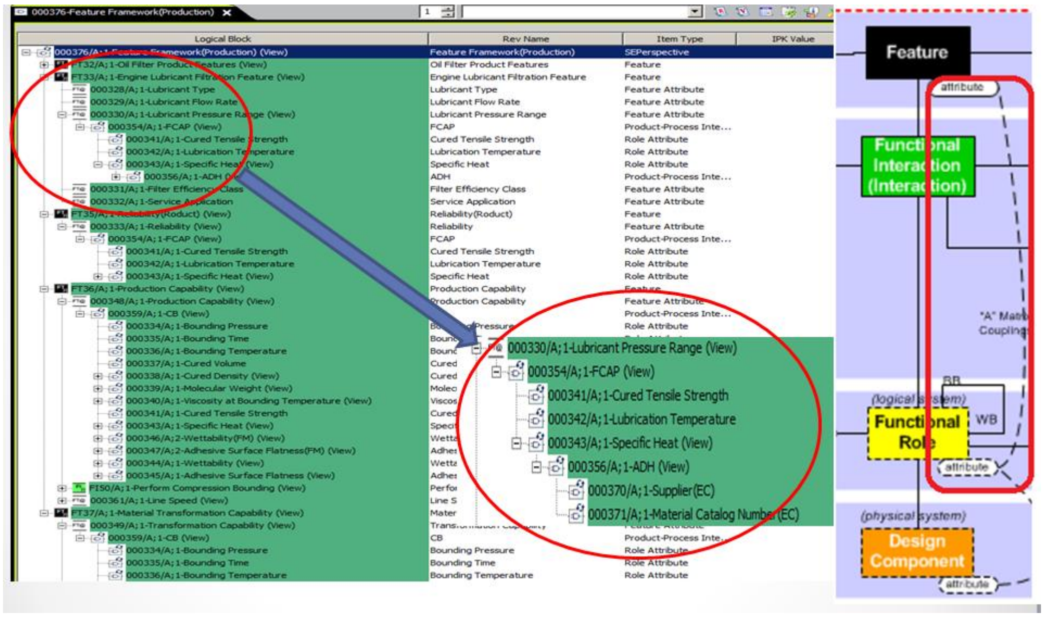
1 | * Interation-Functional Role

Logical Block	Rev Name	Item Type	Trace Link	Notes	RPK Rule
000376/A;1-Feat	Feature Framework(P...	SEPerspective			
FT33/A;1-En	Engine Lubricant Filtr...	Feature			
FT37/A;1-Ma	Material Transformati...	Feature			
FI51/A;1	Perform Compression...	Functional Interaction			
LS34	Bounding Compound	Logical System			IPK
LS35	End Cap	Logical System			IPK
LS33	Filter Media	Logical System			IPK
LS36	Local Airspace	Logical System			IPK
LS37	Manufacturing System	Logical System			
000349/	Transformation Capa...	Feature Attribute			
FT32/A;1-Oil	Oil Filter Product Feat...	Feature			
FT33/A;1	Engine Lubricant Filtr...	Feature			
FT35/A;1	Reliability(Roduct)	Feature			
FT36/A;1-Pro	Production Capability	Feature			
000361/	Line Speed	Feature Attribute			
FI50/A;1	Perform Compression...	Functional Interaction			
LS34	Bounding Compound	Logical System			IPK
LS35	End Cap	Logical System			IPK
LS33	Filter Media	Logical System			IPK
LS36	Local Airspace	Logical System			IPK
LS37	Manufacturing System	Logical System			
0003	PCB_BC	Requirement Relation...			
0003	PCB_EC	Requirement Relation...			
0003	PCB_FM	Requirement Relation...			
0003	PCB_MS	Requirement Relation...			
000348/	Production Capability	Feature Attribute			
FT35/A;1-Rel	Reliability(Roduct)	Feature			

Figure 3.52. Specialized connections between interaction and logical systems

value change of Functional Role Attribute. Matrix B Coupling describes how Functional Role Attributes change related to the value of Physical Component Attributes.

In the Matrix A Coupling, LSPD represents impacts in line speed, CB stands for Compression Bonding Interactions, FCAP represents impacts on Filter Capability and Reliability. In the BOM view structure, tree structures were still used to build relationships between attributes. For example, LSPD attached under a line speed feature attribute and then a bonding time system attribute created directly under LSPD. Once the matrix couplings were built, the change of one attributes could be traced with the other attribute quickly. This tree structure showed explicit up-down structure, but also made it easier for users to trace changes during process. In the Matrix A Coupling, if line speed is getting faster, the coupling called LSPD in the tree structure affect the bonding time in the manufacturing system.



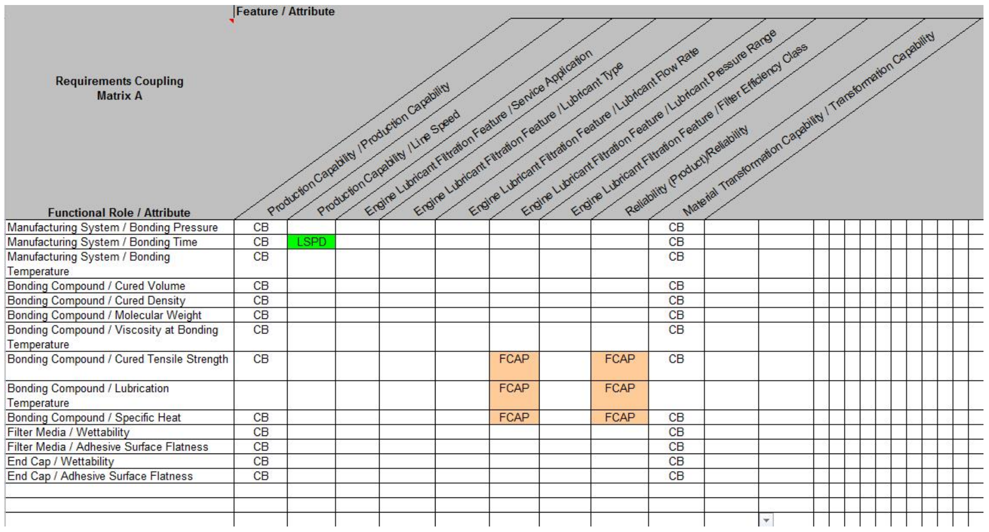
000376-Feature Framework(Production)

Logical Block	Rev Name	Item Type	IPK Value
000376/A;1-Feature Framework(Production) (View)	Feature Framework(Production)	SEPerspective	
FT32/A;1-Oil Filter Product Features (View)	Oil Filter Product Features	Feature	
FT33/A;1-Engine Lubricant Filtration Feature (View)	Engine Lubricant Filtration Feature	Feature	
000328/A;1-Lubricant Type	Lubricant Type	Feature Attribute	
000329/A;1-Lubricant Flow Rate	Lubricant Flow Rate	Feature Attribute	
000330/A;1-Lubricant Pressure Range (View)	Lubricant Pressure Range	Feature Attribute	
000354/A;1-FCAP (View)	FCAP	Product-Process Inte...	
000341/A;1-Cured Tensile Strength	Cured Tensile Strength	Role Attribute	
000342/A;1-Lubrication Temperature	Lubrication Temperature	Role Attribute	
000343/A;1-Specific Heat (View)	Specific Heat	Role Attribute	
000356/A;1-ADH	ADH	Product-Process Inte...	
000331/A;1-Filter Efficiency Class	Filter Efficiency Class	Feature Attribute	
000332/A;1-Service Application	Service Application	Feature Attribute	
FT35/A;1-Reliability(Roduct) (View)	Reliability(Roduct)	Feature	
000333/A;1-Reliability (View)	Reliability	Feature Attribute	
000354/A;1-FCAP (View)	FCAP	Product-Process Inte...	
000341/A;1-Cured Tensile Strength	Cured Tensile Strength	Role Attribute	
000342/A;1-Lubrication Temperature	Lubrication Temperature	Role Attribute	
000343/A;1-Specific Heat (View)	Specific Heat	Role Attribute	
FT36/A;1-Production Capability (View)	Production Capability	Feature	
000348/A;1-Production Capability (View)	Production Capability	Feature Attribute	
000359/A;1-CB (View)		Product-Process Inte...	
000334/A;1-Bounding Pressure	Bounding Pressure	Role Attribute	
000335/A;1-Bounding Time	Bounc		
000336/A;1-Bounding Temperature	Bounc		
000337/A;1-Cured Volume	Cured		
000338/A;1-Cured Density (View)	Cured		
000339/A;1-Molecular Weight (View)	Molec		
000340/A;1-Viscosity at Bounding Temperature (View)	Viscos		
000341/A;1-Cured Tensile Strength	Cured		
000343/A;1-Specific Heat (View)	Specif		
000346/A;2-Wettability(FM) (View)	Wetta		
000347/A;2-Adhesive Surface Flatness(FM) (View)	Adhes		
000344/A;1-Wettability (View)	Wetta		
000345/A;1-Adhesive Surface Flatness (View)	Adhes		
FI50/A;1-Perform Compression Bounding (View)	Perfo		
000361/A;1-Line Speed (View)	Line S		
FT37/A;1-Material Transformation Capability (View)	Mater		
000349/A;1-Transformation Capability (View)	Trans		
000359/A;1-CB (View)	CB	Product-Process Inte...	
000334/A;1-Bounding Pressure	Bounding Pressure	Role Attribute	
000335/A;1-Bounding Time	Bounding Time	Role Attribute	
000336/A;1-Bounding Temperature	Bounding Temperature	Role Attribute	

```
000330/A;1-Lubricant Pressure Range (View)
  000354/A;1-FCAP (View)
    000341/A;1-Cured Tensile Strength
    000342/A;1-Lubrication Temperature
    000343/A;1-Specific Heat (View)
      000356/A;1-ADH (View)
        000370/A;1-Supplier(EC)
        000371/A;1-Material Catalog Number(EC)
```

Figure 3.53. Matrix A coupling built in BOM structure

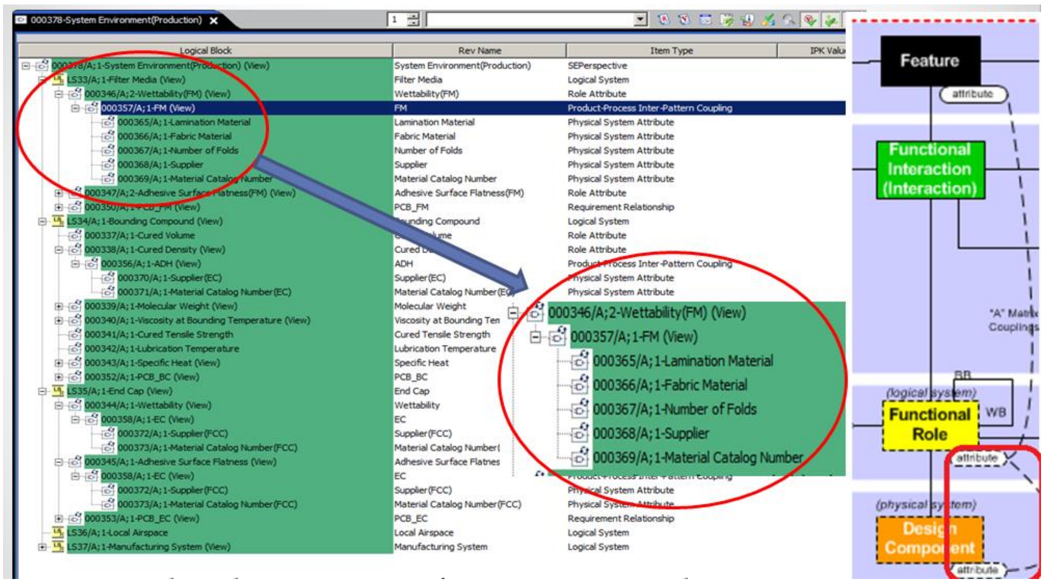
An excel relationship report for Matrix A Coupling is shown in the following figure.



Requirements Coupling Matrix A Functional Role / Attribute	Production Capability / Production Capability	Production Capability / Line Speed	Engine Lubricant Filtration Feature / Service Application	Engine Lubricant Filtration Feature / Lubricant Type	Engine Lubricant Filtration Feature / Lubricant Flow Rate	Engine Lubricant Filtration Feature / Lubricant Pressure Range	Engine Lubricant Filtration Feature / Filter Efficiency Class	Reliability (Product)/Reliability	Material Transformation Capability / Transformation Capability
Manufacturing System / Bonding Pressure	CB								CB
Manufacturing System / Bonding Time	CB	LSPD							CB
Manufacturing System / Bonding Temperature	CB								CB
Bonding Compound / Cured Volume	CB								CB
Bonding Compound / Cured Density	CB								CB
Bonding Compound / Molecular Weight	CB								CB
Bonding Compound / Viscosity at Bonding Temperature	CB								CB
Bonding Compound / Cured Tensile Strength	CB					FCAP		FCAP	CB
Bonding Compound / Lubrication Temperature						FCAP		FCAP	
Bonding Compound / Specific Heat	CB					FCAP		FCAP	CB
Filter Media / Wettability	CB								CB
Filter Media / Adhesive Surface Flatness	CB								CB
End Cap / Wettability	CB								CB
End Cap / Adhesive Surface Flatness	CB								CB

Figure 3.54. Matrix A coupling reference

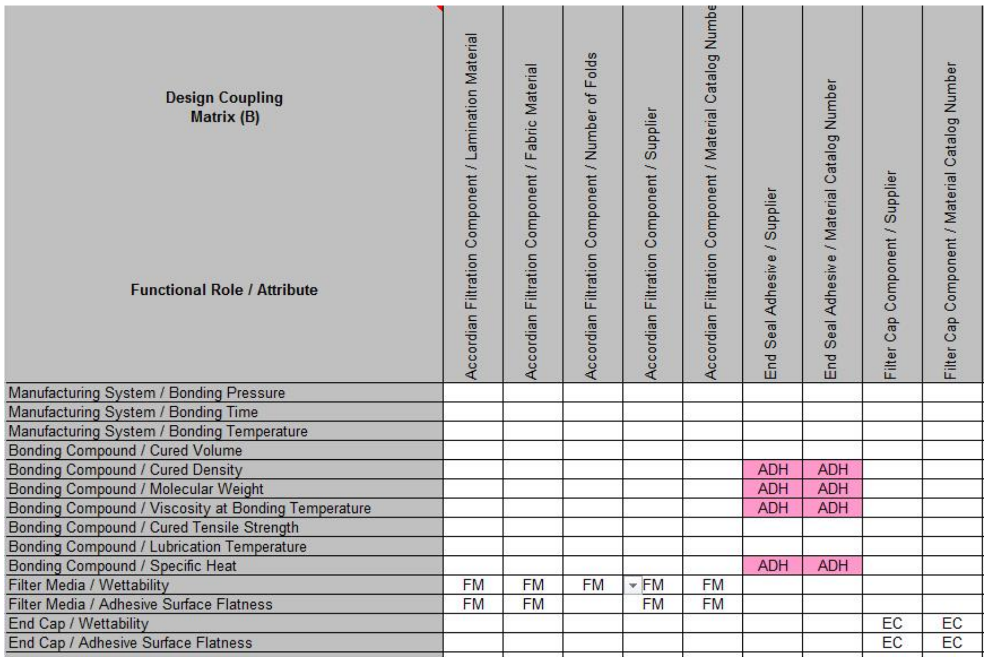
In the Matrix B Coupling, ADH stands for adhesive material data sheets, FM stands for filter media data sheets, and EC means end cap data sheets.



Logical Block	Rev Name	Item Type	IPK Valu
000378/A;1-System Environment(Production) (View)	System Environment(Production)	SEPerspective	
LS33/A;1-Filter Media (View)	Filter Media	Logical System	
000346/A;2-Wettability(FM) (View)	Wettability(FM)	Role Attribute	
000357/A;1-FM (View)	FM	Product-Process Inter-Pattern Coupling	
000365/A;1-Lamination Material	Lamination Material	Physical System Attribute	
000366/A;1-Fabric Material	Fabric Material	Physical System Attribute	
000367/A;1-Number of Folds	Number of Folds	Physical System Attribute	
000368/A;1-Supplier	Supplier	Physical System Attribute	
000369/A;1-Material Catalog Number	Material Catalog Number	Physical System Attribute	
000347/A;2-Adhesive Surface Flatness(FM) (View)	Adhesive Surface Flatness(FM)	Role Attribute	
000350/A;1-PCB_FM (View)	PCB_FM	Requirement Relationship	
LS34/A;1-Bounding Compound (View)	unding Compound	Logical System	
000337/A;1-Cured Volume	lume	Role Attribute	
000338/A;1-Cured Density (View)	Cured D	Role Attribute	
000356/A;1-ADH (View)	ADH	Product-Process Inter-Pattern Coupling	
000370/A;1-Supplier(EC)	Supplier(EC)	Physical System Attribute	
000371/A;1-Material Catalog Number(EC)	Material Catalog Number(EC)	Physical System Attribute	
000339/A;1-Molecular Weight (View)	Molecular Weight		
000340/A;1-Viscosity at Bounding Temperature (View)	Viscosity at Bounding Ten		
000341/A;1-Cured Tensile Strength	Cured Tensile Strength		
000342/A;1-Lubrication Temperature	Lubrication Temperature		
000343/A;1-Specific Heat (View)	Specific Heat		
000352/A;1-PCB_BC (View)	PCB_BC		
LS35/A;1-End Cap (View)	End Cap		
000344/A;1-Wettability (View)	Wettability		
000358/A;1-EC (View)	EC		
000372/A;1-Supplier(FCC)	Supplier(FCC)		
000373/A;1-Material Catalog Number(FCC)	Material Catalog Number(		
000345/A;1-Adhesive Surface Flatness (View)	Adhesive Surface Flatnes		
000358/A;1-EC (View)	EC	Product-Process Inter-Pattern Coupling	
000372/A;1-Supplier(FCC)	Supplier(FCC)	Physical System Attribute	
000373/A;1-Material Catalog Number(FCC)	Material Catalog Number(FCC)	Physical System Attribute	
000353/A;1-PCB_EC (View)	PCB_EC	Requirement Relationship	
LS36/A;1-Local Airspace	Local Airspace	Logical System	
LS37/A;1-Manufacturing System (View)	Manufacturing System	Logical System	

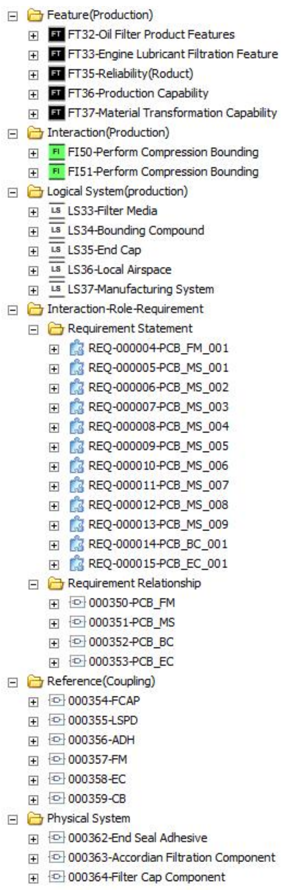
000346/A;2-Wettability(FM) (View)
000357/A;1-FM (View)
000365/A;1-Lamination Material
000366/A;1-Fabric Material
000367/A;1-Number of Folds
000368/A;1-Supplier
000369/A;1-Material Catalog Number

Figure 3.55. Matrix B coupling reference



Design Coupling Matrix (B) / Functional Role / Attribute	Accordian Filtration Component / Lamination Material	Accordian Filtration Component / Fabric Material	Accordian Filtration Component / Number of Folds	Accordian Filtration Component / Supplier	Accordian Filtration Component / Material Catalog Numbe	End Seal Adhesive / Supplier	End Seal Adhesive / Material Catalog Number	Filter Cap Component / Supplier	Filter Cap Component / Material Catalog Number
Manufacturing System / Bonding Pressure									
Manufacturing System / Bonding Time									
Manufacturing System / Bonding Temperature									
Bonding Compound / Cured Volume									
Bonding Compound / Cured Density						ADH	ADH		
Bonding Compound / Molecular Weight						ADH	ADH		
Bonding Compound / Viscosity at Bonding Temperature						ADH	ADH		
Bonding Compound / Cured Tensile Strength									
Bonding Compound / Lubrication Temperature									
Bonding Compound / Specific Heat						ADH	ADH		
Filter Media / Wettability	FM	FM	FM	FM	FM				
Filter Media / Adhesive Surface Flatness	FM	FM		FM	FM				
End Cap / Wettability								EC	EC
End Cap / Adhesive Surface Flatness								EC	EC

Figure 3.56. Matrix B coupling reference



- Feature(Production)
  - FT32-Oil Filter Product Features
  - FT33-Engine Lubricant Filtration Feature
  - FT35-Reliability(Roduct)
  - FT36-Production Capability
  - FT37-Material Transformation Capability
- Interaction(Production)
  - FI50-Perform Compression Bounding
  - FI51-Perform Compression Bounding
- Logical System(production)
  - LS33-Filter Media
  - LS34-Bounding Compound
  - LS35-End Cap
  - LS36-Local Airspace
  - LS37-Manufacturing System
- Interaction-Role-Requirement
  - Requirement Statement
    - REQ-000004-PCB_FM_001
    - REQ-000005-PCB_MS_001
    - REQ-000006-PCB_MS_002
    - REQ-000007-PCB_MS_003
    - REQ-000008-PCB_MS_004
    - REQ-000009-PCB_MS_005
    - REQ-000010-PCB_MS_006
    - REQ-000011-PCB_MS_007
    - REQ-000012-PCB_MS_008
    - REQ-000013-PCB_MS_009
    - REQ-000014-PCB_BC_001
    - REQ-000015-PCB_EC_001
  - Requirement Relationship
    - 000350-PCB_FM
    - 000351-PCB_MS
    - 000352-PCB_BC
    - 000353-PCB_EC
- Reference(Coupling)
  - 000354-FCAP
  - 000355-LSPD
  - 000356-ADH
  - 000357-FM
  - 000358-EC
  - 000359-CB
- Physical System
  - 000362-End Seal Adhesive
  - 000363-Accordian Filtration Component
  - 000364-Filter Cap Component

Figure 3.57. Created Specialized classes in Teamcenter systems engineering

The summary of created specialized classes or relationship objects were showed in Teamcenter systems engineering, this graph not only provides an explicit summary, but also shows a preparation approach to create oil filter end seal compression project from specialized general production pattern. By using this general production pattern based S* Metamodel in Teamcenter, this S* Pattern are formally configurable through configuration rules from selectable, configurable features for individual projects. This pattern is created once for an enterprise and can be updated from leaning occurs later. Since the S* pattern is built out of S* Metamodel components, a configured model is required two transformation operations: Populate (instantiate) and adjust values of attribute. Populated individual classes, relationships, and attributes into a specialized S* Model, on various occasions, more than one instance may be populated in a given element. This population is based on stakeholder needs and configuration rules, and then select feature and find correspondent information built in S* Pattern. Adjust values of attributes are based on specific project requirements for configuration a specialized model from general production pattern.

4. DISCUSSION OF RESULTS

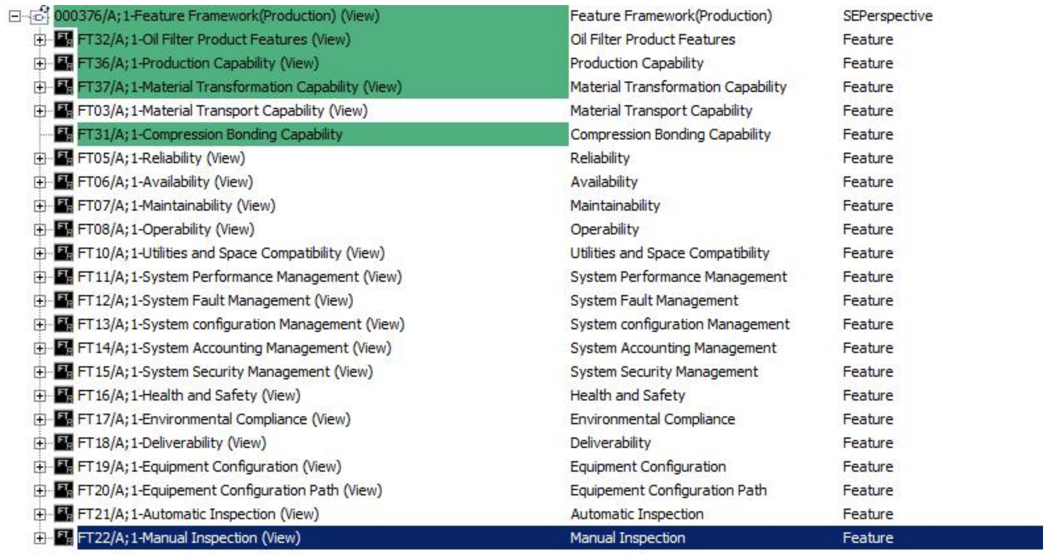
The integration of systems engineering and PLM is achieved in this thesis. The PLM tool was used to support PBSE methodology based on the S*Metamodel; in the meantime, the S*Pattern compatible with S*Metamodel provided a concrete MBSE model to enhance systems engineering application in PLM software. In collaboration with ICTT, we have demonstrated the methodology using a case study, namely, "the oil filter production end seal compression bonding". Below are the steps included in the development process of the oil filter after the implementation of the general production pattern:

- Collecting stakeholders' requirements

  In the oil filter end seal compression bonding production, the requirements may included: the production capability of end seal compression bonding, the transformation capability of end seal compression, engine lubricant filtration capability, filtration reliability etc.

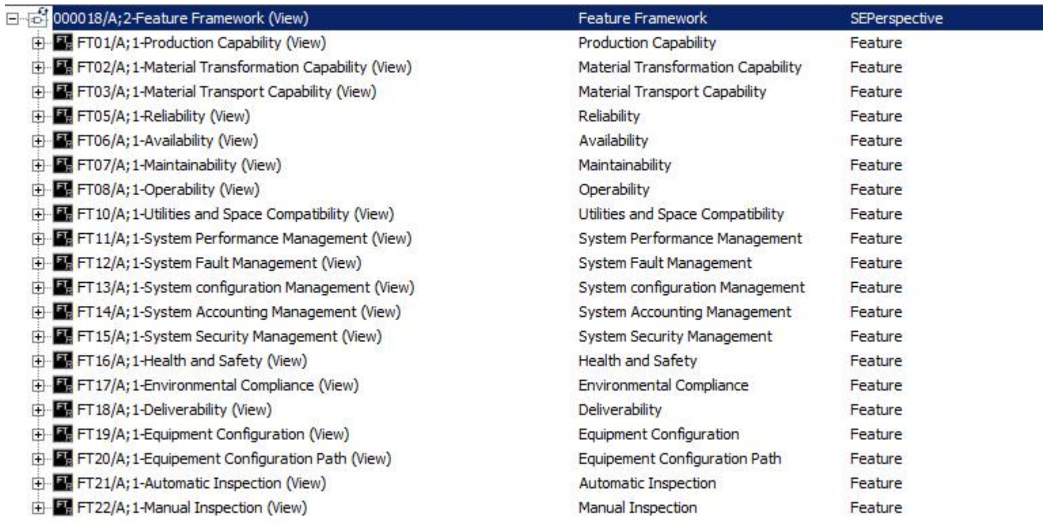
- Configuring the model by systems engineers

  Based on the requirements, the features and feature attributes were populated from the general production pattern. For example, the feature "Production Capability" in oil filter (Figure 4.1) was populated from "Production Capability" in general production pattern (Figure 4.2) based on the requirements of production capability of end seal compression bonding.



000376/A;1-Feature Framework(Production) (View)	Feature Framework(Production)	SEPerspective
FT32/A;1-Oil Filter Product Features (View)	Oil Filter Product Features	Feature
FT36/A;1-Production Capability (View)	Production Capability	Feature
FT37/A;1-Material Transformation Capability (View)	Material Transformation Capability	Feature
FT03/A;1-Material Transport Capability (View)	Material Transport Capability	Feature
FT31/A;1-Compression Bonding Capability	Compression Bonding Capability	Feature
FT05/A;1-Reliability (View)	Reliability	Feature
FT06/A;1-Availability (View)	Availability	Feature
FT07/A;1-Maintainability (View)	Maintainability	Feature
FT08/A;1-Operability (View)	Operability	Feature
FT10/A;1-Utilities and Space Compatibility (View)	Utilities and Space Compatibility	Feature
FT11/A;1-System Performance Management (View)	System Performance Management	Feature
FT12/A;1-System Fault Management (View)	System Fault Management	Feature
FT13/A;1-System configuration Management (View)	System configuration Management	Feature
FT14/A;1-System Accounting Management (View)	System Accounting Management	Feature
FT15/A;1-System Security Management (View)	System Security Management	Feature
FT16/A;1-Health and Safety (View)	Health and Safety	Feature
FT17/A;1-Environmental Compliance (View)	Environmental Compliance	Feature
FT18/A;1-Deliverability (View)	Deliverability	Feature
FT19/A;1-Equipment Configuration (View)	Equipment Configuration	Feature
FT20/A;1-Equipement Configuration Path (View)	Equipement Configuration Path	Feature
FT21/A;1-Automatic Inspection (View)	Automatic Inspection	Feature
FT22/A;1-Manual Inspection (View)	Manual Inspection	Feature

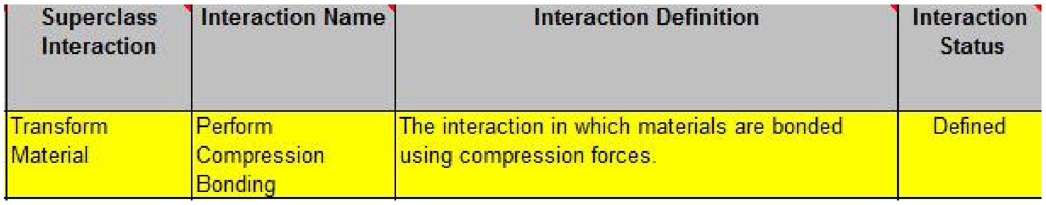
Figure 4.1. Specialized feature framework



000018/A;2-Feature Framework (View)	Feature Framework	SEPerspective
FT01/A;1-Production Capability (View)	Production Capability	Feature
FT02/A;1-Material Transformation Capability (View)	Material Transformation Capability	Feature
FT03/A;1-Material Transport Capability (View)	Material Transport Capability	Feature
FT05/A;1-Reliability (View)	Reliability	Feature
FT06/A;1-Availability (View)	Availability	Feature
FT07/A;1-Maintainability (View)	Maintainability	Feature
FT08/A;1-Operability (View)	Operability	Feature
FT10/A;1-Utilities and Space Compatibility (View)	Utilities and Space Compatibility	Feature
FT11/A;1-System Performance Management (View)	System Performance Management	Feature
FT12/A;1-System Fault Management (View)	System Fault Management	Feature
FT13/A;1-System configuration Management (View)	System configuration Management	Feature
FT14/A;1-System Accounting Management (View)	System Accounting Management	Feature
FT15/A;1-System Security Management (View)	System Security Management	Feature
FT16/A;1-Health and Safety (View)	Health and Safety	Feature
FT17/A;1-Environmental Compliance (View)	Environmental Compliance	Feature
FT18/A;1-Deliverability (View)	Deliverability	Feature
FT19/A;1-Equipment Configuration (View)	Equipment Configuration	Feature
FT20/A;1-Equipement Configuration Path (View)	Equipement Configuration Path	Feature
FT21/A;1-Automatic Inspection (View)	Automatic Inspection	Feature
FT22/A;1-Manual Inspection (View)	Manual Inspection	Feature

Figure 4.2. General production pattern feature framework

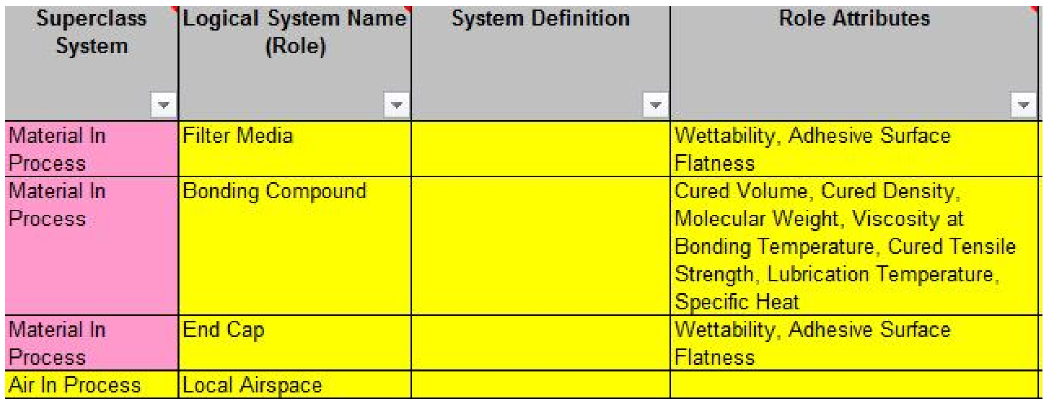
The functional interaction describes all external interactions of a subject system; “Perform Compression Bonding” was selected in the demonstration. As shown in Figure 4.3, this interaction was populated from the interaction “Transform Material” in the general production pattern.



Superclass Interaction	Interaction Name	Interaction Definition	Interaction Status
Transform Material	Perform Compression Bonding	The interaction in which materials are bonded using compression forces.	Defined

Figure 4.3. Specialized interaction in oil filter case

Based on the understanding of features and interactions, the logical systems were defined. “Filter Media”, “End Cap”, and “Bonding Compound” logical systems were created based on “Material in Process” from general production pattern. The “Local Airspace” logical system was created in the oil filter model based on “Air in Process” from the general production pattern, as shown in Figure 4.4.



Superclass System	Logical System Name (Role)	System Definition	Role Attributes
Material In Process	Filter Media		Wettability, Adhesive Surface Flatness
Material In Process	Bonding Compound		Cured Volume, Cured Density, Molecular Weight, Viscosity at Bonding Temperature, Cured Tensile Strength, Lubrication Temperature, Specific Heat
Material In Process	End Cap		Wettability, Adhesive Surface Flatness
Air In Process	Local Airspace		

Figure 4.4. Specialized logical systems in oil filter case

In Figure 4.5, physical systems “Filter Media”, “End Cap”, and “Bonding Compound” in the oil filter model were allocated with logical systems and CAD drawing.

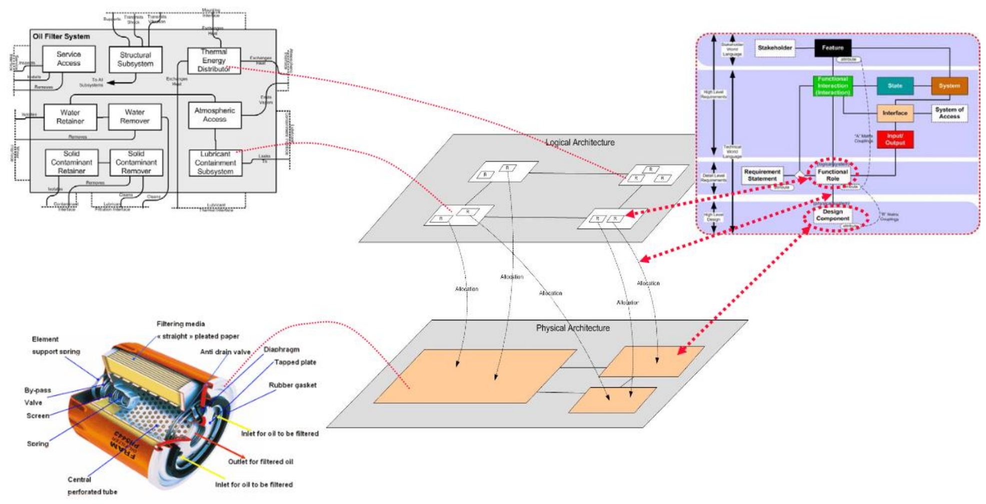
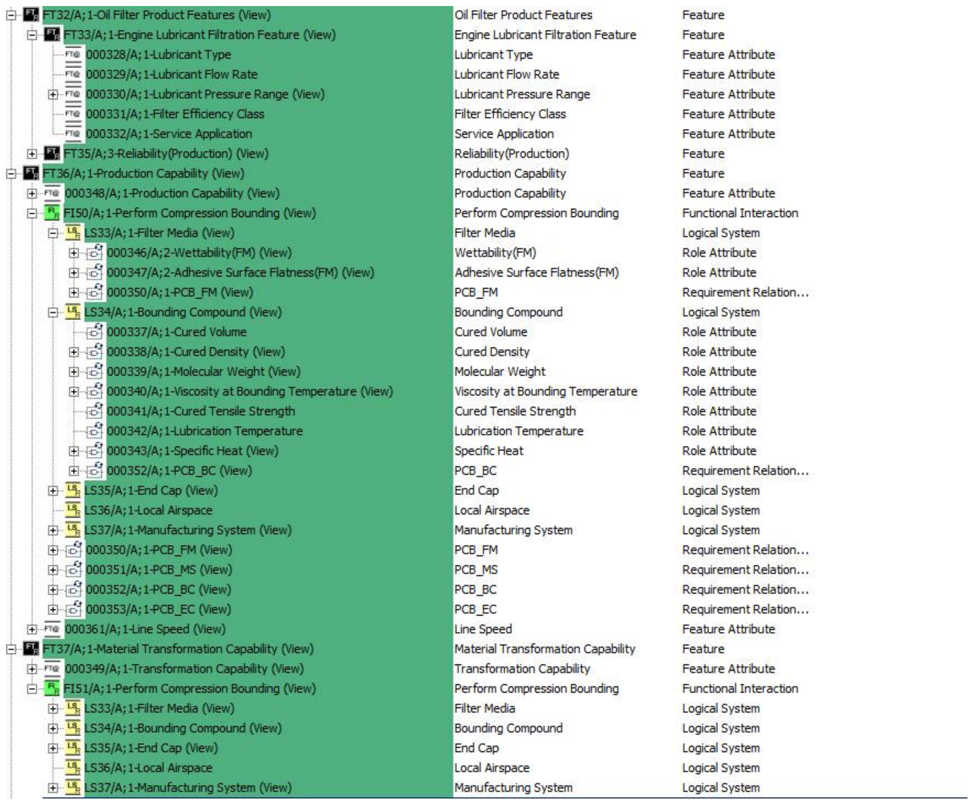


Figure 4.5. Allocation of logical systems and physical system in oil filter case [40]



FT32/A;1-Oil Filter Product Features (View)	Oil Filter Product Features	Feature
FT33/A;1-Engine Lubricant Filtration Feature (View)	Engine Lubricant Filtration Feature	Feature
000328/A;1-Lubricant Type	Lubricant Type	Feature Attribute
000329/A;1-Lubricant Flow Rate	Lubricant Flow Rate	Feature Attribute
000330/A;1-Lubricant Pressure Range (View)	Lubricant Pressure Range	Feature Attribute
000331/A;1-Filter Efficiency Class	Filter Efficiency Class	Feature Attribute
000332/A;1-Service Application	Service Application	Feature Attribute
FT35/A;3-Reliability(Production) (View)	Reliability(Production)	Feature
FT36/A;1-Production Capability (View)	Production Capability	Feature
000348/A;1-Production Capability (View)	Production Capability	Feature Attribute
FI50/A;1-Perform Compression Bounding (View)	Perform Compression Bounding	Functional Interaction
LS33/A;1-Filter Media (View)	Filter Media	Logical System
000346/A;2-Wettability(FM) (View)	Wettability(FM)	Role Attribute
000347/A;2-Adhesive Surface Flatness(FM) (View)	Adhesive Surface Flatness(FM)	Role Attribute
000350/A;1-PCB_FM (View)	PCB_FM	Requirement Relation...
LS34/A;1-Bounding Compound (View)	Bounding Compound	Logical System
000337/A;1-Cured Volume	Cured Volume	Role Attribute
000338/A;1-Cured Density (View)	Cured Density	Role Attribute
000339/A;1-Molecular Weight (View)	Molecular Weight	Role Attribute
000340/A;1-Viscosity at Bounding Temperature (View)	Viscosity at Bounding Temperature	Role Attribute
000341/A;1-Cured Tensile Strength	Cured Tensile Strength	Role Attribute
000342/A;1-Lubrication Temperature	Lubrication Temperature	Role Attribute
000343/A;1-Specific Heat (View)	Specific Heat	Role Attribute
000352/A;1-PCB_BC (View)	PCB_BC	Requirement Relation...
LS35/A;1-End Cap (View)	End Cap	Logical System
LS36/A;1-Local Airspace	Local Airspace	Logical System
LS37/A;1-Manufacturing System (View)	Manufacturing System	Logical System
000350/A;1-PCB_FM (View)	PCB_FM	Requirement Relation...
000351/A;1-PCB_MS (View)	PCB_MS	Requirement Relation...
000352/A;1-PCB_BC (View)	PCB_BC	Requirement Relation...
000353/A;1-PCB_EC (View)	PCB_EC	Requirement Relation...
000361/A;1-Line Speed (View)	Line Speed	Feature Attribute
FT37/A;1-Material Transformation Capability (View)	Material Transformation Capability	Feature
000349/A;1-Transformation Capability (View)	Transformation Capability	Feature Attribute
FI51/A;1-Perform Compression Bounding (View)	Perform Compression Bounding	Functional Interaction
LS33/A;1-Filter Media (View)	Filter Media	Logical System
LS34/A;1-Bounding Compound (View)	Bounding Compound	Logical System
LS35/A;1-End Cap (View)	End Cap	Logical System
LS36/A;1-Local Airspace	Local Airspace	Logical System
LS37/A;1-Manufacturing System (View)	Manufacturing System	Logical System

Figure 4.6. Oil filter end seal bonding BOM view

- Figure 4.6 shows a created BOM view in Teamcenter based on the defined features, interactions, and logical systems. Based on the model, related stakeholders were involved:

  - Project Manager and Systems Engineer

    Systems engineer and project manager were defined by default and were involved in the entire project.

  - Design engineers

    Design engineers were defined as a result of the presence of physical systems "End Cap" CAD drawing.

  - Manufacturing engineers

    Manufacturing engineers were defined as a result of the presence of "Manufacturing System" in physical system. The manufacturing system was managed by manufacturing engineers.

  - Quality engineers

    Quality engineers were defined as a result of the presence of several physical systems, the engine lubricant material, end cap, filter media, and bonding material were evaluated by quality engineers.

  - Material engineers

    Material engineers were defined as a result of the presence of physical system "Filter Media" and "Bonding Compound"; the filter media, lubricant, and bonding material were selected by material engineers.

  - Customers and Supplier

    Customer and suppliers were defined by default.

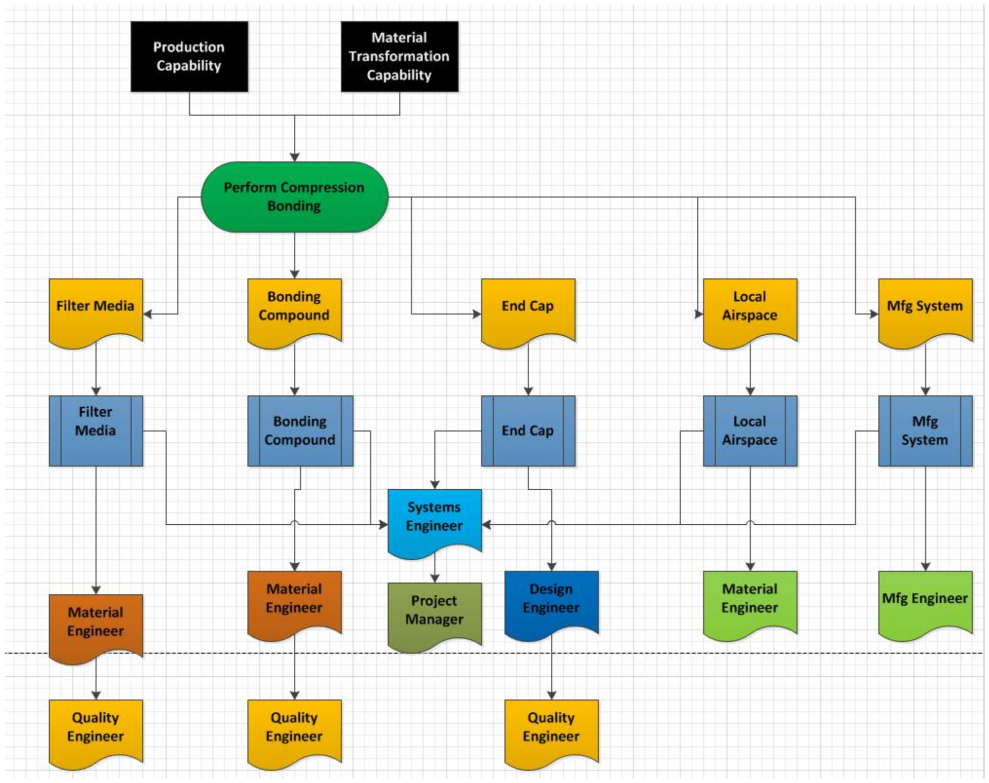


Figure 4.7. Microsoft Visio view of oil filter model

- If a change of requirement happened (e.g. the production capability, in Figure 4.7, changed)

  Based on the connection of production capability feature with interaction, logical systems, and physical systems, design engineers, material engineers, manufacturing engineers, quality engineers, project manager, and systems engineer are informed about the change.

  Systems engineers and project manager should known first, and then depending on which logical systems were involved in the interaction, the other engineers take certain actions.

  Design engineers may change the CAD drawing, material engineers may re-select the materials, quality engineers may re-evaluate materials, and project managers may change schedule management, risk management, or cost management etc..

Compared to the traditional process, the above process has the following advantages:

- Users' requirements and the changes could be captured and reflected in the product design promptly and accurately;
- Systems engineers, especially junior level, are able to create models quickly and prevent mistakes during modeling. Systems engineers can do their job much easier because of the reusable and reconfigurable nature of the models;
- Systems engineers have a trustful single data source to integrate all the systems engineering tools;
- The live integration of PLM system with Microsoft package provides a powerful environment to link Word, Excel, and Visio to generate documents, reports, and diagrams used for systems engineers;
- The new process could increase the re-use of previous designs because similar products' models were configured from one generic model. This will help accelerate the process of part verification and validation, and reduce the time to market and potential failure;
- Systems engineers as administrators in the PLM system are able to set different accesses for different roles and the corresponding actions that can be taken during change.

S*Patterns and S*Models are tool-independent, which always conforms to the underlying S*Metamodel. In order to use S*Metamodel in Teamcenter, mapping was one of the most important steps. Fundamental detailed specifications of the S*Metamodel classes, relationships, and attributes should be fully reflected in a specific schema. The mapping in this research used the extension of Teamcenter base item type which is a logical block to represent classes and attributes. This mapping process provided the General Production Pattern and oil filter specialization model with fundamental capabilities.

The relationship between classes, for example, Feature and Interaction, Interaction and Functional Role, Functional Role and Physical System, and Coupling Matrices were mapped to standard Teamcenter structural relationships which are tree structures. These tree structures had many benefits from representing those relationships. For example, tree structures in BOM management view showed explicit up-down structures; it also made it easier for users to trace changes during the process.

5. CONCLUSION AND FUTURE WORK

The successful integration of PLM and S*Pattern is significant for enterprises, institutions, and especially systems engineers in addressing challenges easier and quicker. In this work, a specialized model was generated using General Production Pattern and implemented in Teamcenter. Using S*Pattern's extensions and re-using legacy data to quickly created a systems engineer model reduce time and cost. In this research, benefits brought to systems engineering practice include: stakeholders' requirements and changes could be captured and reflected in product design process promptly and accurately; systems engineers were able to create models quickly and prevent mistakes during modeling; systems engineers could do their job much easier because of reusable and reconfigurable nature of the models; and the process could increase the re-use of previous designs.

In this work, the key portion of General Production Pattern expressed by S*Metamodel (Feature-Interaction-Functional Role-Design Component) was implemented in Siemens Teamcenter successfully. The Mapping in BMIDE generates the specific S*Metamodel classes, attributes, and relationships into Teamcenter Schema.

Future work should include the completion of the implementation of S*Metamodel in Teamcenter schema. This research work provided a basis for future evaluation of the key portion of the S*Metamodel in general production pattern, as a successful evaluation requires a good understanding of the mapping and implementation of the S*Pattern.

In addition, there is a number of Teamcenter systems engineering abilities which were not manifested in this research, for example: create, maintain and perform physical model structures in Structure Manager, manage changes, integrate specific CAD tools, and manage the impact analysis and process of changes etc. In this case, the mapping process should be updated and upgraded to satisfy all the potential capabilities when using of S* Pattern in Teamcenter environment.

Even though these tree structures showed many benefits when representing those relationships, if the complexity of implementation, specialization and configuration increases, a better relationship mapping method is required to define different relationships separately. Teamcenter also has its own relationship management item type such as “ImanRelation” to develop a variety of relationship types in BMIDE for better management in Teamcenter.

Due to the limited time, this research work did not demonstrate all the Teamcenter capabilities which can be used in this S* Pattern. Applying the S* Pattern with Teamcenter to better support enterprise projects, specializing and configuring S* Pattern in S* Models, and taking full advantage of PLM tool’s capabilities, improve the product performance and competitiveness.

REFERENCES

REFERENCES

[1] P. B. Hart, “A plm implementation for aerospace systems engineering-conceptual rotorcraft design,” vol. M.S. thesis, Dept. Mechanical. Eng., Georgia Institute of Technology., Georgia, May 2009.

[2] A. Saaksvuori and A. Immonen, *Product lifecycle management.* Springer Science and Business Media, 2008.

[3] G. Guo, C. Yu, and F. Liu, “Connotation and technology architecture of product lifecycle management(plm),” *China Mechanical Engineering*, vol. 15, no. 6, pp. 512–515, 2004.

[4] N. I. Standards and T. Manufacturing, *Product’s lifecycle.* NIST Programs of the Manufacturing Engineering Laboratory, 2008.

[5] R. Sudarsan, S. J. Fenves, R. D. Sriram, and F. Wang, “A product information modeling framework for product lifecycle management,” *Computer-aided design*, vol. 37, no. 13, pp. 1399–1411, 2005.

[6] A. Felic, B. Knig-Ries, and M. Klein, “Process-oriented semantic knowledge management in product lifecycle management,” *Procedia CIRP*, vol. 25, pp. 361–368, 2014.

[7] S. Lee, Y.-S. Ma, G. Thimm, and J. Verstraeten, “Product lifecycle management in aviation maintenance, repair and overhaul,” *Computers in industry*, vol. 59, no. 2, pp. 296–303, 2008.

[8] S. Terzi, A. Bouras, D. Dutta, M. Garetti, and D. Kiritsis, “Product lifecycle management-from its history to its new role,” *International Journal of Product Lifecycle Management*, vol. 4, no. 4, pp. 360–389, 2010.

[9] “Why do systems engineering?” *INCOSEUK*, 2009.

[10] A. J. Shenhar and B. Sauser, “Systems engineering management: The multidisciplinary discipline,” *Handbook of systems engineering and management*, pp. 113–36, 2009.

[11] C. Haskins, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities.* INCOSE, 2007.

[12] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis.* Prentice Hall Englewood Cliffs, New Jersey, 2006, vol. 4.

[13] P. Pearce and M. Hause, “Iso-15288, oosem and model-based submarine design.” SETE/APCOSE 2012, 2012.

[14] S. Arnold, “Iso 15288 systems engineeringsystem life cycle processes,” *International Standards Organisation*, 2002.

[15] M. Messaadia and A. Sahraoui, "Plm as linkage process in a systems engineering framework," *International Journal of Product Development*, vol. 4, no. 3, pp. 382–395, 2007.

[16] E. Vezzetti, M. G. Violante, and F. Marcolin, "A benchmarking framework for product lifecycle management (plm) maturity models," *The International Journal of Advanced Manufacturing Technology*, vol. 71, no. 5-8, pp. 899–918, 2014.

[17] T. Milner, M. Volas, and A. Sanders, "Systems engineering methodology for linking requirements to design complexity and manufacturing trade space constraints," *Procedia Computer Science*, vol. 16, pp. 947–956, 2013.

[18] S. B. Gershwin, *Manufacturing Systems Engineering.* Prentice Hall, 1994.

[19] W. J. Hopp and M. L. Spearman, *Factory physics.* Waveland Press, 2011.

[20] J. K. Liker, "The toyota way: 14 management principles from the world's greatest manufacturer," *McGraw Hill*, 2004.

[21] A. MacCalman, H. Kwak, M. McDonald, and S. Upton, "Capturing experimental design insights in support of the model-based system engineering approach," *Procedia Computer Science*, vol. 44, pp. 315–324, 2015.

[22] A. Fay, B. Vogel-Heuser, T. Frank, K. Eckert, T. Hadlich, and C. Diedrich, "Enhancing a model-based engineering approach for distributed manufacturing automation systems with characteristics and design patterns," *Journal of Systems and Software*, vol. 101, pp. 221–235, 2015.

[23] W. Schindel, "Pattern-based systems engineering (pbse), based on s*mbse models," 2015.

[24] W. D. Schindel, *What is the Smallest Model of a System?* Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering.

[25] W. Schindel, "System interactions: Making the heart of systems more visible," *Proc. of INCOSE Great Lakes Regional Conference*, 2013.

[26] W. Schindel, S. Sanyal, J. Sherey, and S. Lewis, "Accelerating mbse impacts across the enterprise: Model-based s* pattern," in *Proc. of INCOSE International Symposium IS2015, Seattle.*

[27] W. D. Schindel, "4.3.2 systems of innovation ii: The emergence of purpose," in *INCOSE International Symposium*, vol. 23. Wiley Online Library, pp. 1006–1020.

[28] G. Fanjiang, J. H. Grossman, W. D. Compton, and P. P. Reid, *Building a Better Delivery System: A New Engineering/Health Care Partnership.* National Academies Press, 2005.

[29] J. A. Estefan, "Survey of model-based systems engineering (mbse) methodologies," *Incose MBSE Focus Group*, vol. 25, p. 8, 2007.

[30] M. Bajaj, D. Zwemer, R. Peak, A. Phung, A. G. Scott, and M. Wilson, "Slim: collaborative model-based systems engineering workspace for next-generation complex systems," *INCOSE IW 2013 - MBSE Workshop*, pp. 1–20, 2011.

[31] "Systematica metamodel: Metamodel version 7.1, methodology release 4.0," ICTT System Science, Tech. Rep., May 29 2009.

[32] I. T. Operations, "Incose-tp-2004-004-02: Systems engineering vision 2020, version 2.03," *International Council on Systems Engineering, Seattle, WA*, September 2007.

[33] S. Friedenthal, A. Moore, and R. Steiner, *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.

[34] W. Schindel, "Pattern-based systems engineering: An extension of model-based se," *INCOSE IS2005 Tutorial TIES*, vol. 4, 2005.

[35] V. Gecevska, T. Stojanova, and B. Jovanovski, "Product lifecycle management tools," *Annals of Faculty Engineering Hunedoara*, vol. 1, pp. 219–222, 2013.

[36] "Learning advantage," (Last accessed July 2015). [Online]. Available: https://training.plm.automation.siemens.com/mytraining/home.cfm?cmd=&details=

[37] "Using teamcenter release 9 with systematica methodology release 4.0: Schema configuration guide," ICTT System Sciences, Tech. Rep., February. 4 2013.

[38] *Teamcenter 10.1 Business Modeler IDE Guide*, Siemens Product Lifecycle Management Software Inc., 2013.

[39] "Generic manufacturing pattern v1.1.13jjs.xls," ICTT System Sciences, Tech. Rep.

[40] "Oil filter production system v1.1.1.xls," ICTT System Sciences, Tech. Rep.