



HAL
open science

Implementation of Systems Engineering Model into Product Lifecycle Management Platform

Shuning Li, Hazim El-Mounayri, Weijie Zhang, Bill Schindel, Jason Sherey

► **To cite this version:**

Shuning Li, Hazim El-Mounayri, Weijie Zhang, Bill Schindel, Jason Sherey. Implementation of Systems Engineering Model into Product Lifecycle Management Platform. 12th IFIP International Conference on Product Lifecycle Management (PLM), Oct 2015, Doha, Qatar. pp.601-608, 10.1007/978-3-319-33111-9_54 . hal-01377487

HAL Id: hal-01377487

<https://inria.hal.science/hal-01377487>

Submitted on 7 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Implementation of Systems Engineering Model into Product Lifecycle Management Platform

Shuning Li¹, Hazim El-Mounayri¹, Weijie Zhang¹, Bill Schindel², Jason Sherey²

¹Department of Mechanical Engineering, Indiana University Purdue University Indianapolis (IUPUI), Indianapolis, Indiana, USA
li33@iupui.edu

²ICTT Systems Science, Terra Haute, Indiana, U.S.A
schindel@ictt.com

Abstract. Manufacturing companies are 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 provide solutions for these challenges. The two concepts not only share many common characteristics, but also complement each other. Even though systems engineering and PLM have become closely related in the past few years, implementation of systems engineering models into a PLM platform has rarely been conducted. In this study, the key portion of a model-based system engineering model was implemented into a PLM platform, and the implementation was validated. The results shows that the current implementation can help capture and reflect stakeholders' requirements and changes in product design process promptly and accurately; and reduce the time and potential mistakes in creating a new systems engineering model.

Keywords: Product Life Cycle Management (PLM), Systems Engineering, Model-Based Systems Engineering, General Manufacturing Model, Systematica Metamodel

1 Introduction

In today's world, companies are facing the challenges of increasing product complexity while at the same time reducing cost and time in a highly competitive global market. A direct consequence of the increasing product complexity is a more complex manufacturing system. Product Lifecycle Management (PLM) and Systems Engineering have the potential to help companies manage the more and more complex manufacturing system, and avoid costly product development and launching, as well as reducing product failure. These two concepts not only share many common characteristics [1], but also complement each other. On one hand, 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 [1]. On the other hand, PLM can provide a reliable data source to system engineers. This is especially helpful for those fast changing data like requirements. PLM also connected all the information within the processes together, which provides system engineers a tool to trace changes and preform analysis on the affections of the changes. Researchers started trying to integrate PLM and systems engineering to overcome the weaknesses of traditional systems engineering tools and achieve the benefits of using PLM platform [1] [2]. But the current integrations either only used PLM as a data source or implemented a traditional systems engineering model. Using PLM only as a data source cannot get the full benefits of a PLM platform, and systems engineers and their works were still isolated from the rest of lifecycle processes and information. Traditional systems engineering models were document centric models which were not easy to implement into a PLM system.

The objectives of this study were to implement the key portion of a model-based system engineering model into a PLM platform, and validate the implementation by generating a product specified model from the general model.

2 Methodology

The systems engineering model used in this study is a general manufacturing model. This model was built by ICTT System Sciences using Systematica Metamodel (S* Metamodel) methodology [3]. The **Fig.1** depict key element of the S* Metamodel. In this study, the trace of Feature, Functional Interaction, Functional Role, and Design Component was being focused. Features were packages of behavior or performance of a system that have stakeholder value [3]. S* Models were aimed at covering all the stakeholders not only just users or customers. Feature attributes were features' parameters that expressed stakeholder valuations in stakeholder language [3]. For example, the Cruise Control Feature for a car had feature attributes in fuel economy and speed variation. Because Features and Feature Attributes covered all stakeholders' value and interest, they affected all the design decisions, trade-offs, and optimization should be made in accordance. In a system, there always existed interactions between components; an interaction means exchange of energy, force, mass or information [4] which lead to change of state. The Systematica called that interaction a functional interaction, and each component played a functional role in that interaction [4]. The functional role was also called logical system in S* Metamodel. Functional Roles were described by their behavior, and the role attributes were parameters of functional role which had technical valuations [5].

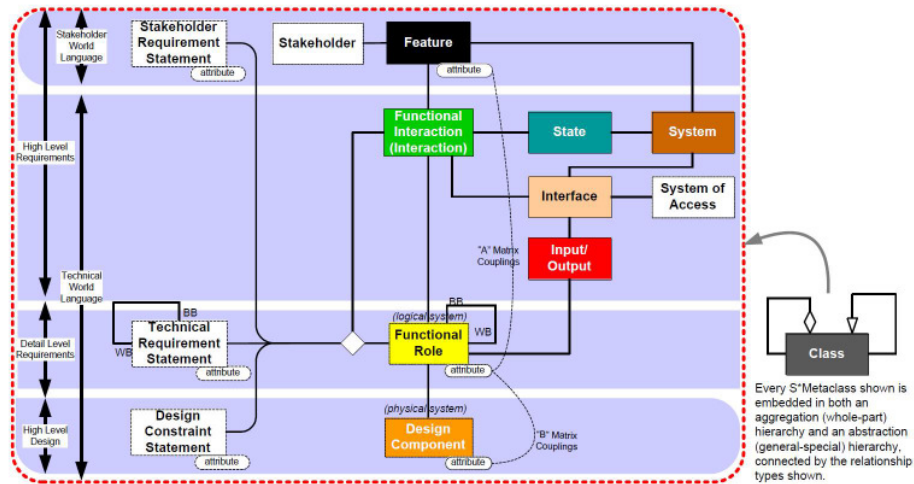


Fig. 1. S* Metamodel [3]

The PLM platform used in this study is Teamcenter® 10.1. Teamcenter is the most widely used PLM software system in the world [6]. It helped companies deliver complex products to the market by connecting people with products and process in order to enhance productivity and integrate global operations. Teamcenter® was the first PLM solution to integrate systems engineering within an entire product lifecycle. It provided a close loop systems engineering environment. The systems engineering environment employed systems engineering methodology to allow an engineer to establish systems requirement, then defined and validated all components and subsystems in the context of the entire system's lifecycle. The benefit was that products meted customers' value satisfaction and understood the entire impact of design decisions in the early stages of the lifecycle.

The research approach in this study consisted of three steps: mapping; implementation; and specialization. The first step was to map the S* Metamodel elements into Teamcenter® classes including relationships. The second step was to build the model into Teamcenter®. The last step was to generate oil filter end seal compression manufacturing model by specializing the general model.

2.1 Mapping

The mapping step transferred systems engineering model elements into PLM classes. The S* model elements, like feature, interface, functional interaction, were mapped Teamcenter® classes. All these new classes were children to Logical Block class. The relationship between two S* model elements were mapped to either structural relationships or trace link relationships. Trace link relationship was used to represent relationships in hierarchies. For example, features in systems engineering model were created based on stakeholders needs. So the relationship between a need

(requirement) and a feature was a trace link type of relationship (A "Need" object is the source of a "Feature" object). Trace links provide traceability between structure elements, and traceability defined as one object was precedent than another object. Most of the relationships were structural relationships which can be presented in a tree view, like the bill of material (BOM) view of a product structure. All of these classes were built by using Teamcenter® Business Modeler Integrated Development Environment (BMIDE)

2.2 Implementation

The implementation step built the general systems engineering model into the PLM platform. A set of features, interactions, and other S* metamodel elements in the general manufacturing model was created in Teamcenter®. The relationships between feature and feature attribute, feature and interaction, interaction and logical system were created, and presented in a tree view. One trace link relationship between a super class and its children was also created.

2.3 Specialization

The oil filter end seal production process was selected to demonstrate the specialization step in this study. Systems engineers usually started with identifying interactions when they built the model. The interaction in this specific model should be specialized from "Transform Material" functional interaction in the general model. A new interaction called "Perform Compression Bonding" was created for the specialized model based on the "Transform Material" functional interaction in the general model.

Once the interaction was decided, the logical systems were modeled. Filter media, bonding compound, and end cap are necessary parts for production, so local airspace, manufacturing system and other manufacturing logical systems were inherited from general production pattern directly. Functional role attributes were created and attached to logical systems, those role attributes were considered as technical valuation of production logical systems.

Based on stakeholders needs, production and manufacturing features were created. Material transformation capability, production capability features and other general production features for manufacturing were able to be populated from the general model directly. And a specialized oil filter production feature that contains engine lubricant filtration feature and reliability (production) features were created, and added to the model. Feature attribute based on stakeholders valuations under each feature were also created.

The Physical Systems were not created in the general manufacturing model because it is difficult to summarize physical system into a general model. But physical systems were necessary for specialized model. In this oil filter end seal case, the physical system included: end seal adhesive, accordion filtration component, and filter cap component. Physical system attributes were created based on the understanding of each physical part and the related physical laws. The CAD models

of these physical parts can be linked to physical system elements in the model as references and for change effects analysis.

3 Results

3.1 General Manufacturing Model

The general manufacturing model included groups of feature (feature framework), interactions (interaction framework), logical systems (system environment), and other S* model elements. Fig. 2 showed an example of the functional interaction framework in tree view. The structural relationships were also created, and presented in a tree view. Fig. 3 was an example of the relationships between features and interactions. Microsoft® Office tools were commonly used by systems engineers, and the reports or diagrams generated in Office tools were still the major deliveries for systems engineers. These reports and diagrams can be generated automatically after proper configuration in Teamcenter®. Fig. 4 gave an example of an automatically generated diagram based on the implemented in Visio.

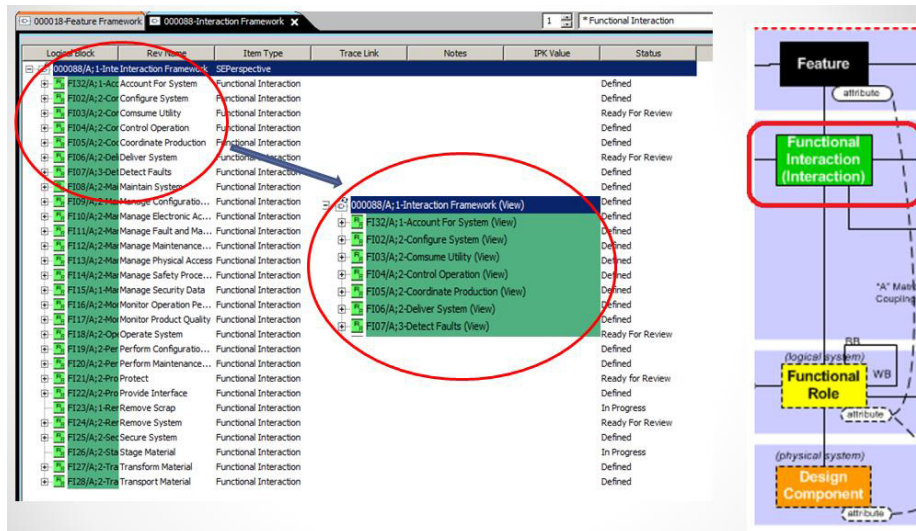


Fig. 2. Functional Interaction Framework in Tree View

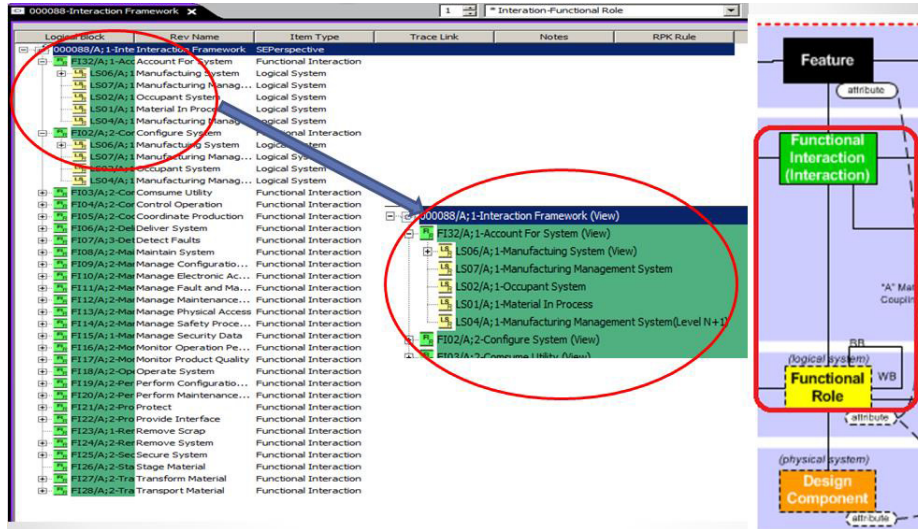


Fig. 3. Relationships between Features and Interactions in Tree View

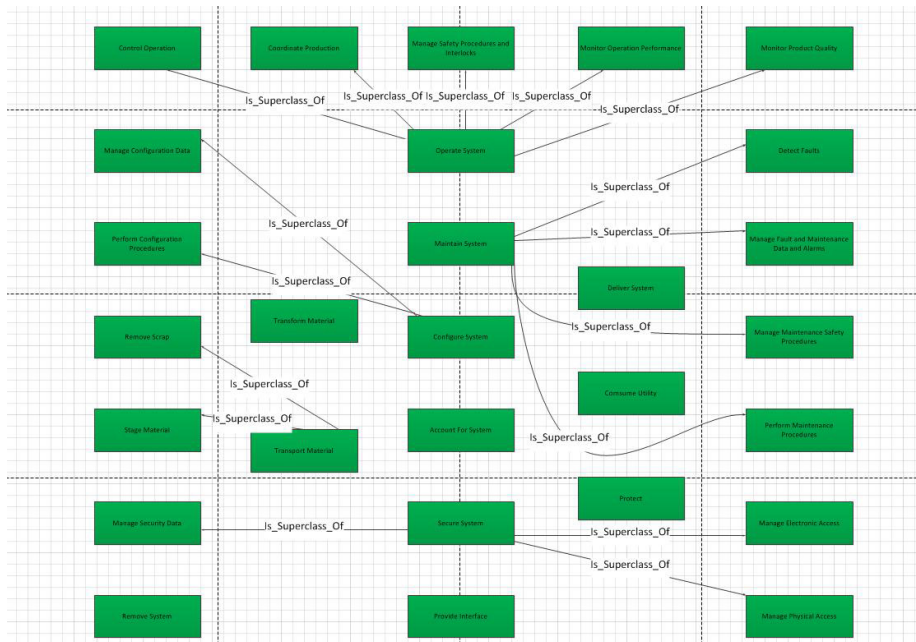


Fig. 4. Interaction Overview Diagram in Visio

3.2 Oil Filter End Seal Bonding Compression Manufacturing Model

Corresponding to the implemented general model, the groups of S* metamodel elements for this specific manufacturing system were created during the specialization step. Fig. 5 showed an example of the specialized functional interaction framework. Fig. 6 was an example of a specialized relationship (relationships between interaction and logical Systems) in the tree view. Fig. 7 gave an example of the automatically generated model in Visio.

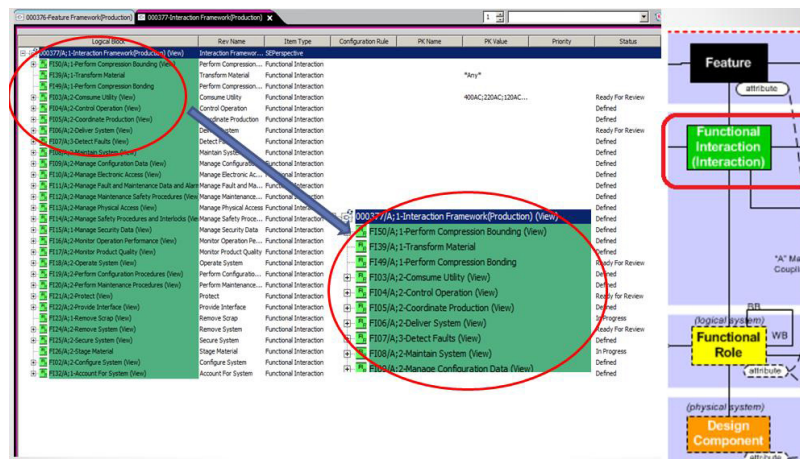


Fig. 5. Specialized Functional Interaction Framework

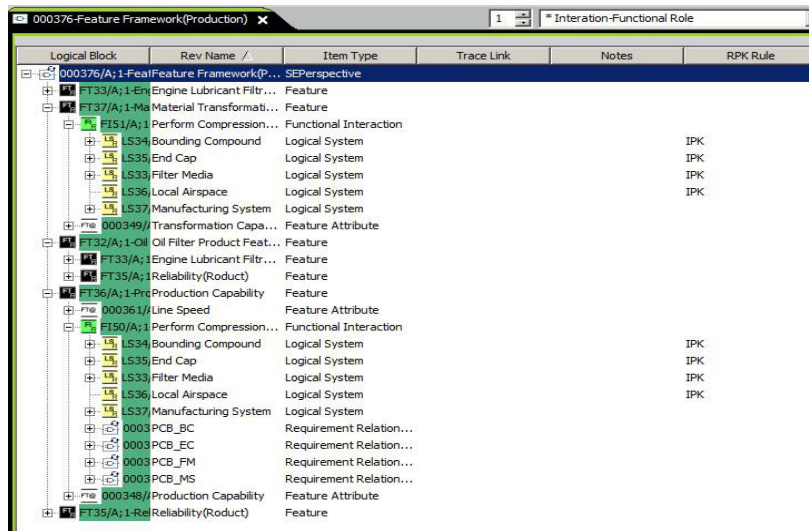


Fig. 6. Specialized Relationships between Interaction and Logical Systems

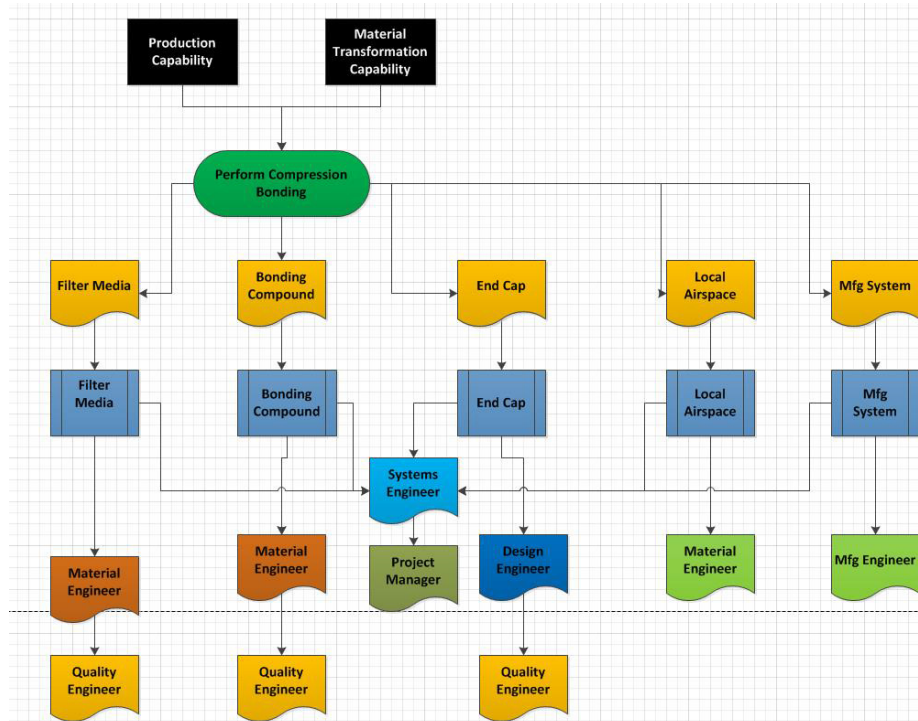


Fig. 7. Specialized Oil Filter Model in Visio

4 Discussions

The current implementation only covered the key portion in the general manufacturing model. This portion of the general model had the core elements and their relationships. The implementation of the portion provided a foundation for evaluation the implementation methodology and the implemented model. It also can help present the value of implementing systems engineering models into PLM platform. The rest of the model will be implemented in the future.

One of the most difficult parts of this study is to model structural relationships. There were so much different structural relationships in the model, and these relationships were normally n: n relationships. The whole model was a basically complex network. The current solution to provide a clear presentation of these relationships was to use the tree view. This solution was working well for the general model, but potential problems existed when the model was built for a higher complexity manufacturing process.

The specialization in this study was done manually. However, an automatic specialization should be very beneficial. There were two possible ways to automate the specialization step, the first one was to develop a new tool to generate product

specific models from the general model based on systems engineering's inputs; and the second one was to consider the model as a product structure and use product configuration functions in the PLM platform to generate new models. Further studies will be needed to identify a better solution.

5 Conclusions

A model-based systems engineering model can be implemented into a PLM platform.

References

1. Messaadia, M., Sahraoui, A., *PLM as linkage process in a systems engineering framework*. International Journal of Product Development, 2007. **4**: p. 382-395.
2. Bajaj, M., Zwemer, D., Peak, R., Phung, A., Scott, A. G., and Wilson, W., *Slim: collaborative model-based systems engineering workspace for next-generation complex systems*. INCOSE IW 2013 - MBSE Workshop, 2011: p. 1-20.
3. Schindel, W. *What is the Smallest Model of a System?* in *Proc. of the INCOSE 2011 International Symposium*. 2011. International Council on Systems Engineering.
4. Schindel, W., *System interactions: Making the heart of systems more visible*. Proc. of INCOSE Great Lakes Regional Conference, 2013.
5. Schindel, W., *Pattern-based systems engineering: An extension of model-based SE*. INCOSE IS2005 Tutorial TIES, 2005. **4**.
6. Gecevska, V., Stojanova, T., Jovanovski, B., *Product lifecycle management tools*. Annals of Faculty Engineering Hunedoara, 2013. **1**: p. 219-222.