INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

DESIGN 2012

PROPOSAL FOR FUNCTIONAL PRODUCT DECRIPTION AS PART OF A PLM SOLUTION IN INTERDISCIPLINARY PRODUCT DEVELOPMENT

M. Eigner, T. Gilz and R. Zafirov

Keywords: model-based systems engineering, product data management, product lifecycle management

1. Introduction

The realization of diverse functionalities of technical products has been constantly shifting from mechanical to mechatronic components over the years, whereby functions get more and more important as a common basis for interdisciplinary development. The cost of component integration during the design process is increasing due to this strong involvement of different disciplines. Therefore, it is in fact the different development processes – from mechanics to electronics, to software – that must be integrated.

Furthermore, software is playing an ever-increasing role in the context of modern interdisciplinary product development. Current research initiatives focus on technology advancements involving software-intensive embedded systems in technical products – the so called Cyber-Physical Systems [Broy 2010]. Software will be determining even more product functions in the future and is dominating the design process in many aspects already [Kühnl 2010]. Therefore it is essential to increase the transparency between disciplines within the design process through expanding the design methodology with methods that are utilizable by software engineering.

Systems Engineering could be such a method. It covers the domains of technical knowledge and engineering management. It is a multidisciplinary approach with the objective to develop balanced system solutions in response to diverse stakeholder needs [Friedenthal 2009]. It helps to engineer complex systems in cases where it is not humanly possible to overview and understand the whole system in detail. The interdependencies between individual components are managed; the system is kept consistent to the specification and satisfies all defined requirements.

Classical methods of Systems Engineering are mainly paper-, or document-based. An extended approach is model-based Systems Engineering, where computer stored system models allow capturing complexity at many different levels. Coming from the direction of Systems Engineering, various modeling languages have been established, which allow for a model-based specification of a product on a general level, comprehensible for all disciplines involved in the design process [Friedenthal 2009]. The issue of costly component integration during the design process could be tackled by using such modeling languages. These would allow for defining correlations between system requirements, structure and behavior.

Introducing a model-based interdisciplinary system specification to the design process does not completely solve the problem with the integration of the different design processes. The reason: the design process is being managed in Product Lifecycle Management solutions. In order to be useful as a collaboration medium between system requirements and different disciplines' engineering BOMs, the system specification model has to be managed on the PDM backbone, which covers most workflows and the important release stages within the design process.

To summarize, the transparency between disciplines in engineering is not sufficient. Especially with software gaining an increasing role within modern products, new methods and platforms for interdisciplinary collaboration have to be defined. This paper presents a concept that addresses this problem on two fronts. On the one hand, the concept of Systems Engineering and other approaches and guidelines for product engineering have been used to define a so called Functional Product Description which is suited for discipline-independent product specification in the concept phase of engineering. On the other hand a data schema has been defined, which enables the integration of Functional Product Description models in PDM in order to properly support the engineering process.

2. Related work

In this section related work on the field of functional modeling, description of mechatronic systems and PDM integration of these descriptions is discussed briefly.

2.1 Conceptual product modeling

When it comes to modeling the first concepts of a product, functional modeling is essential. This is due to the fact the functions are solution- and discipline-independent. Functional models describe a system in an abstract way, with respect only to its goal [Pahl 2007] and not to its technical implementation. This leaves room for creativity in the design process, allowing for an analysis of alternative technical solutions for a product. Solution-independence also turns functional models into one of the most stable type of structure within the design process. The method for functional modeling is considered by some of the new languages for conceptual product modeling. The data schema proposed in this paper builds on this method as well. Therefore, the method of functional modeling and after that, a few languages for conceptual product modeling and some PDM integration activities are presented in this section.

2.1.1 Functional modeling

As mentioned, functional modeling provides an abstract method for understanding and representing an overall product. Its primary task is to support finding suitable solutions in design by creating discipline-independent functional models of a product.

A functional model contains an abstract description of the main goal of a product by stating its overall function (noun-verb combination) [Pahl 2007]. There are several types of functional models: flow-oriented, relational, user-oriented etc. [Lindemann 2008]. The most common representation form is the network structure. Flow-oriented functional modeling also allows for unordered lists or functional hierarchies: the overall function is subdivided into its partial functions, any of which can be further subdivided into its own partial functions. Moreover, the relations between functions in terms of matter, energy and signal turnover can be defined as well, building so-called functional chains as network structures. A typical representation of a functional model is a diagram, with the functions as separate blocks and the flowing medium as arrows.

Functional modeling has not been originally created to support computer aided engineering design. In general, is not considered as a model based approach: throughout the design process, functional models can be created as documents only. Moreover, functional models are not formal (fully interpretable by a computer).

One step toward model based design is the functional basis for engineering design, formed by Stone et.al. [Stone 2002]. It is about the definition of formalized methods (taxonomy, rules) for function description and finding ways to administrate product design knowledge in suitable repositories to accelerate, diversify and/or standardize decision finding in product design.

Function modeling enhances discipline neutral product architecture modeling during specification of product concepts. It can be utilized in PDM to be used in following engineering and downstream processes.

2.1.2 Languages for conceptual product modeling

Gausemeier et.al. propose a Conceptual Design Specification Technique for the Engineering of

Complex Systems (CONSENS). With the help of different views, the system is described. Each view is represented by a partial model visualized by a diagram. The model elements in a partial model can have interrelations between the modeling constructs. The main partial models are requirement, environment, system of objectives, active structure, shape, behavior, functional and application scenario model. The principle solution is the concept model, based on a system of coherent partial models [Gausemeier 2011].
SysML is understood as a method- and tool-independent graphical modeling language for specification, analysis, design, verification and validation of systems, which is standardized by the Object Modeling Group (OMG).
ModelicaML is a graphical modeling language based on an UML Profile with Modelica semantics. ModelicaML facilitates the efficient and effective creation of executable system specifications and system analysis model that can simulate both time-discrete and time-continuous behavior of systems in Modelica [Schamai 2009]. ModelicaML defines new diagrams and re-uses diagram types from UML and SysML. Due to close relation between Modelica and ModelicaML, models described in ModelicaML can be transferred in Modelica language syntax. These models are executable by a Modelica compiler.
Paredis et.al. worked out a specification for translate SysML into Modelica, to combine the analytic power of Modelica with the descriptive freedom of SysML [Paredis 2010].
The focus of this paper is to make use of languages for concept modeling and give a first proposal for integrating these models in a PLM solution. SysML, being an open, standardized language, is considered as a valid candiate to start with.

2.2 PDM integration activities

Model and data interchange standards are important for data exchange. The XML Metadata Interchange specification (XMI) standardized by the OMG supports interchange of model data between tools for MOF-based modeling languages.
Another exchange standard for interchange of Systems Engineering is STEP AP233. AP233 is standardized in ISO EXPRESS language. In addition to this a XML schema is existent based on the EXPRESS to XML bindings but which is not as powerful as the EXPRESS definition. In the SEDRES project the STEP AP233 was formulated for integration of Systems Engineering data in PDM systems. AP233 supports program management and system modeling, which integrates system structure and behavior. Due to significant overlap with Systems Engineering aspects, the AP233 concepts of functional and system breakdown structures were standardized by the PLCS project as joint modules [Eckert 2005]. In the OMG a working group for SysML and AP233 is performing a mapping of the common data constructs between them for an AP233 based change management for SysML.
PDM system vendors have been working on integrating requirements management, functional, logical and physical processes. This is realized for example by a built-in graphical modeling environment for architecture with integration of detailed proprietary simulation models in TDM/PDM.
The proposal in this paper represents a more comprehensive approach for addressing the whole lifecycle with PDM integration on the Functional Product Description level.

3. Extended model-based approach in virtual product engineering design

Model-based Design is central in virtual product development. Models can be for example topological, physical, process-oriented, geometric or mathematical models [VDI 2206 2004].
In model-based Systems Engineering (MBSE) formal models are used for specification, structuring of complex engineering problems, to capture relations between properties and for analysis on a higher structural level. Stakeholders from various disciplines are involved in the design and development of a complex system. Every stakeholder has a different view to the specification, because of the specific discipline [Friedenthal 2009]. The methods of model-based Systems Engineering can help to design a multidisciplinary product in an abstract way.
The VDI 2206 [VDI 2206 2004] defines a systematic approach for developing mechatronic systems, but is not addressing all aspects of Model-based Design. The focus in this paper lies on the left wing of

the "V", whereby it's extended with the use of methods from model-based Systems Engineering (see Figure 1). Three views of modeling can be identified:

Modeling and specification
In this early specification the system is described by qualitative models, like requirement, function or system oriented models. The models are descriptive and cannot be simulated. In this view an interdisciplinary language like SysML can be used to specify the first design. A "computer-stored" common system specification model can be evolved, to support the specification process for all involved disciplines.

Modeling and first simulation
In this view quantitative aspects are integrated in an interdisciplinary view. This can comprise of simulation models (in e.g. Matlab or Modelica) for multidisciplinary simulations.

Discipline specific modeling
In this view the system is modeled more precisely in a discipline-specific way. These are mainly tool-dependent CAx models that represent discipline-specific aspects. An example for such models can be a mechanical CAD model, which contains the concrete geometry representation and many other properties, like e.g. mass.

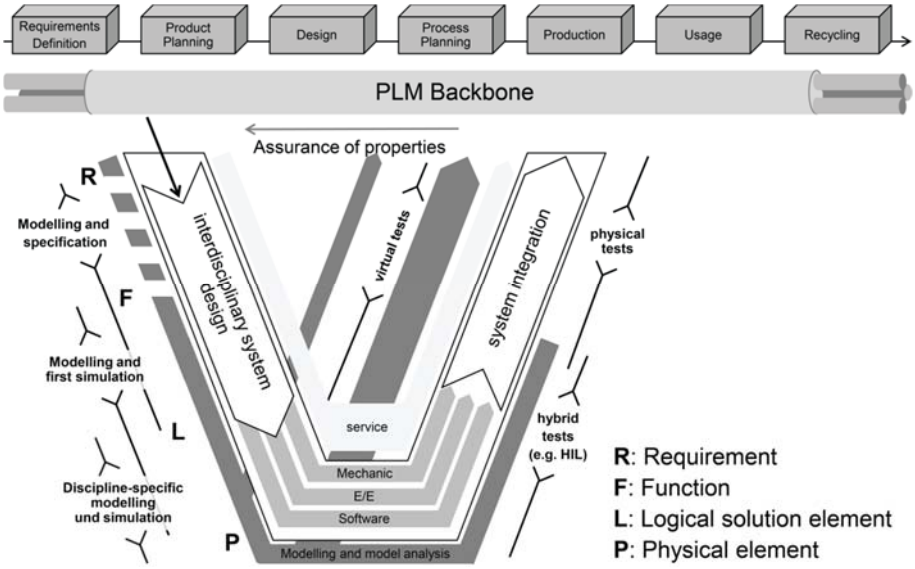

Figure 1. Extended V-model based on VDI 2206

Requirement definition is the starting point of development. It reflects more or less the abstract idea in the form of customer needs or user requirements. The idea needs to be specified in the requirements analysis and translated into logically consistent, technical developer requirement model. This is marked with **R** in Figure 1.
In the phases of product planning and design an interdisciplinary system design is essential for bringing all disciplines together and forming a functional solution of a technical system. Starting with a rough functional description, this concept can be refined step by step. The decomposition into functions, marked with **F** in Figure 1, provides an initially discipline- and solution-neutral view.
A solution concept is described through the definition of logical components (marked with **L** in Figure 1), that realize (implement) the functional elements. This solution concept includes logical and physical behavior and structure.
Formal and semi-formal languages, such as UML or SysML and simulation-based languages, such as Matlab/Simulink/Simscape or Modelica support the system design. At the end of interdisciplinary system design, virtual tests can assure product properties. Virtual tests can be seen as first system integration in a virtual environment. The incremental and integrated assurance of properties with virtual tests validates the system under development against requirements and test scenarios.

Based on these first simulations and the functional description each discipline can start with its discipline specific detail design, which results in physical elements of the system, like hardware parts or software code (marked with **P** in Figure 1).

3.1 Example of a Functional Product Description

The Functional Product Description relates to the view of modeling and specification (Figure 1, top left) and describes the system from a function-oriented perspective. This includes the requirements, functional elements and logical system elements (**R**, **F**, **L**). For modeling these aspects we focus on SysML due to the fact, that SysML is a modeling language standardized by OMG and many tools for modeling are existent. Tools like Magicdraw, Enterprise Architect or Eclipse with the plugin Papyrus allow modeling with SysML. In addition to SysML modeling tools, requirement management tools like Doors, RequisitePro or MKS provide powerful support for modeling requirements. Connectors between SysML editors and requirement tools are commercially available.
Figure 2 depicts an example for a Functional Product Description model of a windshield wiper in SysML. Requirements are modeled in a hierarchical requirements diagram. Functions and logical system elements are modeled hierarchically in a block definition diagram. They are also connected with each other within internal block diagrams, which represent a view of the functional and the logical architecture.

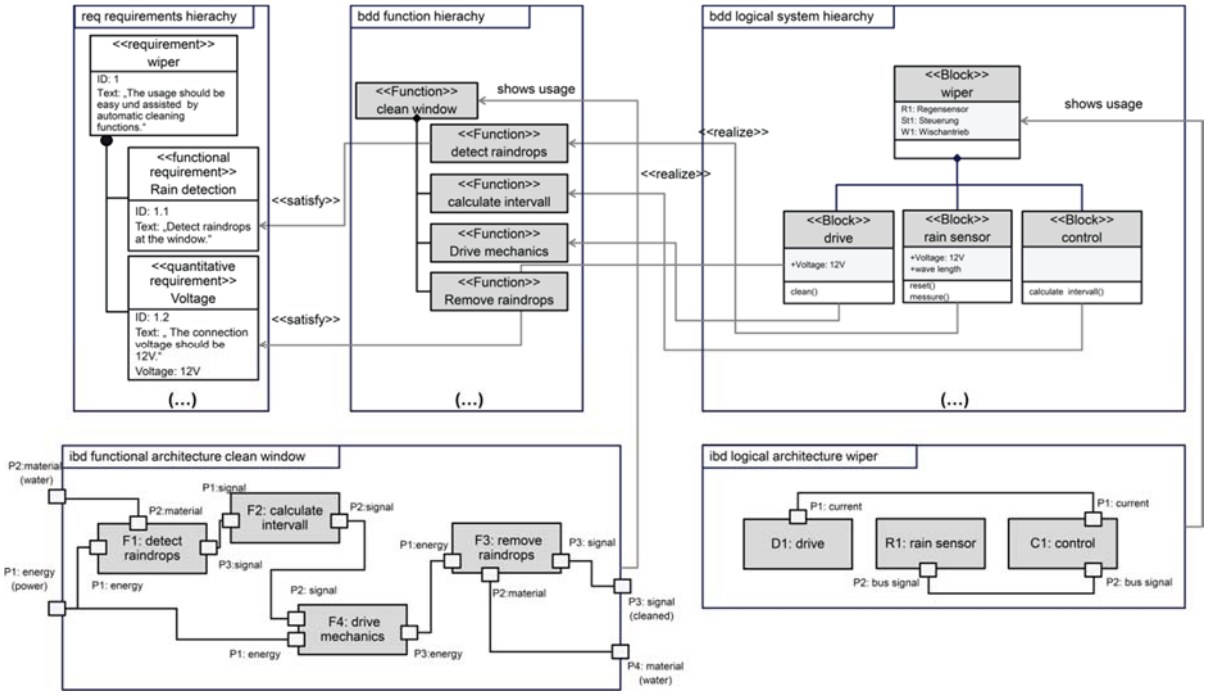

Figure 2. Example for a Functional Product Description

Between two different model elements, an allocation relationship can be established. The allocations can be constrained to different types. In SysML standard allocations are predefined, which can be used. In our example the allocation "realize" is added as a relation between logical and functional system elements. The element "rain sensor" realizes the function "detect raindrops". In SysML, the allocation "satisfy" can be used between requirements and properties of other system elements. In our proposal this allocation is extended, so that a function can satisfy a requirement. If these cross-references are defined, requirements can be related to the functional and logical architecture. If, for example, the windshield wiper in figure 2 has to be implemented in a truck and the required connection voltage changes, this change of the requirement can be traced to the affected logical or functional system elements.

4. Data management for Functional Product Description

This paper focuses on the specification of a product in a functional way. Product development may start with loose definitions of requirements in a document-based way and not with the full or precise definition of requirements in a model. Therefore, the design process according to the V-model presented above should be viewed as incremental loops that steadily refine the product by analyzing, modeling and validating the system. Iterations should begin at the smallest "V".

At the descriptive stage (Figure 1, Modeling and specification) functions of the system are analyzed and modeled. Moreover, solution elements for these functions are selected and the requirements are refined upon initial model analysis. Iteration loops at the stage of modeling and first simulation (Figure 1) involve the generation of simulation models using the proposed logical solution elements from stage 1 and the use of these models for a first validation of the roughly defined requirements to help create a more precise descriptive model. Versioning as an essential functionality of PDM can support this. Later the iteration loops switch to more specific virtual test iterations with detailed information until it comes to hybrid and physical tests.

Every iteration means increase of knowledge on the product. At the same time it signifies change. Engineering change is one of many examples for design process workflows that are mostly managed on the PDM backbone. It is there, where product information, design processes and all involved persons are organized together. The modeling and specification view (Figure 1, top left) at the early stages of product development is there for the same purpose: to manage the system and to verify all critical aspects, involving all stakeholders in design. It can best serve its goals when managed in PDM.

A good example for beneficial management of system specification models in PDM is collaboration during the design process. Based on the system specification model, different aspects of the product can be "frozen" and "released" for detail design. Furthermore, important insights (e.g. component parameters) from certain discipline-specific design and simulation can be fed back into the system specification model for other disciplines to use or even for later co-simulation.

Management of system specification models in PDM can also be useful during the design stages and later on along the product lifecycle as a medium for tracking and back-tracking impact influence. A good example here is being able to track through the functional description, which requirements a change in the product structure would affect and vice versa.

4.1 Content of the Functional Product Description

The Functional Product Description model should grow consequently during design. Therefore, the different stages of the model elements should be managed (as versions, with change tracking etc.). SysML editors can assist to model the first concept of a system which is under development in a graphical way. This can be integrated as simplified instance model of a data schema in a PDM system. The following three model perspectives have been identified as different levels of PDM integration:

Hierarchies

Requirement, function and logical element structures can be seen in a hierarchical way. Therefore, these structures can be stored in a PDM system hierarchically, like engineering BOMs. This view of models as hierarchies is PDM-centric. It is considered as the first, most rudimentary level of PDM integration for the Functional Product Description.

Typed internal connections between elements via ports

Hierarchies are mainly used to maintain a good overview of complex systems but if the functional or logical system elements are considered with their internal structures, hierarchies are not sufficient. Schematic connection diagrams, e.g. in Modelica or internal block diagrams in SysML show the internal references in a system element. These network structures are currently not implemented in PLM solutions. Internal connections use ports which are based on the same port type, so that a connection is valid. Connections, ports and other entities that have a specified type will henceforth be referred to as "typed".

Cross-references with allocations between different model elements

Cross-references are relationships between different types of model elements, which can be modeled with allocations in SysML. The advantage of model-based Systems Engineering is that these cross-

references can be managed together with the system elements. PDM integration on this level is essential, since it enables traceability, e.g. through referencing requirement to function elements and those to elements of the logical or physical architecture.

4.2 Data model for product data management

Existing PDM systems like Teamcenter 8.3 from Siemens Industry Software allow the capturing of functional descriptions in Microsoft Visio. Catia V6 Systems with Enovia VPM from Dassault Systèmes provide a special built-in editor for modeling a Functional Product Description and simulation models in Modelica. In both cases usable implementations and concepts are existent, but the specification is not open enough and mature usable for a seamless model-based Systems Engineering. Because of this we propose a more open data model for PDM.
Figure 3 shows a simplified data schema for the integration of the Functional Product Description with the perspective of hierarchies, cross-references between different model elements and the typed internal connections in a PLM solution.

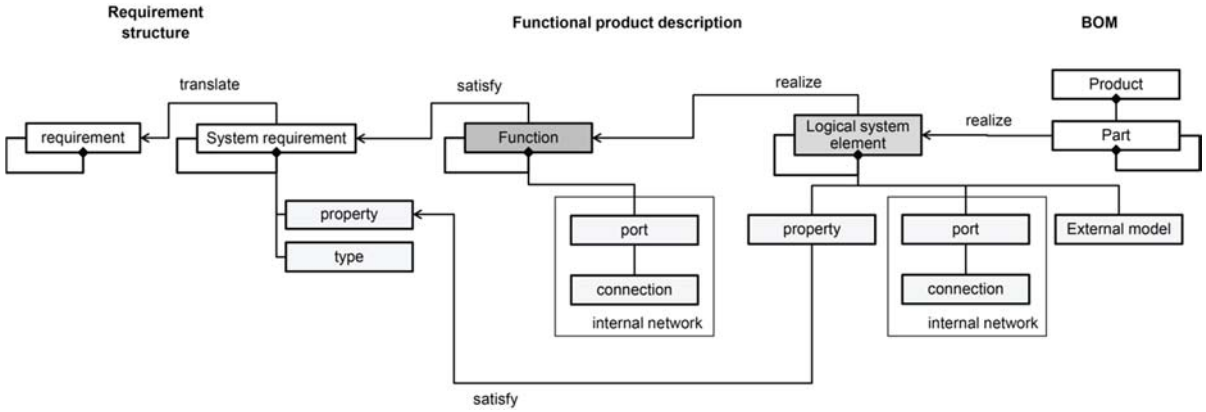

Figure 3. Simplified data schema of the Functional Product Description model

The capability to manage requirement and the BOM structure is already existent in PDM data concepts, so that for this paper these structures are considered as given (shown to the left and to the right on the figure). In early development BOM structures may not be instantiated in PDM. Still, along the further development process, logical system elements are realized by specific BOM parts. In Hence, the Functional Product Description can be connected to the BOM structure whenever it appears in PDM during the engineering process.
Functions can be modeled in a simple way as hierarchies (see section 2.1.1). This can be modeled in SysML block definition diagrams (see figure 2) and extracted to a hierarchical function structure in PDM.
Logical system elements are similar to functions but represent the realization of functions with a physical effect and have defined properties. A system element can hold external models as binary files. This could be for example a Modelica or Matlab/Simulink/Simscape model file, which has been developed to represent and analyze its corresponding logical system element. The logical system element is linked to a part of the BOM which represents the physical part.
A function or a property of a logical system element can “satisfy” a quantitative or functional system requirement. A logical system element “realizes” a function. This represents the cross-references between different model elements (see section 3.1) within the data model.
The linking of functions or logical system elements at the same level with each other is done with typed ports and connections between them. This internal network cannot be easily integrated in a PLM solution. Figure 4 shows a possible simplified data schema for the connection between typed ports.
In the general case, system element A and system element B, which can represent a function element or a logical element, are connected with each other. This can be an internal block diagram in SysML or an equivalent connection diagram in Modelica. The connections are specified by the use of port types. For a proper connection it is important that the ports are from the same type. In this case the

connection direction is not considered. In the right part of Figure 4 a concept for a data model is shown, which allows the management of the ports and the connection between the system elements A and B, which is essential for Systems Engineering.

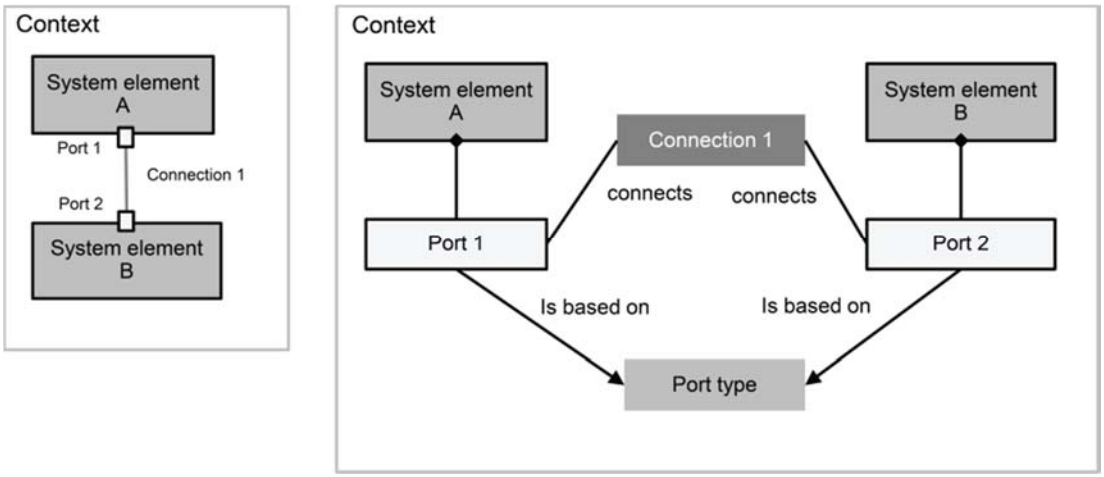

Figure 4. Core-concept for a data modell with typed network connections

4.3 Correlation between the Functional Product Description data schema and STEP 233

The Functional Product Description in our concept is a simplified data model of a system model in regard to the heavyweight STEP AP233. Table 1 shows basic mapping of data model AP233 and the Functional Product Description and important modeling elements of SysML. The focus is on the subpart of AP233 for system modeling and especially on the definition of requirements, function based behavior and system design hierarchy.

Table 1. Basic mapping of AP233 and SysML with proposed Functional Product Description

STEP AP233	Functional product description	SysML
	Hierarchy	
Requirement view definition, Requirement version, Requirement	System requirement	Requirement with additional stereotype
Functional representation item, Task Step, System View Definition	Function	Block with stereotype Function or Activity
System View Definition, System Version, System	Logical system element	Block with additional stereotype
Assembly Component Relationship	Composition Association	Composition Association
	Cross-references	
Requirement satisfied by	Satisfy	Satisfy
View definition relationship + Classification (‘Allocate’)	Allocate, Realize	Allocate
	Typed connection	
Interface Connector	Port	Port
Interface Connection	Connection	Connector

The Functional Product Description is a simplified view on the system. Its data model for PDM integration (presented in section 4.2) holds an explicit reduced view on the requirements, functional and logical perspective. Systematic methods for function modeling are explicitly supported by the data model, due to the function element and the functional architecture with ports and typed connections, which is not part of AP233. The Functional Product Description data model presented in this paper is

proposed to provide an easier access to methods of model based Systems Engineering for organizations that are involved with interdisciplinary product design. Its scope is focused on hierarchical and internal structures and cross-references between model elements. The advantage of integrating the Functional Product Description in PDM is to enable tracking of requirements, functions and logical system elements from a common system perspective, which should support a system oriented management in regard to ISO 9001.

5. Conclusion and outlook

An integration of Functional Product Description is important for enabling interdisciplinary product development and collaboration. For the integration a data model is proposed, which is related to building blocks of SysML. It enables model based Systems Engineering. The data model implements three perspectives: hierarchies, cross-references between model elements and typed connections within model elements. These perspectives should enable the management of functional and logical architectures in a PLM solution.

The future extensions of the proposed data model include the implementation of further system modeling aspects on the PDM backbone, like: parameterized system analysis views that could be further used for simulation, system behavior representations that are especially interesting when it comes to refining requirements with behavior definitions. The data model has also to be extended to support variants, for example for modeling alternative logical solution elements for a function. Directed connections with input and output ports, that extend the internal representation of model elements and analysis views, could also be integrated in the data model.

References

Broy, M., "Cyber-Physical Systems - Innovation durch softwareintensive eingebettete Systeme", Springer, Heidelberg, 2010.

Eckert, R., "STEP AP233 + Standard PDM = Systems Engineering PDM?" In Proceedings of 11th international Conference on Concurrent Engineering, W. Mansel, ed. CCE Press, Munich, 2005.

Friedenthal, S., "A Practical Guide to SysML - The Systems Modeling Language", Morgan Kaufmann Pub, 2009.

Gausemeier, J.; Dumitrescu, R.; Tschirner, C.; Stille, K.S.: Modellbasierte Konzipierung eines hybriden Energiespeichersystems für ein autonomes Schienenfahrzeug. Tag des Systems Engineerings 2011 (TdSE) , Nov. 2011.

Kühnl, Claus: ""Software gibt den Takt vor"", in: "[me] - Mechatronik & Engineering", Jhrg. 2010, HeftNr. 2, AGT Verlag Thum GmbH, Ludwigsburg, 2010, S. 24-25.

Lindemann, U.; Ponn, J.: "Konzeptentwicklung und Gestaltung technischer Produkte", Springer, Heidelberg, 2008. - ISBN: 978-3-540-68562-3, ISBN: 978-3-540-68563-0.

Modeling of Executable Behavior Using Graphical Notations." In Proceedings 7th Modelica Conference, P. Fitzson, ed., Como/Italy, 2009.

Pahl, G., "Konstruktionslehre; Grundlagen erfolgreicher Produktentwicklung; Methoden und Anwendung", Springer, Berlin; Heidelberg; New York, 2007.

Paredis, C.,"An Overview of the SysML-Modelica Transformation Specification", OMG SysML PortalURL, last visited: 11.01.2011

Schamai, W., "Towards Unified System Modeling and Simulation with ModelicaML:

Stone, R. B., "A functional basis for engineering design - reconciling and evolving previous efforts." Research in Engineering Design 2002, 2002, 65-82.

VDI 2206, "Entwicklungsmethodik für mechatronische Systeme - Design methodology for mechatronic systems." Beuth, 2004.

Dipl.-Ing. Torsten Gilz
Research Assistant
University of Kaiserslautern, Institute for Virtual Product Engineering
Gottlieb-Daimler-Straße 44, Kaiserslautern, Germany
Telephone: +49 (0)631 205 3787; +49 (0)631 205 3872
Email: Torsten.Gilz@mv.uni-kl.de
URL: http://vpe.mv.uni-kl.de