Interdisciplinary Product Development - Model Based Systems Engineering

Martin Eigner
Torsten Gilz
Radoslav Zafirov

1 Introduction

Innovative interdisciplinary product development requires a rethinking of current methods, processes, IT solutions and organizational forms. In particular, there is a lack of support from appropriate IT solutions for the functional specification and design of system architectures. For the interdisciplinary system modeling the design phase has limited IT support. Electronics and software are gaining an ever increasing share in the product development process (PDP). Design methods of these disciplines should be put to the test and their suitability for a modern interdisciplinary design approach should be reviewed. Model Based Systems Engineering (MBSE) could act as an integrative design collaboration method and form a bridge between the various engineering disciplines. System modeling languages such as SysML that allow for interdisciplinary system description are presented here as an "enabler" for the MBSE. On a more concrete level, simulation languages such as Matlab/Simulink or Modelica allow for early multidisciplinary simulation in conjunction with the system description languages thus facilitating an early concept formulation. This article elaborates on new methods, processes and IT solutions for interdisciplinary virtual product development. [1]

Model Based Systems Engineering (MBSE) is a multi-disciplinary approach with the goal to develop a balanced system solution in response to various stakeholders' needs [2]. MBSE helps the engineer to keep track of complex systems – to understand the context and the specification – in order to be able to fulfill all defined requirements. While traditional methods of systems engineering are paper or document based, the MBSE design process based on digital models. This allows for a better handling of complexity and facilitates information exchange among disciplines. Modeling languages introduce a model-based specification of a product on a general level that is understandable for all disciplines involved in the development. The current problems with integration of components during the development process can be addressed as early as possible through the use of such modeling languages by defining correlations between system requirements, functions, structure and behavior.

Moreover, in today's enterprises the accumulated data during the development process is usually managed in Product Lifecycle Management (PLM) solutions, so that only an integration of a functional product description in PLM can support the interdisciplinary collaboration at an early design stage.

2 Background
The implementation of various functionalities of products has shifted in recent years from purely mechanical to mechatronical components. Functions and behavior as a neutral description of a technical system are more and more important when it comes to an interdisciplinary design project. The complexity of the integration of mechatronic components during the Product Development Process (PDP) has increased by the strong participation of various disciplines. In addition, software is playing an increasingly important role in the context of modern interdisciplinary product development. When products intensively communicate among each other, new functionalities and features are possible. In such scenarios we are speaking of cyber-physical and cybertronic systems. Current research initiatives focus on technological progress with software intensive embedded systems in industrial products [3]. Software will allow for a number of additional product features in the future. This requires an even greater involvement of the software engineering discipline in the PDP. Therefore it is important to increase the transparency between disciplines.

There is a lack of established, i.e. industrially used methods, processes and IT solutions for the interdisciplinary development of multidisciplinary systems. So in turn, there is lack of intelligent and networked products and production systems. In the 60s and 70s a development methodology has been proposed in Germany, which is based on functions as a way of describing the product. It was meant for mechanical design and was not based on formal languages at that time. At the same time many software development methods based on formal languages were developed at an international level (see Figure 1).

Since the 50s Systems Engineering (SE) has been defined and used as an interdisciplinary document-driven approach for the development and implementation of complex technical systems in large projects, especially for the U.S. aerospace and military. This approach has been permanently extended from the perspective of software and electronics industry and offers nowadays modeling and simulation support of complex, highly networked systems.

PLM and ERP concepts that are in use today cannot meet the requirements of the interdisciplinary product and production system development in the early phase of the PDP. The data models in use until now are too rigid and the separation in monolithic systems for development (PLM), logistics, personnel and production management (MRP, ERP), customer and supplier integration (CRM, SCM) is obsolete. It no longer facilitates a tracking of complex relationships within the products themselves and their interaction with each other especially in the early phase of the development process.

At the various stages of the PDP, there is already quite a variety of application languages and tools:

- modeling and specification -> SysML (OMG standard), ModellicaML
- modeling and simulation -> Matlab/Simulink, Simscape, Modellica
- discipline-specific modeling and simulation -> here there are a number of M-CAD, E-CAD, CASE
The IT solutions at the second and especially the third listed stage are being widely implemented and accepted in the industry. No industrial application for interdisciplinary product and production system development are known at the first PDP stage, although international standards are existent. In the scientific community there are efforts to develop own modeling languages alongside the OMG standard SysML. The interfaces between the three modeling stages listed above are only partially existent and need to be developed and implemented in order to achieve a continuous, consistent design process.

In summary, the current situation can be characterized by a lack of integrative methods, processes and IT solutions – both on the administration side (PLM and ERP) and on the application side – as well as their lack of integration in a consistent modeling approach for requirements (A), functions (F) logical (L) and physical (P) description of interdisciplinary products and production systems. The approach of systems engineering and in particular the model based design could be a future guide to methods, processes and IT solutions for the development of interdisciplinary products and production systems.

3 State of Research

This section briefly presents the work associated with the field of conceptual product modeling. This includes the functional modeling of mechatronic systems, description languages and a discussion on the broader scope of PLM integration.

3.1 Functional modeling

The functional model is an abstract method for understanding and for the total value of a product. The primary task is to support the search for appropriate solutions in design and the creation of a discipline-independent model of a product.

A functional model provides an abstract description of the main goals of a product by specifying its functions (noun-verb combination) [4]. The main function is divided into its sub-functions, each of which can be further divided into sub-functions. This leads to a functional hierarchy. A typical representation of a functional model is a diagram that shows the functions as separate blocks and connections between them for flows such as material, energy or signal exchange.

Until now, functional models have not been considered as a part of a model-based approach, but rather understood as a document-based solution finding method. In addition, functional models are rarely formal and thus not interpretable by a computer.

A step in the direction of model-based design is the approach of Stone [5]. It's about the definition of a formalized approach based on taxonomies and rules for the description of functions that describe the product design.

3.2 Modeling languages for conceptual product modeling

Gausemeier et. al propose the specification technique CONSENS (CONceptual design Specification technique for the ENgineering of complex Systems) for describing self-optimizing mechatronic systems. A system can be described with the help of different views. Each view is a partial model as represented in the form of a diagram. In a sub-model, specific details are shown. The main sub-models describe environment, the target system, behavior, use cases, requirements, functions, structure and form of action. The concept model is formed by the sum of coherent partial models [6], [7].
Another language for describing systems is SysML which has been proposed by the OMG. SysML is seen as a language for independent graphical modeling in the context of specification, analysis, design, verification and validation of systems.

There are efforts to integrate simulation models based on SysML and UML system models. ModelicaML is a graphical modeling language based on a UML profile to describe simulation models in Modelica with semantics similar to SysML. ModelicaML facilitates the creation of descriptive models using the methods of model-based systems engineering, which can also be executed in Modelica [8]. Paredis et.al. also proposed a specification for the model transformation of SysML models into Modelica models in order to combine the analytical advantages of Modelica with the description freedom of SysML [9].

### 3.3 PLM integration activities

An integration of the system model in PLM is needed, in order to consistently support the interdisciplinary product development process. Mainly common standards are considered for the integration into a PLM solution.

The XML Metadata Interchange specification (XMI) standardized by the OMG model supports the exchange of data between tools for Meta Object Facility (MOF)-based modeling languages, but does not define data schema.

Another systems engineering model exchange standard is STEP AP233. STEP AP233 has been defined in the SEDRES project group for the integration of system engineering data in PDM systems. AP233 supports the system modeling that integrates the system structure and behavior [10].

In the OMG, a working group is still investigating the correlations between SysML and AP233. Goals are the mapping of the common data constructs as well as a change management for SysML.

### 4 Extended model-based approach to virtual product development

The model-based development is of central importance within virtual product development. Models can be, for example, topological, physical, process-oriented, geometric or mathematical.

In model based systems engineering (MBSE) models for description and specification are used to facilitate the structuring of complex technical problems. Thereby relationships between properties are being captured for the analysis at a higher structural level. Stakeholders from different disciplines are involved in the conception and development of any complex system. Each stakeholder has a different view on the specification [2]. The methods of model based systems engineering can help to describe a multi-disciplinary product in an abstract way. The VDI 2206 guideline defines a systematic approach for the development of mechatronic systems. This article focuses on the left wing of the "V"-model and extends it with the use of methods from the model based systems engineering (see Figure 2).
The three views of the models that are identified as:

- **Modeling and Specification**
  A system is described with qualitative models. These include requirement, functional and system structures. These models are descriptive and cannot be simulated. Editors for modeling languages such as SysML serve as authoring tools.

- **Modeling and first simulation**
  Models at this level are quantitative for the most part, incorporate multiple disciplines and can be simulated (e.g. multi-physical simulation models). Simulation editors such as Dymola and Matlab/Simulink serve as authoring tools here.

- **Discipline-specific modeling**
  Models at this level (e.g. geometry and CAE models) have a very discipline-specific character. Authoring tools are here e.g. CAD systems or discipline-specific analysis and simulation software.

The requirements definition is the starting point of design. It more or less reflects an abstract idea of the product in the form of customer needs or user demands. In the requirements analysis that follows, the customer requirements are translated into logically consistent technical requirements. These are marked with an “R” on Figure 2.

A multidisciplinary system design that allows for the creation of a functional solution, which can represent all disciplines, is essential for the early stage of product development. This is often referred to as requirements engineering. Beginning with a broad functional and behavioral description, this concept can be refined step by step. The classification and description in the form of functions and sub-functions is marked with “F” on Figure 2. This provides a discipline- and solution-neutral perspective on the overall system as a first system specification.

The solution concept is described with the definition of logical components (labeled “L” on Figure 2) that implement functions and behavior. The solution concept comprises of the logical and physical behavior...
and of the structure of the system.

Semi-formal modeling languages such as UML or SysML and simulation-based modeling languages such as Matlab/Simulink/Simscape or Modelica support the interdisciplinary system development, so that at the end of the interdisciplinary system development specific product properties can be verified by virtual testing. So far there is no consistent data and information exchange between the model-based system specification, the first simulations and the corresponding disciplines. An incremental and integrated verification of properties through virtual testing requires the modeling of requirements and the definition of test scenarios.

The discipline-specific development is based on the initial simulations and the functional description. It addresses the physical elements of the system such as hardware components or software code (labeled as “P” in Figure 2). The majority of the CAx processes in the virtual product development take place here.

5 Data Management for the conceptual product modeling

A complete definition of requirements, functions and logical system elements is difficult to achieve directly from the start. Therefore, the development process should follow the “V” model presented above going in incremental loops that refine all aspects further. The iterations start with the smallest ”V”, through the virtual test iterations with detailed simulation models until it comes to the physical tests. Each iteration loop represents an increase of knowledge on the product. At the same time this means that changes may be made, which must be managed accordingly. The PDM system, as a core of a PLM solution, is suitable for this purpose since this is where key product information from everyone involved in the development process is gathered and organized.

Models at the stage of modeling and specification (Figure 2, top left) are created at the early stages of product development with the purpose of managing the product description and to examine all critical aspects, involving all stakeholders.

Based on the system specification model, different aspects of the product can be "frozen" and "released" for further use in discipline-specific design. Parameters of components for certain discipline-specific planning and simulation can in turn be used for other discipline-specific models.

Functional description models can serve as a medium for the tracking of changes to requirements, functions, behavior, logical and physical elements both during the early planning stages and later on along the product life cycle through their administration in PDM. An example for this is the facilitation of element allocation through the functional description which shows which requirements affect the product structure (of physical elements) and vice versa.

5.1 Tool support for the Functional Description

The functional product description refers to the stage of modeling and specification (Figure 2, top left) and describes the system from a function-oriented perspective. This includes the requirements, the functional and logic elements of the system. SysML is suitable for the modeling of these aspects. It is a standardized modeling language and many tools exist for modeling with SysML. XMI could be an excellent basis for data exchange but many tools do not fully support XMI. In addition to the SysML modeling tools such as MagicDraw or Enterprise Architect, requirements management tools such as Doors, RequisitePro or MKS provide strong support for the modeling the requirements.

Figure 3 shows a SysML model example of a windshield wiper system. The requirements are modeled in a hierarchical requirements diagram. Functions and logical system elements are modeled hierarchically in a
block definition diagram. The internal structure is represented by internal block diagrams, which represent the functional and logical architecture.

![Diagram of a functional product description](image)

**Figure 3: Example of a functional product description**

A mapping relationship can be established between two model elements. These so-called allocations may be restricted in several ways. Preset types of allocations exist in SysML. Beyond that, new allocation types can be defined. In the given example, the allocation type "realized" has been defined as a relationship between logical and functional elements of the system. The rain sensor “realizes” the function "detect rain." A property of a system element can “satisfy” a requirement. In our proposal, this allocation type has been extended so that also a function can “satisfy” a requirement. As long as such allocations exist, requirements can be associated with the functional and logical architecture.

If the wiper from Figure 3 were to be re-used in a different product (e.g. designed for passenger cars but to be integrated in a commercial vehicle), this could lead to changes in requirements: e.g. vehicle voltage change to 24 V. To implement this change in requirements, it is important to know all affected logical or functional elements of the system.

### 5.2 Contents of a functional product description model

SysML editors can help to graphically model a first concept of a system with the aid of diagrams. A data schema for the integration of modeled data in a PDM system is proposed in the next section. It can be reproduced in e.g. XML and contains the following different perspectives:

- **Hierarchies**
  Requirement, functional and logical element structures can be viewed in a hierarchical manner. These can be managed in PDM systems in a simple manner (similar to a bill of materials).

- **Cross-references between model elements of different types**
  Cross-references are the relationships between different types of model elements that can be modeled with allocations in SysML. The advantage of model based systems engineering is that the information on cross-references is managed along with the system element.

- **Typed internal connections among model elements of the same type**
Connection diagrams, e.g. in Modelica or internal block diagrams in SysML describe the internal structure of a system or a system element via connections between its elements based on ports of the same type.

Hierarchies are used primarily for the management of complex systems. When considering functional or logical relations within a functional product description, hierarchies are usually no longer sufficient.

5.3 Data Model for Product Data Management

Figure 4 shows a possible integration of the functional product description in a PLM environment. The PDM/PLM solution for the management of product data is shown at the top. In this article it is assumed that the PDM system already supports the management of requirement and BOM structures. The authoring tools throughout the different phases of the design process are listed at the bottom of the figure (e.g. “Doors” as a requirements management tool in the phase “planning”).

The functional product description is integrated between the requirements and the BOM. Functions can be modeled in a simple way as hierarchical structures (see section 3.1). These can be modeled as block definitions in SysML editors and extracted to XML.

Logical system elements represent the realization (implementation) of functions with specific physical effects and contain specific properties. A system element can be linked to other models as external files. These can be for example a Modelica, Matlab/Simulink/Simscape model files that have been created to represent and analyze physical properties of the system. Logical system elements are connected with the physical BOM, which contains M-CAD, E-CAD models, as well as software.

6 Summary and Outlook
An extension of the V-model according to VDI 2206 guideline was proposed in this article, which further addresses the challenges of the virtual model based product design.

In addition, a data model for the functional product description was introduced, which should provide easier access to the methods of model based system engineering for organizations, thus supporting an interdisciplinary product development. The focus is set on hierarchical and internal structures, as well as cross-references between model elements. An integration of the functional product description into a PLM concept involves the tracking of changes and influences of requirements, functions and logical system elements upon each other. The management of the functional description makes an early product documentation and quality assurance possible.

The proposed approach in the extended V-model is based on the model based systems engineering, which requires a fundamental rethinking of product development. The proposed data model implements three perspectives: hierarchies, cross-references between model elements and typed connections between model elements. These perspectives should allow for the management of functional and logical architectures in a PLM solution.

**Bibliography**


Contact
Prof. Dr.-Ing. Martin Eigner
Dipl.-Ing. Torsten Gilz
Dipl.-Ing Radoslav Zafirov
Technische Universität Kaiserslautern
Lehrstuhl für Virtuelle Produktentwicklung
Gottlieb-Daimler-Straße 44
67663 Kaiserslautern
http://vpe.mv.uni-kl.de