21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17
21-25 AUGUST 2017, THE UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER, CANADA

MECPRO² - A HOLISTIC CONCEPT FOR THE MODEL-BASED DEVELOPMENT OF CYBERTRONIC SYSTEMS

Eigner, Martin (1); Dickopf, Thomas (1); Schneider, Marc (2); Schulte, Tim (3)
1: Technische Universität Kaiserslautern, Germany; 2: :em engineering methods AG, Germany; 3: Schaeffler AG, Germany

Abstract
While today's products are increasingly developing into high-tech products in the meaning of Industrie 4.0, Internet of Things or Industrial Internet, the processes, methods and concepts for the development of such Cyber-Physical Systems or Cybertronic Systems are adapting very slowly to the new requirements. This paper introduces a holistic concept for the development of such cybertronic systems. The focus here is on the integration of two important partial results of the German research project mecPro², the mecPro² Process Framework and the mecPro² Architectural Framework. While the process framework designates the development process for cybertronic systems, the architectural framework defines how the system information required in the process is created, further processed and represented in a system model including its interrelations. Based on methods of Model-based Systems Engineering this concept especially helps to improve solution finding process as well as the consistency and collaboration in the early development phases.

Keywords: Cybertronic systems, Model-based engineering, Systems Engineering (SE), Early design phases, Product Lifecycle Management (PLM)

Contact:
Thomas Dickopf
Technische Universität Kaiserslautern
Institute of Virtual Product Engineering
Germany
thomas.dickopf@mv.uni-kl.de

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 3: Product, Services and Systems Design, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

Innovative and interdisciplinary engineering of consumer products and their production systems requires a rethinking of current design methodologies, processes, IT solutions, and the entire enterprise organization, especially since mechatronic systems evolve more and more to so-called Cyber-Physical Systems (CPS) or Cybertronic Systems (CTS). Main components of these CPSs are networked electronics with embedded software. However, a wide range of technical systems is centred on mechatronics, where mechanical and electrical engineering are the dominant disciplines. The term CTS has been coined as a consequent evolution of mechatronic systems in the context of an ubiquitous interconnection of these systems via open networks.

Engineering design processes are constantly changing and pose new challenges. Rapidly changing market situations in a global economy and an increasing number of customer requirements have a significant influence on the design process. The growth in product systems complexity results from products of larger varieties serving multiple markets as well as from multidisciplinarity in product design and development, particularly due to the increased use of embedded software in technical systems.

To fulfil the above-mentioned challenges, an interdisciplinary and holistic approach is needed, which includes a design process suited for cybertronic systems and a design methodology to specify the entire system as well as the interfaces to the discipline-specific design of its mechanical, electronic and software parts.

Because the results presented in this article has been generated within the German BMBF research project mecPro², chapter two will give a short overview of the research project and a positioning of the results with regard to the project context. While chapter three will give an overview about related work in the project context, chapter four will explain how these results are integrated in a holistic approach. At last, section five will conclude the results of this paper and give an outlook for further need for research.

2 THE RESEARCH PROJECT MECPRO²

As a joint research project of academic and industrial partners within the high-tech strategy "Industrie 4.0" of the German government, mecPro² (Model-based Engineering of Products and Production Systems) had the aim to increase efficiency in the development of Cybertronic Systems (CTS) by using Model-based Systems Engineering (MBSE) in an use-case oriented way.

The focus of the research project was the integration of information and data from all relevant disciplines involved in the development process. In particular, an integrated development process of Cybertronic Products (CTP) and Cybertronic Production Systems (CTPS) enables a collaborative development of the product, production processes and related production system and resources. This should be achieved by the concepts of MBSE and Product Lifecycle Management (PLM).

The results presented in this article have been developed within the scope of the above mentioned research project and illustrate the integration of two important partial results: the mecPro² Process Framework and the mecPro² Architectural Framework.

The Process Framework defines when and by whom, in the development process of Cybertronic Systems, system-specific information is used or is created. So far, this information has been recorded spread over several documents mostly in natural language. Frequently this has led to information gaps between organizational or corporate interfaces. These information gaps can be closed through the usage of formalized information, which is stored within one logical system model.

The mecPro² Architectural Framework describes such formalized information. In general, the Architectural Framework defines this information, its interrelationships as well as its kind of representation. Therefore, both frameworks are inextricably integrated in the mecPro² approach and have to be presented together.

To formalize the information of the development process the mecPro² approach uses the open and standardized modeling language SysML (Systems Modeling Language) and its mechanism to add additional domain specific vocabulary to its language area. These additional elements are primarily derived from the characteristics of Cybertronic Systems. In general, the mecPro² approach is not only suited for the development of CTS but also for mechatronic systems.

All artefacts, resulting from a model-based development process, have to be managed over the entire lifecycle of the system. The development of a concept to manage these artefacts via so-called System

Lifecycle Management (SysLM) solutions (Eigner et al., 2014) was as well an important part of the joint project but will not be further described in this article.

3 RELATED WORK IN MECPRO²

3.1 The mecPro² Process Framework

In general, the Process Framework is based, as seen as in Figure 1, on two essential parts: Process Modules and Process Activities. Both of them have specific descriptors, which describe the modules or activities and help to define uses and generates information who in a process, in which sequence and for which purpose. These descriptors are a name, a textual description, an aim, inputs and outputs, as well as a process role. Process Modules have also a list of activities as descriptor, which describes what activities belong to a specific module. Process Activities can as well be grouped as part of specific Process Areas like management or design activities. This allows a consistent description of the development process planned. The elements of the Process Framework are organized in a process library and can thereby be used for various development scopes through their generic description and modular design. (Cadet et al., 2015)

The Process Framework is described in the same three dimensions as the Model Framework of the mecPro² Architectural Framework shown below. The three dimensions are detailing, concretisation and variability. Detailing represents the information enrichment of the development process and its describing process model through its modules and activities. The concretisation includes the information enrichment/ adaption of the process model for the development within the scope of a typical business type (customizing) and/ or a concrete development project (tailoring). The selection of the required methods and IT-Systems takes place in this dimension. The detailing always leads to a concretisation of the process model at some point. Inevitably, variability occurs. Therefore, the Process Framework itself must be capable of depicting variability. One approach to solve this variability is given in Schulte et al. (2016a).

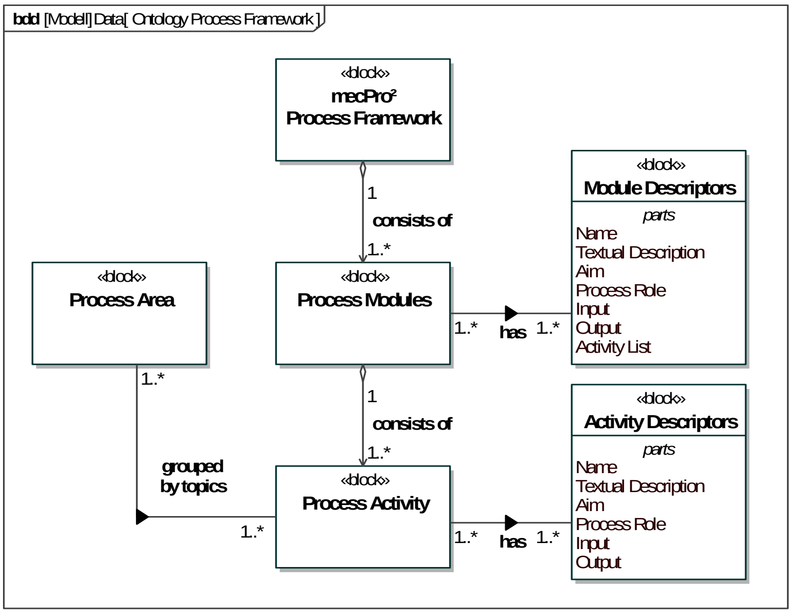

Figure 1. Relation between the Parts of the mecPro² Process Framework

3.2 The mecPro² Architectural Framework

The mecPro² Architectural Framework is embedded in the mecPro² Process Framework and defines the information that is needed by the Process Framework, its correlations and representation. The whole approach is model-based and supported by the modeling language SysML. The Architectural Framework consists basically of three essential parts, (1) the Model Framework, (2) Ontology & Profile, (3) Views & Viewpoints, which are described in detail below (Eigner et al., 2015). The left side of Figure 2 shows the general structure of the mecPro² Architectural Framework.

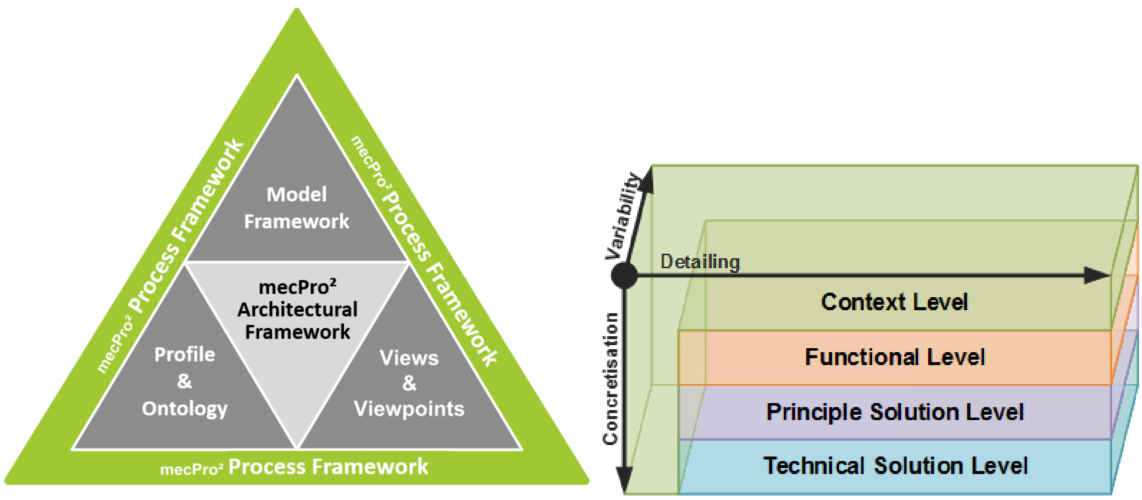

Figure 2. mecPro² Architectural Framework and the general structure of its Model Framework (Eigner et al., 2015)

3.2.1 The Model Framework

The mecPro² Model Framework, as part of the Architectural Framework, describes the design pattern for developing cybertronic systems. The resulting system model shapes the core of the description of the technical system and serves as an integration model between the discipline-specific partial models. The framework unites basic ideas of various development processes and methodologies in the fields of mechatronic, mechanic, electric/ electronic, software and systems engineering: the RFLP approach from the extended V-model by Eigner et al. (2012); the viewpoints and the translation of natural language requirements to a model-based requirements description of the SPES Modeling Framework (Pohl et al., 2012); the consideration of principle solutions (VDI 2221, 1993; Ponn and Lindemann, 2011) and the subdivision in requirement and solution space including the three axes fractionize, vary and concretize of the so-called Munich Model of Product Concretisation (Ponn and Lindemann, 2011). The description of the system is organized in four levels with increasing solution concretisation. Figure 2 (right side) shows the general structure of the Model Framework. The dimension detailing comprises the accumulation of information without the restriction of the solution space until further detailing is no longer possible without a solution concretisation. When this point is reached, the transition to a lower level takes place. This transition also results in variability, because the solution concretisation does not occur without the consideration of design alternatives. Each level describes a system from a structural and behavioural point of view. The following sections will give a short description of the four levels of the Model Framework. A detailed description is given in Eigner et al. (2015, 2016).

Context Level

System requirements arise from the context in which the system is being analysed. At the Context Level, a translation and synchronisation process takes place, where context-based system requirements will be translated from natural language into a model-based system description. At the end of the translation process, the system requirements are available in two languages: natural language requirements and as model-based requirements in SysML with a higher degree of formality. This approach has two major advantages. A maximum formalisation of requirements has great advantages for consistency and continuity in the development process. Stakeholders can be answered in the language in which they have formulated their requirements that may help avoiding misunderstandings. On the Context Level, the system as such will be described as a black box with its interfaces to elements in its environment as well as by its perceptible external behaviour. (Eigner et al., 2015; Eigner et al., 2016; Schulte et al., 2016)

Functional Level

The Functional Level is the first concretisation step and is mainly based on the model-based SysML approach of the FAS Method (Weilkiens et. al., 2016). The aim of this level is a mostly solution-neutral description of the system functionality based on the contexts in which it is used during its lifecycle. Based on the detailed descriptions of the expected behaviour at the Context Level, non-redundant functions are identified and placed in a hierarchical relationship by using various heuristics. The level's final result is a functional structure as known from the VDI Guideline 2221 (1993) that arose from the connectivity of system functions via material, energy and signal flows. (Eigner et al., 2015; Eigner et al., 2016; Schulte et al., 2016)

Principle Solution Level

On the Principle Solution Level, the technical aspects, which realize the desired function, are considered. Therefore principle solution variants should be systematically identified, analysed and evaluated in order to make an optimal selection with respect to the requirements. The selection of the best principle solution takes place using a Value Benefit Analysis, which is based on various criteria from the system requirements like timing constraints, effectiveness, performance or cost. The evaluation and selection should be made in two stages: the degree of fulfilment of a function considered by a solution principle and the degree of fulfilment of possible principle solution structures, which are based on the functional structure of the functional level. Regarding to the analysis and evaluation of variability, this level describes a starting point for an early simulation based on defined test cases. (Eigner et al., 2015; Eigner et al., 2016; Schulte et al., 2016)

Technical Solution Level

The Technical Solution Level identifies the technical solution components, which realize the system functions by applying the chosen principle. The result of this level is an abstract system structure whose individual system components are related to each other by their interfaces. These components combined deliver the expected system behaviour defined on context level.

On the technical solutions level the maximum concretisation of a solution, for which an organizational unit is responsible for, is reached. The elements of the solution cannot / should not be decomposed any further by this unit. The responsibility for these elements can be assigned clearly and without any overlapping to another organizational units. This is possible because the information of the Technical Solution Level represents the sum of information of the contexts of all solution elements. A special case occurs when elements can be classified as discipline-specific. In this case, the concretisation of solutions and partitioning is on the same level. (Eigner et al., 2015; Eigner et al., 2016; Schulte et al., 2016)

3.2.2 Ontology and Profile

The ontology identifies and describes the concepts, terminologies and correlations used in the system analysis and system design. It ensures the consistency of the Model Framework and is the knowledge base for further activities like process implementations, extensions by domain-specific concepts, integration of other approaches, automation and transformation concepts. Because the process activities and artefacts are related to the ontology, they are the basis for the data model for managing the model information during the life cycle in a system lifecycle management system. (Eigner et al., 2015).

The ontology is also the basis for creating the mecPro² SysML profile. Through the profiling mechanism of the Unified Modeling Language 2.0 (UML) new domain-specific vocabulary can be introduced with the aid of stereotypes into the model. The SysML itself is a profile of the UML. For example, these can be terms which are derived from the characteristics of Cybertronic Systems. The UML profiling mechanism allows a restriction of elements as well. This possibility has not been pursued within the scope of the research project in order to guarantee an intact meta model.

3.2.3 Views and Viewpoints

The views define the use of specific parts of the ontology and thereby it represent a section of it. A view is defined through the type of SysML diagram and the elements and relations to be shown in the diagram. Each view conforms to a so-called viewpoint. Viewpoints are defined by a necessity, an addressee, the ontology section, the stakeholder/role, specific rules and other attributes. The sum of all views and viewpoints represents the entire ontology. The concept applied in mecPro² has been developed based on the concept of Holt and Perry (2013) (Eigner et al., 2015).

4 INTEGRATION OF MECPRO² PROCESS FRAMEWORK AND THE MECPRO² ARCHITECTURAL FRAMEWORK

This chapter explains how the individual parts of the holistic mecPro² approach are mutually related and integrated. In the first section it is explained how the essential parts of the mecPro² Architectural Framework are interrelated, the integration respectively the embedding of the Architectural Framework into the Process Framework is discussed in the later section.

4.1 Relation between the parts of the mecPro² Architectural Framework

The three main parts of the Architectural Framework are closely related. Only through the continuous use of all components a formal and consistent system model can be developed.

The viewpoints of the Architectural Framework define the visual focus on the system during its development. For each level of the Model Framework there is a viewpoint, which considers the system during its specification based on different aspects. Each of these viewpoints has a variety of views on the system model. In general, there could be also other viewpoints and views on the system model, like from a management or production point of view. Each view clearly shows with the help of a SysML diagram a section of the system model and thereby a section of the system to be developed. Each view uses exactly one diagram type of the SysML. In this diagram, the corresponding elements of the SysML as well as the extensions added by the mecPro² profile can be used to describe the system. How these elements are generally related to each other is defined by the ontology. Accordingly, each view represents a section of the Ontology. Due to that, the sum of all views of one viewpoint maps the ontology on one level of the Model Framework, and accordingly the views of all viewpoints correspond to the entire ontology of the Architectural Framework. Figure 3 shows the relations between the parts of the Architectural Framework with regard to the development of a system model.

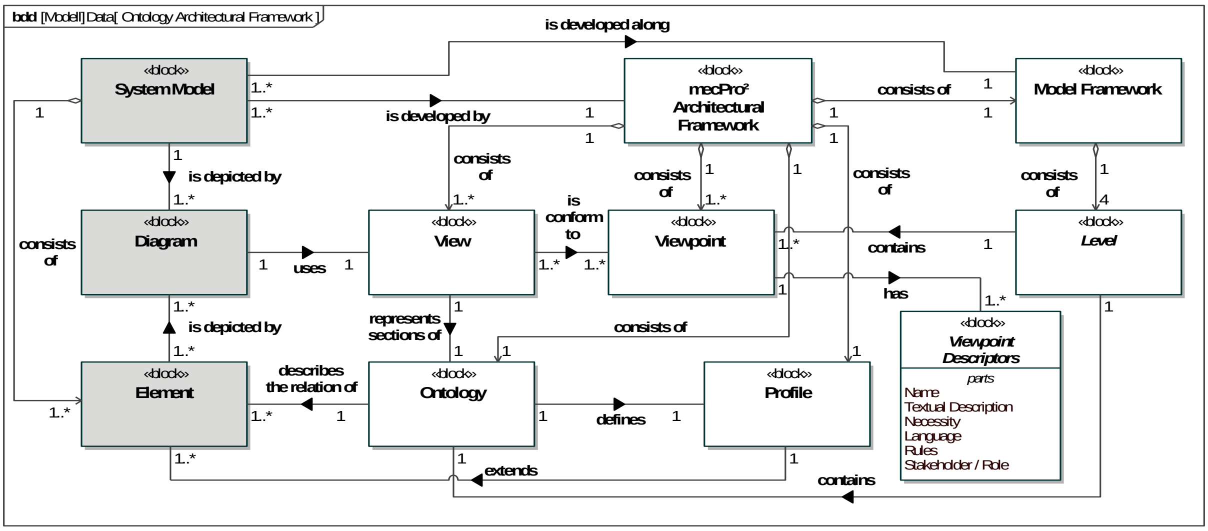

Figure 3. Relation between the parts of the mecPro² Architectural Framework

While Figure 3 shows the conceptual relations between the parts of the Architectural Framework, Figure 4 exemplifies the interrelationships between these parts with an example of the Context Level. The viewpoint associated with the Context Level of the Model Framework is the so-called SYSContextLevelViewpoint. This viewpoint is addressed to Requirements and Systems Engineers and utilized both SysML as well as the natural language to represent the requirements of the system under development. Views of this viewpoint are, for instance, the ContextUseCaseView, which is used to identify the main use cases of the system, or the ContextDefinitionView, which defines with which external systems the system of interest interacts during these use cases. Another view is the UseCaseActivityView. This view describes the system use cases in detail by using use case activities. The SysML diagram type associated with this view is, as shown in Figure 4, the activity diagram (the example shows the detailed description of the use case trenching of an autonomous excavator). The lower part of Figure 4 shows the ontology of the context level. The area with the thick border represents the elements, which are allowed for the use in the UseCaseActivityView and thus also in the activity diagram. Therefore the other, thinner bordered parts represent elements, which belong to other views of the Context Level. Elements of the ontology, which are included in more than one view are interfaces between these views. Each view must have at least one of these interface elements, otherwise there would be isolated parts within the system model.

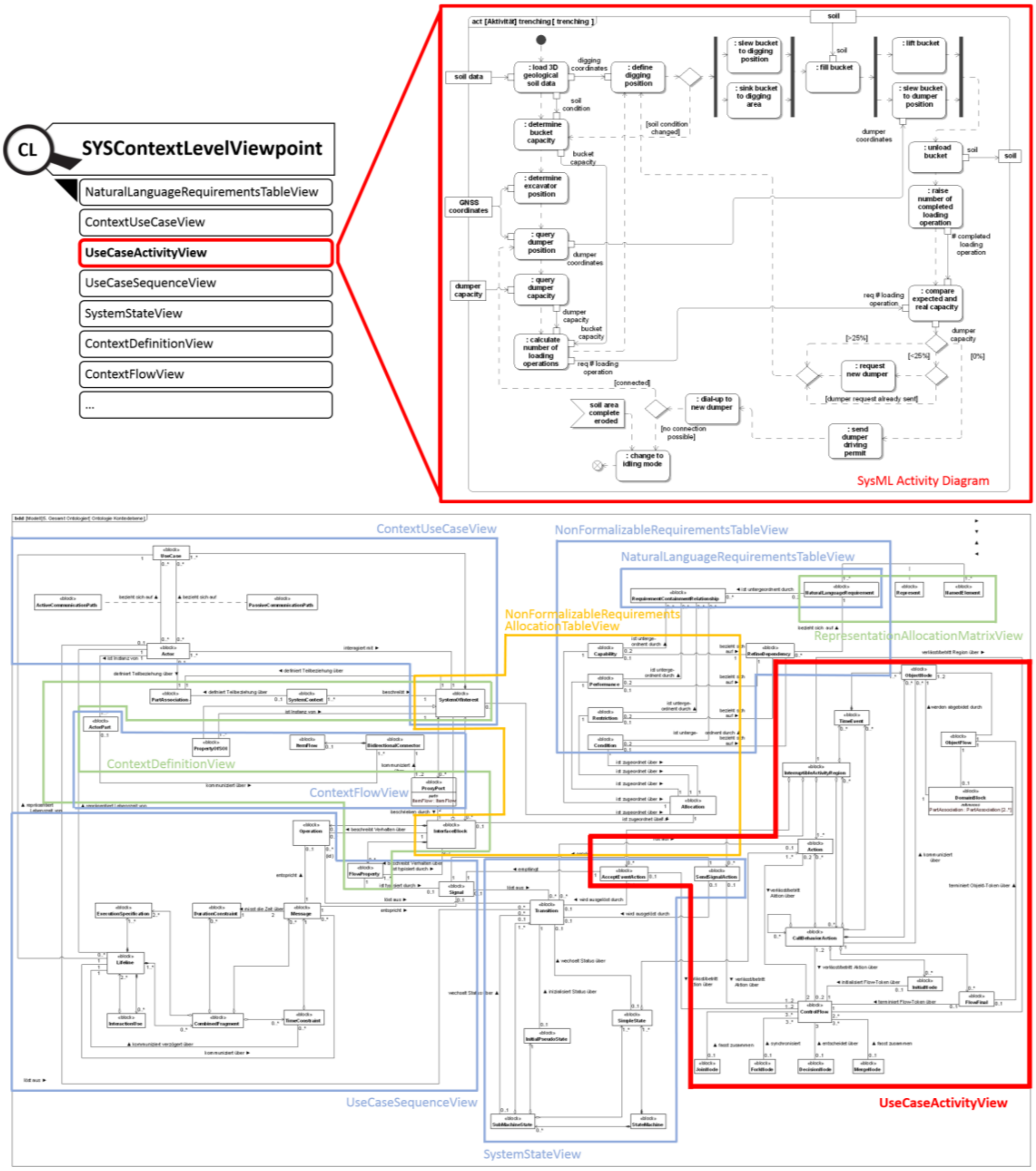

Figure 4. Representation of the relations between the parts of the mecPro² Architectural Framework based on the example of the Context Level of the Model Framework

4.2 Relation between the mecPro² Process Framework and the mecPro² Architectural Framework

The integration of the Process Framework and the Architectural Framework is realised by the people in the development process, more precisely via their process roles. These process roles execute the activities in the development process by using or creating information from the architectural framework. Through their acquired process role, they have a defined perspective (viewpoint) on the system model. Depending on the process activity, they require defined views on the Architectural Framework, more specifically on the ontology. Figure 5 shows the relationship between the terms used in the Process Framework and the Architectural Framework. There is a defined viewpoint for each process role. Corresponding views conform to the rules of the viewpoints and represent an ontology section.

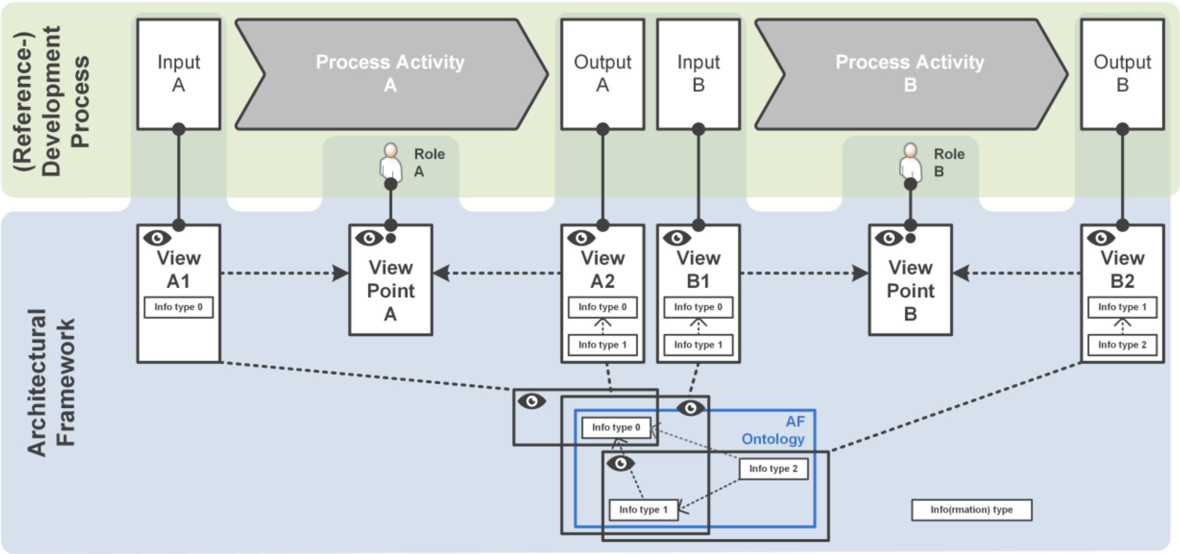

Figure 5. Relation between the mecPro² Process Framework and the mecPro² Architectural Framework

If a system has been described based on the Model Framework of the Architectural Framework once, a system breakdown results. The information transfer between the levels of the system breakdown takes place via the layers of collaboration. As seen in Figure 6, the solution space of the system of interest and the requirement/ problem space of the next lower level overlap.

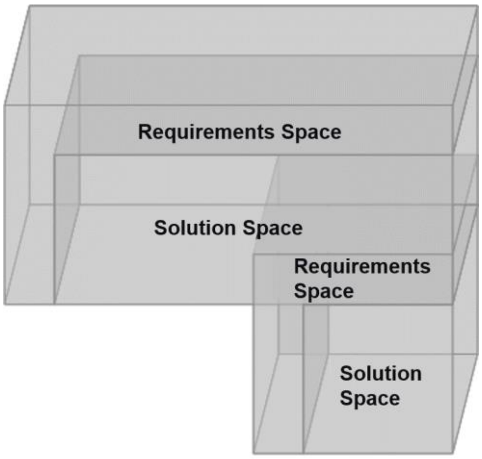

Figure 6. Relation between requirements space and solution space

While the outer level of the Model Framework (Context Level and Technical Solution Level) are used for the formal and model-based description of the requirements, the inner level serve the development of a solution. The information of the Technical Solution Level represents the sum of information of the Context Levels of all the solution elements (respectively their requirements). So, this means that the solution of one system breakdown level is the context for the next lower level. In other words, the solution elements of one breakdown level are the systems of interest of the following system breakdown level.

At this point, it can also be seen that a continuity in a model-based development process along the system breakdown to a discipline-specific partitioning can only be realized if the system model itself is used as requirements specification. In the case of a translation of the model-based information of this level back to natural language requirements, it would inevitably lead to a media break. This would hinder the continuity of the model-based development process. At this point the questions arise: how the transition from system development to discipline-specific development can be accomplished and how is it possible to describe discipline-specific solutions in a model-based way. Model-based approaches in these areas are scarcely present and still in the research phase like the FAS4M approach for mechanical engineering (Grundel et al., 2014), the INCONIX approach for electronic engineering (Rosenberg and Mancarella,

2010) or the SPES 2020 Methodology for embedded software systems (Pohl et al., 2012; Pohl et al., 2016).
Therefore, a pragmatic approach is to propose the Architectural Framework also for the discipline-specific development. Due to linking the disciplines to the system development through the collaboration layers, the levels of solution finding are encapsulated and therefore can be changed in the future without any fundamental adaptations to the Architectural Framework. The entire concept of collaboration is shown in Figure 7.

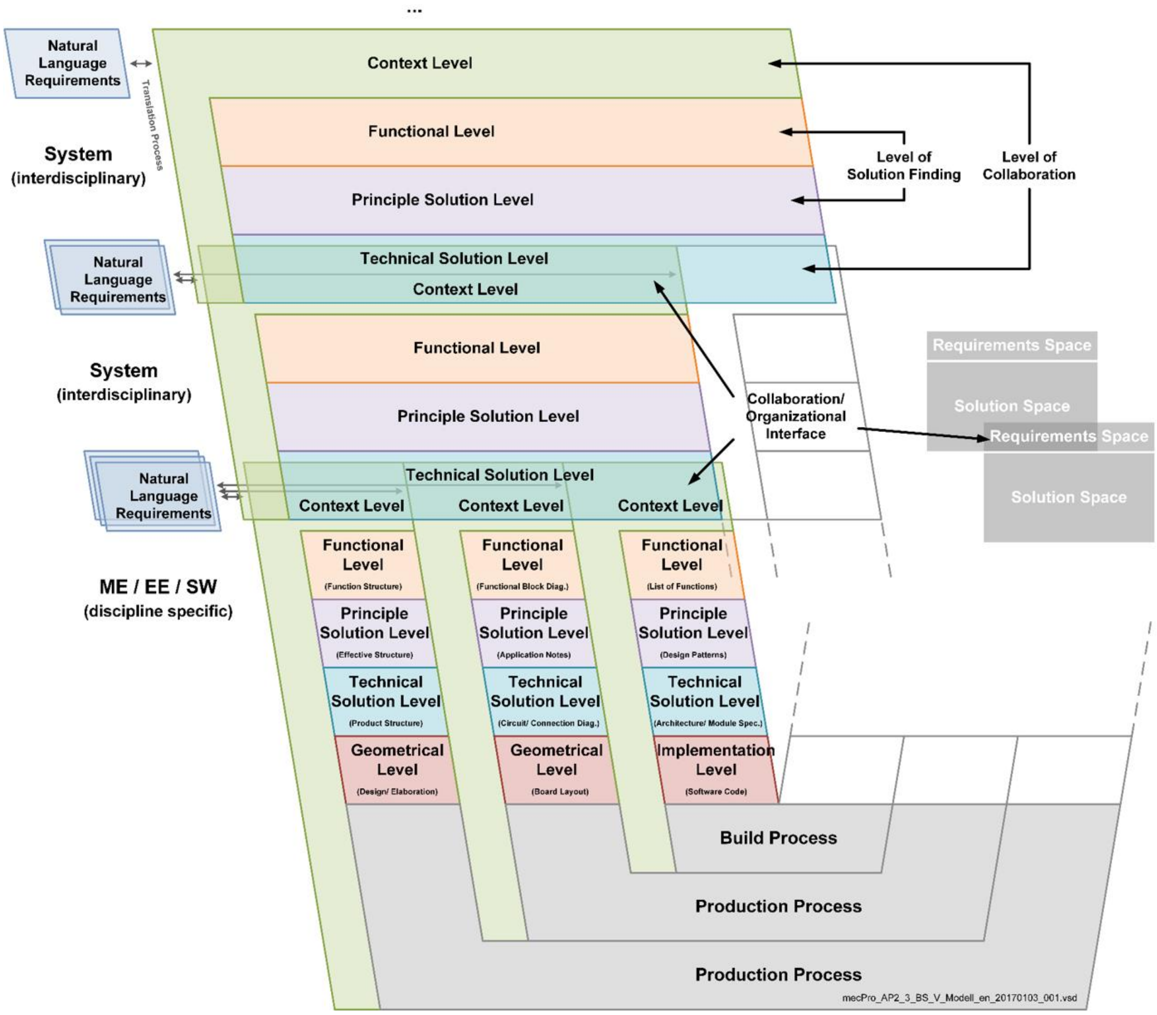

Figure 7. Concept of collaboration during the development process (Eigner et al., 2015)

5 CONCLUSION AND OUTLOOK

This article provides a consistent and integrated approach for developing and managing system lifecycle information based on SysML models. Two concepts work together as an integrated approach: on the one hand the mecPro² Process Framework provides a tailorable process for system development; on the other hand the mecPro² Architectural Framework provides an approach for describing a system using SysML. Compared to existing concepts, one essential improvement of this integrated and model-based approach is that requirements and system information will be highly formalized based on the conversion from the natural language into a model-based one. On the one hand, this allows a smooth transition from the requirements to the solution finding process and on the other hand, it improves the collaboration and consistency from the specification of a system to the specification of its components on system level as well as on discipline-specific level.
This holistic concept for the model-based development of cybertronic systems was developed together with OEMs, suppliers and system and consulting companies from the German automotive industry and has been verified and validated using two automotive-specific application scenarios. Furthermore, implementations by PLM software vendors have shown that model management with PLM and SysLM

solutions can be implemented (Kirsch et al., 2016). Future work will concentrate on iteratively improving this approach with feedback of practitioners. For example collaboration and handover to discipline-specific models need deeper examination. Other work will focus on managing variability (Schulte et al., 2016a) and configuration (Schulte et al., 2016b) in a model-based manner and consider how to integrate these concepts with the approach described in this article.

REFERENCES

Cadet, M., Meissner, H., Hornberg, O., Schulte, T., Stephan, N., Schindler, C. and Aurich, J.C. (2015), "Modellbasierter Entwicklungsprozess cybertronischer Produkte und Produktionssysteme - Grundlagen, erste Ansätze und weiteres Vorgehen", Stuttgarter Symposium für Produktentwicklung 2015, Stuttgart, Germany, June 18-19, 2015.

Eigner, M., Dickopf, T., Apostolov H., Schäfer P., Faißt K.G. and Keßler A. (2014), "System Lifecycle Management - Initial Approach for a Sustainable Product Development Process Based on Methods of Model Based Systems Engineering", 11th IFIP WG 5.1 International Conference - PLM 2014, Yokohama, Japan, July 7-9, 2014, Springer, Heidelberg, pp. 287-300.

Eigner, M., Dickopf, T. and Huwig, C. (2016), "An Interdisciplinary Model-Based Design Approach for Developing Cybertronic Systems", Proceedings of the 14th International Design Conference DESIGN 2016, Cavtat, Dubrovnik, Croatia, May 16-19, 2016, University of Zagreb, Zagreb, pp. 1647-1656.

Eigner, M., Dickopf, T., Schulte, T. and Schneider, M. (2015), "mecPro² - Entwurf einer Beschreibungssystematik zur Entwicklung cybertronischer Systeme mit SysML", Tag des Systems Engineering 2015, Ulm, Germany, November 11-13, 2015, Hanser, Munich, pp. 163-172.

Eigner, M., Gilz, T. and Zafirov, R. (2012), "Proposal for Functional Product Description as Part of PLM Solution in Interdisciplinary Product Development" Proceedings of 12th International Design Conference DESIGN 2012, Dubrovnik, Croatia, May 21-24, 2012, University of Zagreb, Zagreb, pp. 1667-1677.

Grundel, M., Abulawi, J., Moeser, G., Weilkiens, T., Scheithauer, A., Kleiner, S., Kramer, C., Neubert, M., Kümpel, S. and Albers, A. (2014), "FAS4M - No more: "Please mind the gap!"", Tag des Systems Engineering 2014, Bremen, Germany, November 12-14, 2014, Hanser, Munich, pp. 65-74.

Holt, J. and Perry S. (2013), SysML for Systems Engineering - 2nd Edition: A model-based approach, The Institution of Engineering and Technology, London.

Kirsch, L., Müller, P., Eigner, M. and Muggeo, C. (2016), "SysML-Modellverwaltung im PDM/PLM-Umfeld", Tag des Systems Engineering 2016, Herzogenaurach, Germany, October 25-27, 2016, Hanser, Munich, pp. 333-342.

Pohl, K., Broy, M., Daembkes, H. and Hönninger, H. (2016), Advanced Model-based Engineering of Embedded Systems - Extensions of the SPES 2020 Methodology, Springer, Berlin.

Pohl, K., Hönninger, H., Achatz, R. and Broy, M. (2012), Model-based Engineering of Embedded Systems - The SPES 2020 Methodology, Springer, Berlin.

Ponn, J. and Lindemann, U. (2011), Konzeptentwicklung und Gestaltung technischer Produkte - Systematisch von Anforderungen zu Konzepten und Gestaltlösungen, Springer, Berlin, Heidelberg.

Rosenberg, D. and Mancarella, S. (2010), Embedded Systems Development using SysML. Sparx Systems Pty Ltd and ICONIX. Available at: http://www.sparxsystems.com.au/downloads/ebooks/Embedded_Systems__Developmentusing_SysML.pdf (January 02, 2017).

Schulte, T., Schneider, M., Dickopf, T. and Mayerhofer, L. (2016a), "Erweiterung des integrierten Konzepts aus Prozessrahmenwerk und Beschreibungssystematik von mecPro² um ein modellbasiertes Variantenmanagement", Tag des Systems Engineering 2016, Herzogenaurach, Germany, October 25-27, 2016, Hanser, Munich, pp. 259-268.

Schulte, T., Schneider, M., Judaschke, U. and Batz, D. (2016b), "Systemmodelle verwalten mit ConfigML - Motive, Grundlagen und erste Konzepte einer Sprache für das modellbasierte Konfigurationsmanagement", Tag des Systems Engineering 2016, Herzogenaurach, Germany, October 25-27, 2016, Hanser, Munich, pp. 323-332.

VDI 2221:1193-05 (1993), Systematic approach to the development and design of technical systems and products, Beuth, Berlin.

Weilkiens, T., Lamm, J.G., Roth, S. and Walker, M. (2016), Model-based System Architecture, Wiley, USA.

ACKNOWLEDGMENTS

The authors would like to acknowledge funding from the German Federal Ministry of Education and Research (BMBF) attended by the Project Execution Organization Karlsruhe (PTKA). The authors would also like to thank all project members for their contribution and research on the topic of Cybertronic Systems and for their additional input and assistance. Further acknowledgments and information about the research project mecPro² can be found at: http://www.mecpro.de/static/kt1_de.