INTERDISCIPLINARY SYSTEMS MODELING USING THE CONTACT & CHANNEL-MODEL FOR SYSML

Albers, Albert [1], Zingel, Christian [1]
(1) IPEK - Institute of Product Engineering at Karlsruhe Institute of Technology (KIT)

ABSTRACT
One of the most challenging aspects in product engineering is clear communication between diversified areas of expertise. Many terms are used in varying comprehension, which leads to different understandings of problems and hence to gaps in the information transfer. The reason for this issue is a missing adequate overall language for product engineers and hence a lack of capable tools for imaging comprehensive system understanding and knowledge.
This paper introduces an approach to perform modeling of technical systems using the standardized Systems Modeling Language (SysML) in combination with the Contact & Channel approach (C&C-A). The intention is to combine strengths of SysML in visual modeling of complex systems with capabilities of C&C-A in integrated modeling of system structures in the corresponding functional context. The objective is to provide an easily manageable software-based language for interdisciplinary systems modeling in order to improve domain-crossing communication and teamwork.
For this purpose, SysML and C&C-A are initially introduced. Afterwards, the C&C-A metamodel for SysML is elucidated. An application example is used to demonstrate the employment of this modeling approach. Concluding, the current value of this approach for systems engineers is reflected and an outlook towards future research tasks is given.

Keywords: Model-Based Systems Engineering, Systems Modeling Language, SysML, Product Engineering, Product Development Process, Contact and Channel Approach, C&C-A, Contact and Channel Model, C&C-M

1. COMMUNICATION AS A CHALLENGE IN PRODUCT ENGINEERING
Most of today’s technical products are highly complex systems. On the one hand, growing global competition forces companies to accelerate technical progress. Concurrently, emerging development costs must be kept low. On the other hand, customer requirements are permanently rising. New innovations are taken for granted in a little while and further features have to be provided in technical products to awaken customer’s interests. During the last decades, computer science has advanced very rapidly. This also has influenced former purely mechanical products through a severe electrification. The emerging mechatronic systems provide a multiplicity of new functions. Today, about 90 percent of all innovations are electronic- or software-driven, which can be observed impressively in the development of the automobile branch in recent years [1].
But also research and development in the other engineering disciplines has advanced. Hence, an increasing number of specialized disciplines is emerging. Some examples are fluid dynamics, software design, kinematics, electric/electronic (E/E) engineering or measurement and control. All these specialists use their own technical expressions, and many terms are multi-allocated with deviant meanings. The main contemporary challenge is to find domain-spanning definitions for technical terminology in order to enable or improve communication between product engineers. Sticking points are reduction of complexity without forfeiting the clearness, what is a balancing act between simplicity and integration. An intuitive communication assumes a limited scale of vocabulary and ideally visualization facilities. Clear interfaces between disciplines also have to be defined in order to clarify competences and to provide all needed information.
These issues are still not solved what leads to a missing software support providing such a unified technical language. Unlike to this challenge for interdisciplinary systems modeling, computer scientists found an object-oriented solution for software systems: the Unified Modeling Language (UML™ [2], [3]). This approach was adapted on mechatronic systems by the Systems Modeling

Language (SysML™ [4] - [6]). However, the SysML metamodel is set up in a very generic manner. On the one hand, it can be applied for interdisciplinary systems modeling by providing elements for modeling different aspects like requirements, structure and behavior of systems [6]. On the other hand, the wide-ranged application in daily industrial engineering work has not taken place yet [7]. A possible reason is that a specialized engineer is not able to communicate relevant information through such a generic system model. The provided modeling elements are still very software-oriented and hard to understand by mechanical engineers. But the SysML is an adaptable language, which enables users to build up their own language profile. This feature of SysML is applied in this paper, which is elucidated after a short introduction of the theory of model-based systems engineering and the C&C-A. An application example demonstrates the intuitive employment of this new modeling approach.

2. MODEL-BASED SYSTEMS ENGINEERING

According to Stachowiak [8], a model is characterized by three main features. The *mapping attribute* means, that a model is a representation of natural or artificial originals (which can also be a model). Attributes of the original are mapped to the model. The *reduction attribute* states that models never contain all attributes of the represented original, but only the relevant from a modeler's or a model user's viewpoint. The *pragmatical attribute* signifies that a model always has a pragmatic purpose of usage. This purpose is for instance defined by time, intention or users. Hence, a model is always interpreted.

According to INCOSE[1], systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements and then proceeding with design synthesis and system validation while considering the complete problem. Hitchins [9] describes Systems Engineering as the art and science of creating whole solutions to complex problems.

Model-based systems engineering (MBSE) applies systems modeling as part of the systems engineering process to support analysis, specification, design and verification of the system being developed [10].

At the IPEK – Institute of Product Engineering at the Karlsruhe Institute of Technology (KIT), a meta-model of product engineering, called *integrated Product engineering Model* (iPeM) has been developed [11]. It divides the system of product engineering into the system of objectives, which is transferred into the system of objects by the operation system. Hence, the only active system is the operation system. It transforms information to objectives and transforms those again to objects by utilization of resources. The operation system is divided into further sub-systems. These are the activity matrix, a system of resources and a phase model, which finally visualizes the individual process (Figure 1). The System of Objectives and the System of Objects are exemplary depicted in Figure 1 using SysML diagrams. This illustration stands for the long-term aim of the authors to apply an interdisciplinary Modeling Language as networking tool in the product engineering process.

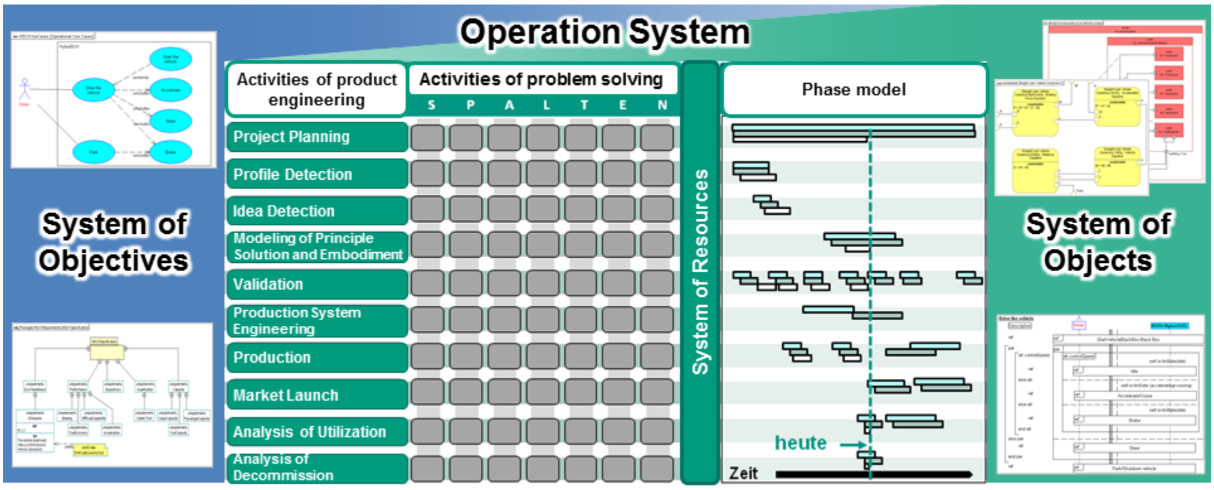

Figure 1: The integrated Product engineering Model (iPeM)

[1] International Council on Systems Engineering, www.incose.org

SysML is a general-purpose graphical modeling language for systems engineering applications, developed by the OMG[2] SysML Revision Task Force (RTF). SysML supports the specification, analysis, design, verification, and validation activities in the product development process of a broad range of complex systems. These systems may include hardware, software, information, processes, personnel, and facilities. SysML reuses a subset of UML 2 and provides additional model elements to satisfy the systems engineering focus of the language [6]. The intention of the OMG was to establish SysML as the official language of Systems Engineers, which has not sufficiently happened yet [7]. Because of its very widespread scope, the SysML relies on a very generic metamodel and has strengths in visualization of complex systems of any kind. This metamodel is built up by the so-called Meta Object Facility (MOF), a standardized metadata management framework [12], also developed by the OMG.

This means that – adequate MOF-editor provided – the SysML can be adapted for more specific purposes, what is done in the approach at hand. A realized example of such a specialization is ModelicaML, where exactly this has been applied for adapting UML to formal executable modeling according to Modelica [13]. The aim in this paper is the provision of a formal, unified language, which is easy to apply by product engineers of all disciplines within a product development process.

The solution statement is an extension of the SysML metamodel by elements for modeling mechatronic systems using the *Contact & Channel-Approach* (C&C-A) [14]. This is characterized by the integrated modeling of a systems form and function, including the according processes and interfaces. This work introduces the first realization step, an integrated software-based modeling of system structures in their functional context. For a better understanding, the C&C-A for modeling technical systems is introduced in the next chapter.

3. THE CONTACT AND CHANNEL APPROACH

The Contact & Channel-Approach (C&C-A) is used for building Contact & Channel-Models (C&C-M) of technical systems. The C&C-A provides generic elements to deduce specific models. This approach is developed at the IPEK in order to put abstract function structures across on the form of technical systems [14], [15].

The C&C-A for modeling technical systems is based on three basic hypotheses. It defines the realization of a function through at least two Working Surface Pairs (WSP) and their connecting Channel and Support Structures (CSS). WSP are pair-wise interfaces between systems. This can be solid surfaces of bodies or boundaries with surfaces of liquids, gases or fields which are in permanent or occasional contact with the Working Surface. They take part in the exchange of energy, material and information within the technical system. CSS are volume elements (solids, liquids, gases or spaces containing fields), which connect at least two Working Surface Pairs. These elements can either transfer the system variables material, energy or information between WSP or store them (i. e. mass inertia). WSP and CSS are applicable in the level of abstract function description or in a concrete level of system component description. This enables the visual description of impacts and the mapping to corresponding locations in the system [15]. Further C&C-M elements are the Limiting Surface (LS) and the Remaining Structure (RS), which are used to describe system elements that are not involved in a currently regarded function [16]. An application example, a C&C-Model analysis of the process "writing with a ball pen", is shown in Figure 2. Relevant WSP and CSS are depicted in a lower level of detail (left side) and a more detailed section of the pen's ballpoint (right side). The Connectors stand for the connection of the system under consideration with its environment. They contain and describe all relevant information (influencing parameters, border conditions and their networking) of Working Surfaces at the system boundary. This element becomes necessary to adequately describe the system in its functional context. Hence, a connector is a reduced description of the system environment [15].

[2] Object Management Group. www.omg.org

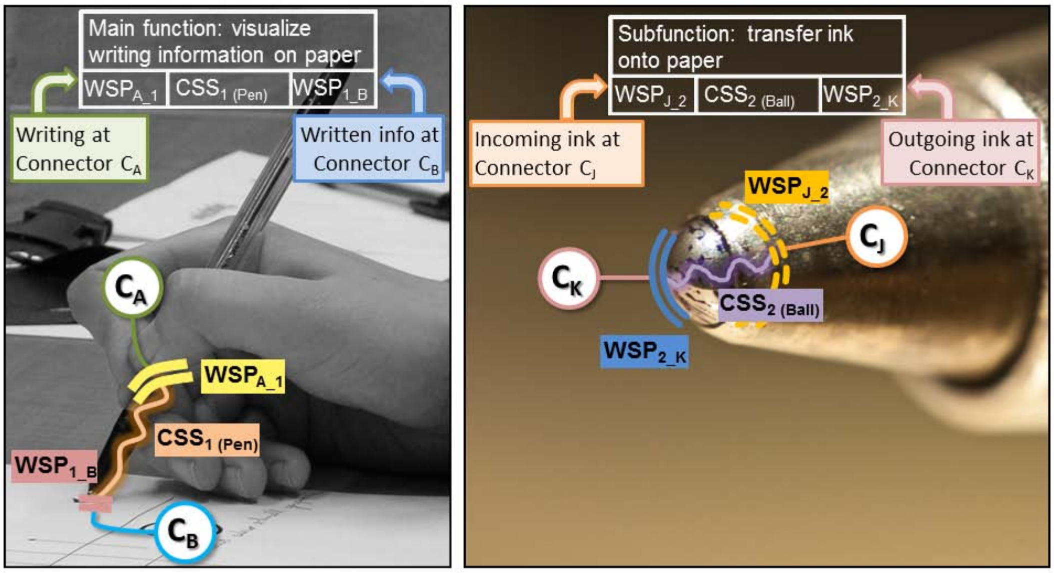

Figure 2: C&C-A description of the process of writing with a ball pen [17]

C&C-A analysis can be performed on every kind of technical system and every level of detail. Properties of the WSP and CSS can be used to explain intended or unmeant effects and resulting functions. For instance, a more detailed consideration of the WSP_{1_B} between the pen and the writing surface can help to obtain in the given example, why the ball does not write on glass. Responsible effect in this case could be an insufficient friction in order to turn the ball, or the properties of the liquid ink prevent a wetting of glass through ink [17].

Schyr [18] extended the C&C-A by merging its model elements with the simulation language Modelica. The basic concept is to mathematically define physical properties within Modelica. In this context, Schyr enhances Working Surfaces of the C&C-M by properties of connectors from Modelica. The physical properties of CSS are modeled and described by equations. He applies this approach in validating drive trains.

Enkler [16], [19] presented a generalized C&C-A based on the application for the combination of several software tools to visualize interfaces between the applied tools in simulation based product development processes. His aim was the adequate consideration of multi-domain effects in the virtual product development. The first step in using simulation tools is the definition of the corresponding target system. Secondly, adequate simulation methods and processes are selected, supported by C&C-A as linking language between target system, operating system and object system. In this generalized approach, the target system contains the desired results of the simulation, the object system contains the adequate simulation model and the system of objects contains the results of the simulation. Generalized Working Surface Pairs now can also be applied to describe interfaces between software tools. Channel and Support Structures are replenished by the tools themselves (cf. Figure 3). The generalized C&C-A enables a product engineer to design customized simulation processes. Thus target-oriented processes are achieved by modularization of software tools and a linking language.

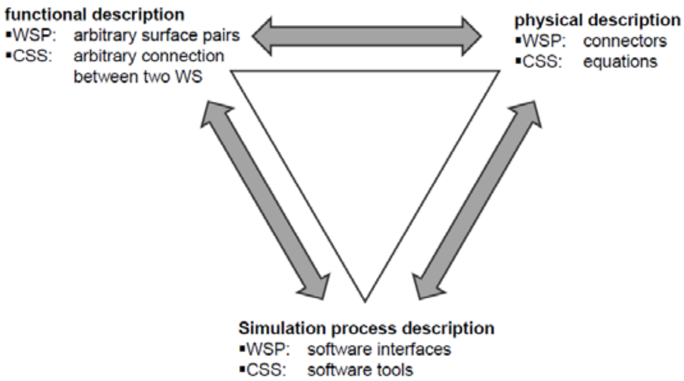

Figure 3: Scheme of the generalized C&C-A [16]

Enkler also realized a first implementation of the C&C-A in a software tool, the ConChaCoach C³ (Contact & Channel Coach). This tool provides a base for the usage of C&C-A in complex and real world applications. The generalized approach was exemplified at a deformation analysis of a micromechanical part made of anisotropic material and a kinematic study of a humanoid robot [16].
The idea of a software implementation of the C&C-A was advanced in a cooperation of the IPEK with the Cambridge Engineering Design Centre (at the University of Cambridge) [20]. That approach aimed to support the design of system architecture, including interrelated geometrical and functional elements. It has been implemented in a software tool in order to facilitate modeling of complex products. The software tool uses the CAM (Cambridge Advanced Modeler) platform, which is developed at the University of Cambridge [21].
Both software implementations are based on specific meta-models of the C&C-A, created for the applied software-frameworks. This preliminary work builds the scientific basis for this approach at hand. The essential difference between the preliminary work and this approach is that the meta-model is not applied in a proprietary developed framework but extends a standardized and established modeling language, based on and supported by a worldwide community[3]. Thus, the preconditions are different: the metamodel is developed by applying the MOF as a standardized meta-modeling language in order to extend the generic SysML. This is particularly done through allocating C&C-M elements to existing SysML elements through extending their properties. The main advantage in contrast to the prior approaches is a full compatibility of this metamodel with SysML, UML and other existing modeling standards. Thus all existing (commercial an open-source-based) frameworks, software interfaces and also interface protocols like the STEP[4] format are supported, which facilitates the integration of such a modeling tool into an existent tool environment.

4. CONTACT & CHANNEL - METAMODEL FOR SYSML

The aim of this work is the combination of the UML/SysML's strength in graphical and descriptive modeling with the C&C-A's capabilities in multi-level system analysis and synthesis, combined with the methodical linking on product development processes. Object-oriented modeling like in UML and SysML is well-established, that is why the most important invention is the implementation of the C&C-A in a metamodel extending SysML with according modeling methodology. Furthermore, a commercial SysML software platform is used. For this paper, Artisan Studio V7.2 was applied.
The extension of the SysML metamodel in the applied software is realized by creating *stereotypes* for new model elements and *tag definitions* for new attributes. The technical implementation, which is not addressed in detail here, is done by visual basic scripts.
The first step is an assignment of existing elements in SysML to Contact & Channel-Model (C&C-M) elements. This is not done by direct one to one assignment but by transfer of the logical meaning within the respective metamodels. SysML *Blocks* are modular units of system description [6]. In the context of a technical system, it stands for a form-afflicted physical or a software unit. A CSS's final form is initially unknown. If a modular physical unit will be used to fulfill only one function, it consists exclusively of one[5] according CSS and a Remaining Structure (RS). This RS is accounted by manufacturing or design reasons and does not fulfill any function. Vice versa no system function is affected by the RS. Removing it would at most cause additional costs. All CSS and RS of a modular unit together build up an Entire Structure, which consequentially equates a Block in SysML.
A block for itself can never fulfill any function, because it does not yet interact with its environment. It additionally needs Working Surfaces (WS), which can – paired with WS of other blocks to WSP – realize effects. Furthermore, at least two WSP and a connecting CSS have to be built per dedicated function to realize[6]. WS are modeled in SysML through using the so-called *Flow Port*. This is specified as an interaction point through which information, material or energy can enter or leave the owning block [6]. Hence, one Flow Port exactly has the same meaning as one WS. The visualized assignments of WS and CSS in C&C-A and the corresponding SysML-model are depicted in Figure 4.

[3] The Object Management Group. www.omg.org
[4] ISO 10303 Standard, particularly AP233 (application protocol 233)
[5] According to revised basic hypothesis II, cf. [15]
[6] According to revised basic hypothesis I and II, cf. [15]

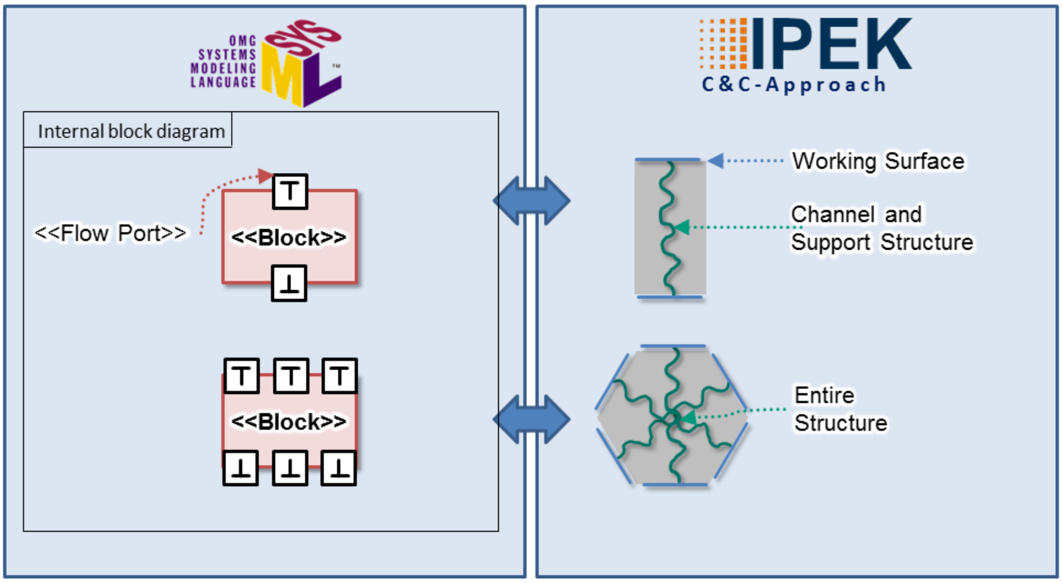

Figure 4: C&C-M elements for SysML (1)

Flow Ports are in SysML connected to one or more other Flow Ports via *Connectors* [6]. These are applied in *Internal Block Diagrams* to build up WSP in SysML. Though, C&C-A defines a WSP as a pair of exactly two WS. This means for SysML, that Flow Port connections should be restricted to only occur one-to-one. One the one hand, Flow Ports can be defined as "atomic", what means that their interaction consists of one physical or logical parameter. Otherwise, a "nonatomic" Flow Port can also contain a so-called *Flow Specification* which lists multiple items that can flow. This means for the C&C-M elements in SysML, WSP have to be built up through connecting atomic Flow Ports via Connectors. The different possible flows (information, energy and material) are visualized by the colors red, blue and yellow for easier differentiation (Figure 5).
The information about the items which can flow between CSS is deposited as *Flow Property* with appropriate *unit* and *dimension* in the Flow Port's properties. Analogical, attributes of CSS can be deposited in Block's properties (for instance mass, inertia, material strength etc.). SysML provides a constraint diagram, which allows modeling of physical relationships between these attributes. Usually these equation systems are better modeled in specialized tools like Maple, Matlab/Simulink or Modelica, for which reason constraints are not considered in detail here. A methodical approach using C&C-M for SysML for modeling equation systems with according software interfaces to multi-body simulation tools is part of current researches at IPEK. The results of Schyr [18] and Enkler [19] are picked up and advanced here in particular.

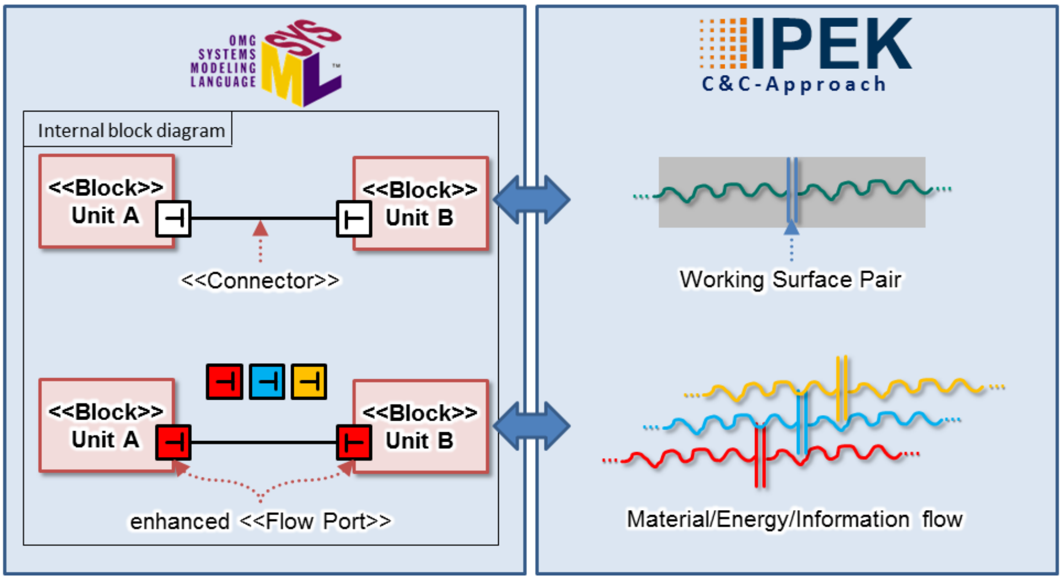

Figure 5: C&C-M elements for SysML (2)

Using the previously introduced elements enables a systems engineer to model system structures. Though, a technical system is always developed for providing specific functions. Hence, this aspect has to be modeled as well. This is done in *Activity Diagrams*, using *activities* with input- and output *PIN*'s for depicting input values, their processing and the corresponding output values (cf. Figure 6). Activities can be put into a logical order by dashed arrows (so-called *Control Flows*), PIN's are connected by solid arrows (the *Object Flows*). Both flow elements are not depicted here, but will be applied in the application example for modeling functional progressions. Finally, the reference to according structural elements is modeled using *Activity Partitions* (or *Swim Lanes*) by mapping them to Block's.

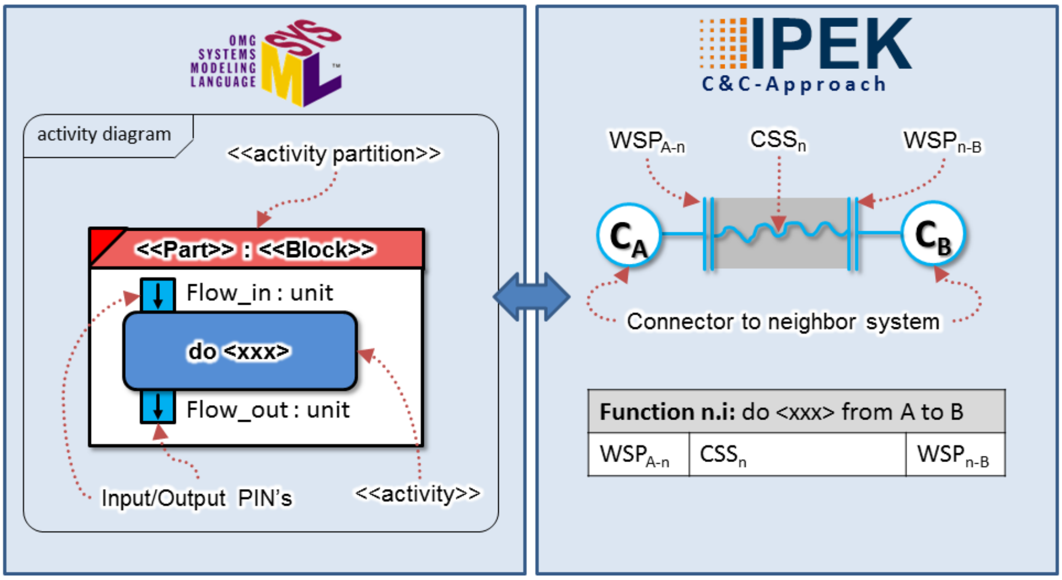

Figure 6: C&C-M elements for SysML (3)

The C&C-A describes functions using a table which contains a textual description of the function, the affected WSP's and the connecting CSS (cf. Figure 6). When one or more WSP's interact with neighbor systems which are not considered in detail, they are linked to *Connectors*[7], which contain function-relevant parameters and border conditions from the system's environment (Cf. [15]).
Summing up, the C&C-M for SysML enables a systems engineer to model technical systems in different levels of detail using a software tool. Especially the weakness of SysML in description of mechanical effects and functions in relation to their corresponding form is eliminated. This is demonstrated using predominantly mechanical systems in the following application examples.

5. APPLICATION EXAMPLE

The software implementation of the introduced C&C-M for SysML is demonstrated in this chapter using a simplified vehicle assembly. Real vehicles are outstanding examples for highly complex mechatronic systems. Manifold subsystems interact among one another but also with the environment and the human as well. About up to 100 control units are implemented in modern upper-class vehicles and realize widespread functions, beginning with injection regulation up to manifold driving assistance systems [22]. Hence, no comprehensive example can be given here for modeling such an integrated, complex technical system. Nevertheless, a simplified assembly of some subsystems of a typical vehicle shall give examples how to apply the Contact & Channel-Model for SysML.
A common modeling approach in SysML starts with the system environment. Afterwards, the system of objectives is modeled using requirements and use-cases, but also activity- and sequence-diagrams for demanded system behavior. When starting structure modeling, the system units are assembled top-down. Figure 7 shows an example for the top level of a vehicle in an Internal Block Diagram (cf. [5] for details to diagram types in SysML). Due to the intended context, not all sub-systems are shown in the diagram, but only the main drivetrain elements. For a better visual appearance, the blocks are depicted as pictures of the respective units.

[7] Note, that Connectors in C&C-A (Figure 6) are different in their meaning to Connectors in SysML (Figure 5)

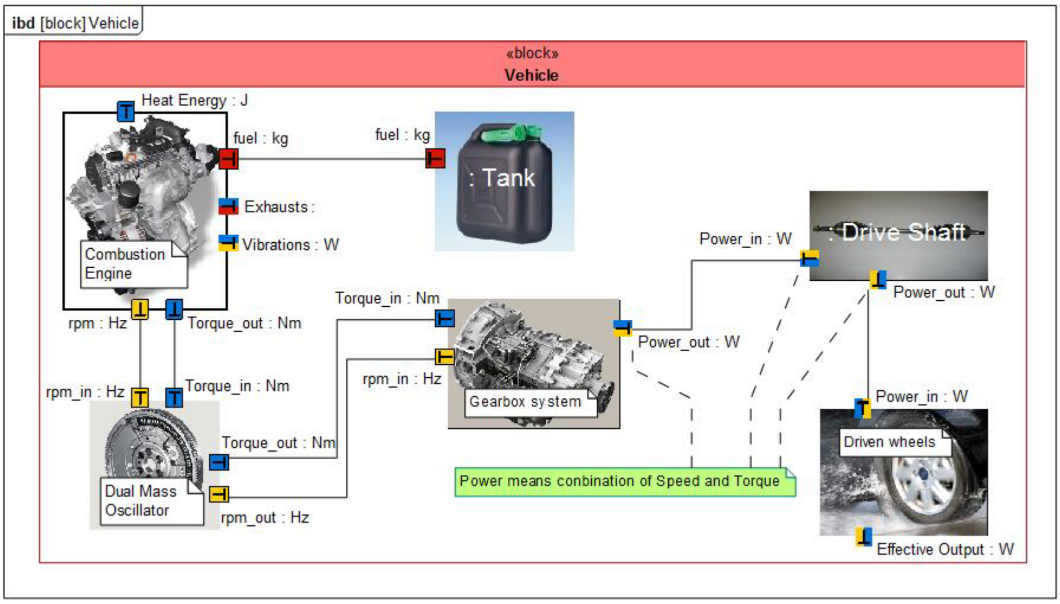

Figure 7: Internal Block Diagram of the exemplified vehicle assembly

The shown WS and WSP are just an extract of occurring flows in reality. All the depicted elements are stored in the SysML model and can be reused in other diagrams again arbitrarily. All elements and their properties are unambiguous. When a block is assigned in an Internal Block Diagram, this unit becomes part of the system, which internal structure is being modeled in this diagram. In that case, a *Part* is created. In other words, a Part is an instance of a Block in a defined role. Hence, it inherits the Block's properties, but also can gain concrete parameter values. This is why the same Block can have multiple identical parts. For example, a car has 4 wheels, but only two of them are driven. These are shown in the diagram above (multiplicity is blanked out). The other two wheels just differ in their role (they are not driven) through differing connection of their Working Surfaces to WSP, but they still have identical properties (i.e. material, size).
A more detailed level of a technical subsystem is shown in Figure 8. A predominantly mechanical example is used to demonstrate the negotiation of the challenge of modeling geometrical information in a functional context. A simplified input shaft of a gearbox is shown in a sectional representation and analyzed using the C&C-A. Again, not all WS, WSP and CSS are illustrated for clarity reasons. Table 1 describes two according functions.

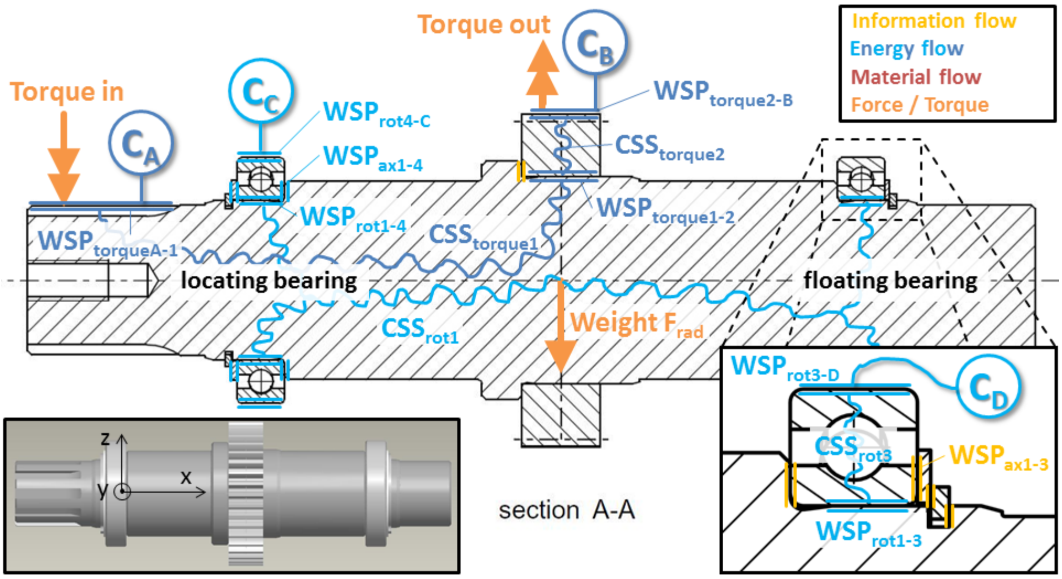

Figure 8: C&C-A analysis of a simplified gearbox input shaft

Function 1: Transmit drive torque from spline shaft to gearwheel carrier			**Function 2:** Brace appearing radial forces and allow rotation of shaft around x-axis		
$WSP_{torqueA-1}$	$CSS_{torque1}$ (shaft)	$WSP_{torque1-2}$	WSP_{rot1-3}	CSS_{rot3} (shaft)	WSP_{rot3-D}

Table 1: Abridged functions of gearbox input shaft

The given assembly is stressed by an input torque on the spline shaft, which is carried via $CSS_{torque1}$ to the gearwheel. The shaft bearing arrangement is locating-floating, at which the locating bearing is on the left side, where axial loads can be transmitted via WSP_{ax1-4} (blue colored for energy). By contrast, WSP_{ax1-3} at the loose bearing is yellow-colored, because it only transmits information about the bearing arrester for mounting the assembly correctly. Both bearings transfer the assembly's weight (Force "Frad" in Figure 8) via WSP_{rot1-3} and WSP_{rot1-4}. Hence the shaft's degrees of freedom are restricted to rotation around the center-axle (x-axis).

The same assembly is modeled using C&C-M for SysML (Figure 9). The assembly is simplified in comparison to the engineering drawing, so are circlips and washers omitted.

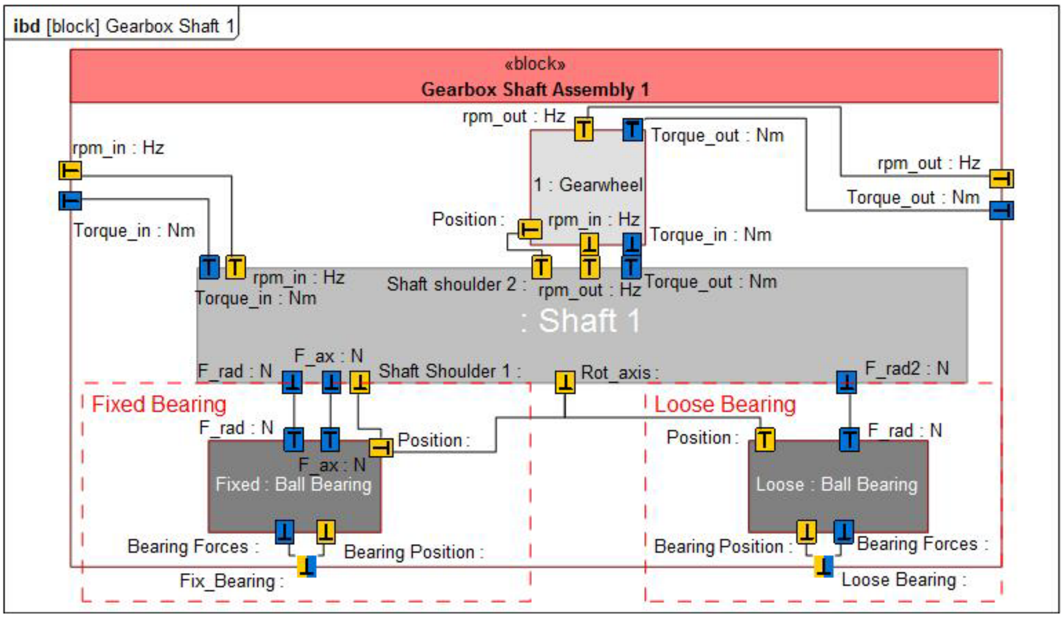

Figure 9: Gearbox input shaft modeled in C&C-M for SysML

In such a detailed level, manifold information about the parts and sub-assemblies in this assembly can be depicted. Loads and friction coefficients can also be specified for surfaces, which are represented through the Flow Ports as well as geometrical dimensions of parts (the comprehensive structures).

The functional structure of a system is modeled in activity diagrams in SysML. In case of modeling an existing system like in the actual application example, the structure of a system with according information flow already exists. In that case, not only activities can be modeled, but also the mapping to the structural elements is also possible. This is done for the entire vehicle function "provide driving force for vehicle movement" in Figure 10.

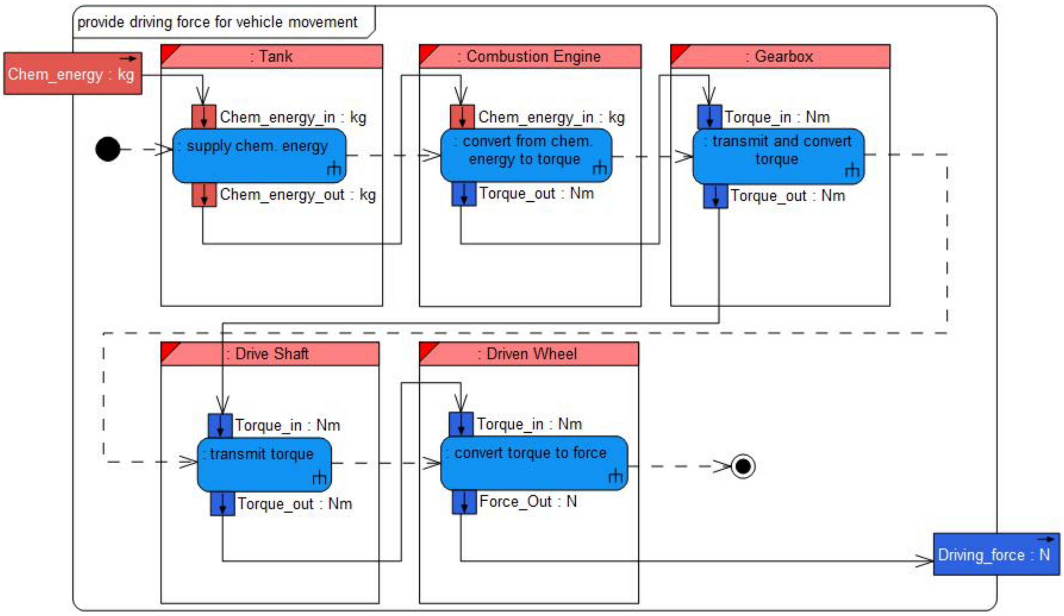

Figure 10: Function decomposition "provide driving force for vehicle movement" in a SysML-activity diagram

The input PIN contains the refueling access to the vehicle tank. The material flow for the fuel, called "Chem_energy", flows into the tank, which provides the combustion engine with fuel supply. This chemical energy is therein converted into mechanical rotational energy, the torque, which is furthermore transmitted and converted (transformed) through the gearbox and the drive shaft to the wheels, which convert the torque to a driving force.

Going into a more detailed level of the gearbox' sub-function "transmit and convert torque", the depicted functional structure in Figure 11 results.

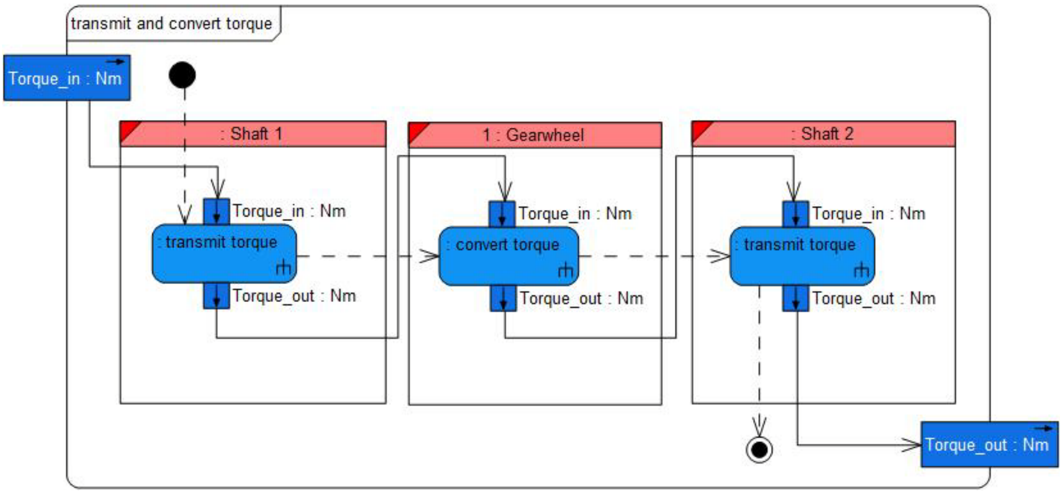

Figure 11: Function decomposition "transmit and convert torque"

This simplified decomposition transmits the incoming torque from PIN "Torque_in" through shaft 1, which is then converted by the transmission ratio of the gearwheel pair. Afterwards, the transmitted torque leaves the gearbox via shaft 2 at the PIN "Torque_out". SysML allows to reuse activities like "transmit torque", assumed all properties, the transfer function and input/output PIN's are identical. For that purpose, the consolidated function basis by Hirtz et. Al. has been partially implemented [23].

6. CONCLUSION

Concluding, the examples show that the introduced approach "C&C-M for SysML" enables systems engineers to combine the advantages of C&C-A in system analysis and synthesis with the strengths of SysML in graphical and descriptive modeling in a software framework. A predominantly mechanical

example is intentionally used to demonstrate the negotiation of the challenge in modeling geometrical information in a functional context.
Thus, an important step is carried out towards providing a software-based approach which aims to support engineers throughout the whole product development process. This target shall be gained through modeling mechatronic systems in a formal and consistent manner and hence improving interdisciplinary communication and teamwork. The way of implementation of the C&C-A in a MOF-metamodel preserves the applicability of well-known modeling tools. Furthermore, existing and emerging software interfaces for UML and SysML are intended to be integrated later for the exchange of information between the interdisciplinary C&C-M for SysML-model and discipline-specific tools like multi-body-simulation tools, CAE or CAD-tools. Particularly the ISO 10303 Application Protocol 233 for Systems Engineering Data exchange, which is currently under way and which extends the popular STEP-family, aspires to provide a standardized and well-defined information management. SysML is already fully compatible to this upcoming standard [24].
Currently, some limitations are still given. Especially the commercial tools restrict the access on the metamodel, so that some intended model elements, properties or connections could yet not be included completely. Furthermore, SysML still has to be adapted more extensively to avoid modeling errors by the model user before the presented approach is integrated with a satisfying degree of maturity.

7. OUTLOOK ON FUTURE RESEARCH

In close cooperation with the other research works at IPEK (cf. [20]), C&C-A will be advanced towards becoming a technique for integrated, interdisciplinary model-based Systems Engineering. Especially a consistent modeling of relationships between use-cases and requirements in the system of objectives and the structural and functional elements in the system of objects is part of the current researches. Furthermore, the approach is currently validated in real industrial applications. The gained feedback of these experiences will be integrated into the modeling methodology as well. Especially the uniqueness of every development process, as it is stated by Albers et. al. [11], requires a highly flexible modeling approach for branch- or company-specific claims. One essential issue is the identification of a reasonable level of detail in interdisciplinary modeling before going into domain-specific development tools. Another issue is an automatic matching of modeled subsystem functionality and properties with design catalogues in order to support engineers in solution choice.
In order to keep a maximum degree of freedom in software integration of the C&C-A, the IPEK cooperates with electric/electronic- and software-modeling tool providers for a considerably increase of compatibility, usability and intuitivity of a future C&C-A modeling tool. Furthermore, this emerging modeling-tool will be validated concerning its task of improving interdisciplinary communication, information transfer and teamwork.

REFERENCES

[1] Frischkorn, H.-G.: Keynote-Topic: *IT im Automobil-Innovationsfeld der Zukunft.* In: Proceedings of the INFORMATIK 2003, Frankfurt, Germany, 2003.

[2] Oestereich, B.: *Objektorientierte Softwareentwicklung -- Analyse und Design mit der UML 2.0*, 2004 (Oldenbourg Publishing, Munich, Vienna).

[3] Specification: *OMG Unified Modeling Language (OMG UML™), Infrastructure and Superstructure, Version 2.3*, www.omg.org, 2010 (publisher: Object Management Group).

[4] Hause, M. and Thom, F. and Moore, A.: The Systems Modeling Language - SysML, *INCOSE UK Chapter – Spring Symposium Proceedings*, 2004.

[5] Weilkiens, T.: *Systems Engineering mit SysML/UML*, 2. edition, 2008 (dpunkt, Heidelberg).

[6] Specification: *OMG Systems Modeling Language (OMG SysML™), Version 1.2*. http://www.sysml.org, 2010 (publisher: Object Management Group).

[7] Estefan, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. In: *INCOSE MBSE Initiative*, INCOSE MBSE Focus Group, California, USA, 2008.

[8] Stachowiak, H.: *Allgemeine Modelltheorie*, 1973 (Springer, Vienna).

[9] Hitchins, D.: *Systems Engineering - A 21st century Systems Methodology*, 2007 (Wiley, West Sussex).

[10] Friedenthal, S. and Moore, A. and Steiner, R.: *A Practical Guide to SysML*, 2008 (Morgan Kaufmann, Burlington).

[11] Albers, A.: Five Hypotheses about Engineering Processes and their Consequences. In: *Proceedings of the TMCE 2010*, Ancona, Italy, 2010.
[12] Specification: *OMG Meta Object Facility (OMG MOF^{TM}), Version 2.0*. http://www.omg.org, 2006 (publisher: Object Management Group).
[13] Schamai, W.; Fritzson, P.; Paredis, C.; Pop, A.: Towards Unified System Modeling and Simulation with ModelicaML: Modeling of Executable Behavior Using Graphical Notations. In: *Proceedings 7th Modelica Conference*, Como, Italy, 2009.
[14] Matthiesen, S.: Ein Beitrag zur Basisdefinition des Elementmodells ‚Wirkflächenpaare und Leitstützstrukturen' zum Zusammenhang von Funktion und Gestalt technischer Systeme. *Dissertation, Vol. 6, publisher: mkl*, A. Albers, Karlsruhe, 2002.
[15] Alink, T.: Meaning and notation of function for solving design problems with the C&C-Approach. *Dissertation, Vol. 48, publisher: IPEK*, A. Albers, Karlsruhe, 2010.
[16] Albers, A.; Enkler, H.-G.; Ottnad, J.: Managing Complex Simulation Processes – The Generalized Contact and Channel Model. In: *Int. Journal of Product Development*, 2010.
[17] Albers, A.; Matthiesen, S.; Thau, S.; Alink, T.: Support of engineering activity through C&CM – Temporal decomposition of design problems. In: *Proceedings of TMCE2008*, April 21-25 2008, Izmir, Turkey, 2008.
[18] Schyr, C.: Model-based methods for the validation phase in the product development process of mechatronic systems by example of powertrain development. *Dissertation, Vol. 22, publisher: IPEK*, A. Albers, Karlsruhe, 2006.
[19] Enkler, H.-G.: Computer aided design of components with channel and support structures subject to high scatter using high strength molded micromechanical systems as an example. *Dissertation, Vol. 44, publisher: IPEK*, A. Albers, Karlsruhe, 2010.
[20] Albers, A.; Braun, A.; Sadowski, E.; Wyatt, D. F.; Wynn, D. C.; Clarkson, P. J.: Contact and Channel Modeling using part and function libraries in a function-based design approach. In: *Proceedings of the IDETC/CIE 2010*, August 15-18, Montreal, Quebec, Canada, 2010.
[21] Wynn, D. C.; Nair, S. M. T.; Clarkson, P. J.: The P3 Platform: An approach and software system for developing diagrammatic model-based methods in design research. In: *Proceedings of the ICED'09*, Palo Alto, California, USA, 2009.
[22] Schenk, J.: Prüfplattform für mechatronisch ausgestattete Fahrzeuge in Entwicklung und Produktion. *Dissertation, Fakultät für Maschinenwesen, TH Aachen*, Aachen, 2007.
[23] Hirtz, J., Stone, R., McAdams, D., Szykman, S., Wood, K.: *A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts*. In: Research in Engineering Design, NIST Technical Note 1447, 13(2):65-82, Missouri-Rolla, USA, 2002.
[24] Valilai, O. F.; Houshmand, M.: Advantages of using SysML Compatible with ISO 10303-233 for Product Design and Development based on STEP Standard. In: *Proceedings of the WCECS*, October 20-22, San Francisco, California, USA, 2009.

CONTACT:
Prof. Dr.-Ing. Dr.h.c. Albert Albers
Karlsruhe Institute of Technology (KIT)
Institute of Product Engineering (IPEK)
Kaiserstr. 10, 76131 Karlsruhe, Germany
Tel: +49 721 608 42371
Fax: +49 721 608 46051
Email: albert.albers@kit.edu
URL: http://ipek.kit.edu

Albert Albers is head of the IPEK - Institute of Product Engineering at the Karlsruhe Institute of Technology (KIT), Germany. After working as head of development of driveline systems and torsion vibration dampers at LuK GmbH & Co. KG, he moved on to the University of Karlsruhe – today's KIT – in 1996. His research focuses on product development processes as well as the support of product development by methods for computer-aided engineering, innovation and knowledge management in mechanical and automotive engineering.