Interdisciplinary Functional Systems Modeling Approach Applied for Hybrid Powertrain Development

A. Albers, C. Zingel, M. Maletz

Today's technical systems become more and more characterized by complex networking of multiple disciplines. The development process is attended by massive challenges in improving product quality with coevally reduction of development time and costs.

IPEK and AVL have established a research cooperation project to cope with that issue. The result is a software-based approach for functional decomposition and modeling of interdisciplinary systems and their interrelations to customer objectives. The standardized OMG Systems Modeling Language (SysML) has been applied and extended by an according profile to become the underlying documentation and communication basis for this approach. An application example of a hybrid powertrain is introduced to validate the additional benefit for automotive development engineers.

***Index Terms*— Model-Based Systems Engineering, SysML, Functional Modeling Approach, Interdisciplinary Systems, Interrelations to Customer Objectives**

I. INTRODUCTION

Dealing with complexity is a challenge for automotive development engineers, in particular caused through rapidly electrification of many system functions and hybridization of the powertrain. Today, about 90% of all innovations in automotive systems are electronic- or software-driven and up to 100 control units take place in upper-class vehicles [1]. The mission is a successful satisfaction of permanently increasing customer requirements in terms of performance, comfort and safety, for instance through manifold assistance systems with coevally simplified and more intuitive handling by the driver. Hence, reams of interacting and possibly conflicting mechanical, electrical and software systems have to be handled. In fact, the average automobile now has several millions of lines of code - more than a space shuttle [2] and consists of several tens of thousands of parts.

Emerging customer demands for individualization at high quality and efficiency, international markets, technical and legal regulations amplify this challenge on the one hand, cost and time pressure force vehicle manufacturers and suppliers to act rapidly on the other hand. The only way to meet this challenge is an increase of efficiency in research, development and production.

This paper introduces an approach to perform an important step towards higher efficiency by modeling interdependencies between functional system structures (applying elements from the C&C-M for SysML, [3]) and customer objectives in order to permanently sustain target orientation. The approach describes necessary steps to achieve an allocation from customer objectives to realized functions, performing system structure and characterizing parameters. Furthermore, a metamodel, realized as plugin profile for SysML ([4], [5]), the standardized modeling language for Systems Engineering provided by the Objects Management Group (OMG), is introduced.

II. CHALLENGE: THE TENSION FIELD OF VEHICLE AND POWERTRAIN DEVELOPMENT

IPEK and AVL have established a research cooperation project in order to identify and classify reciprocal effects between subsystems within hybrid automotive powertrains. The aim was to establish a model-based approach in contrast to document-based communication between involved development departments in order to avoid redundancies and inconsistencies in information transfer.

One of the first findings was a strong networking of emerging interdependencies between subsystems and hence a necessity of a wide-ranging communication, data & information exchange between involved disciplines. Thus, different customer objectives never may be met separately without also considering the impacts on other objectives. This leads to a tension field for vehicle development, optimization and validation as illustrated in Figure 1.

Manuscript received July 15, 2011; revised August 09, 2011.

1st author: Albert Albers: IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany (phone: 0049-721-60842371; fax: 0049-721-60846051; e-mail: Albert.albers@kit.edu).

2nd (corresponding) author: Christian Zingel: IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany (phone: 0049-721-60845486; fax: 0049-721-60845487; e-mail: Christian.zingel@kit.edu).

3rd author: Michael Maletz: AVL List GmbH, A-8020 Graz, Austria. (phone: 0043-664-8379249; e-mail: michael.maletz@avl.com).

Figure 1: Tension field of vehicle development

The main challenge is to set up an approach how to handle and how to comprise these partially conflicting targets throughout the whole vehicle and powertrain development process in order to enable frontloading and to channel critical decisions into the right direction.

III. NETWORKING OF FUNCTIONS WITH REQUIREMENTS

A first important task is to clearly define objectives as requirements and a quantifiable validation of target achievement state. A further essential task is a consistent understanding of the meaning of the term **function** in contrast to a **functional requirement**. A functional requirement defines what a system or a subsystem has to perform using measureable parameters with fix, minimum or maximum values. The realized solution in a system (an assembly or a single component of the developed product) performs one or several function(s). These functions receive input flows (any kind of material, energy or information input) and process them using or comprising further parameters (properties of the component that performs the function). The generated output is again one or a set of any kind of flows (material, energy or information). The required parameters within functional requirements have hence to be met by the emerging output flows (cf. Figure 2).

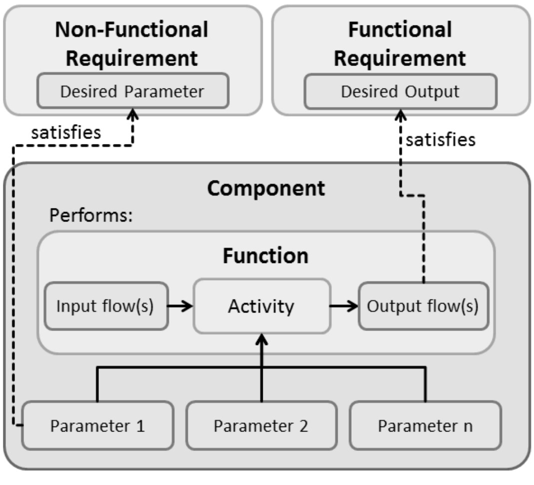

Figure 2: Connectivity of Functional and Non-functional Requirements

Non-functional requirements can also define parameters to be met. In contrast to functional requirements they are not the output of functions, but property parameters of system components (i.e. friction coefficient or material density).

An approach which applies such a functional view on systems requires a modeling language and an application guideline for establishing a functional system model and a requirement model.

Furthermore, requirements as well as components need measureable parameters which can be mapped to each other in order to track the relation of required and satisfying system properties. This also applies to input and output flows (the interfaces of components and their deposited functions to perform). Usually, input flows are given from border conditions like technical or non-technical neighbor-systems or coming as output from other existing functions. These input flows are processed (i.e. transmitted or converted) to output flows within the function. The processing procedure itself relies on physical relations, technical processes or software-code and is primarily determined within discipline-specific development tools. The introduced functional view reduces these mostly very complex activities to a characterizing name and few parameters. These can either be comprised as given properties or result from the calculations and specify system components.

IV. AN INTEGRATED FUNCTIONAL MODELING APPROACH FOR INTERDISCIPLINARY SYSTEMS

The aim of the presented functional modeling approach is an interdisciplinary application throughout all involved development departments in order to improve information transfer, communication and documentation.

Especially customer objectives are to be comprised at all times in order to develop and balance the system according to desired priorities, considering all relevant appearing interrelations.

First of all, a formal metamodel has to be defined for clear differentiation between the meanings of applied modeling elements for describing an interdisciplinary system in an adequate manner. Such a metamodel defines all elements of a modeling language and allows the creation of relations between them. Secondly, a software tool is crucial for a successful implementation in industrial development processes and existing tool environments. The OMG Systems Modeling Language (SysML, [4]) lends itself to become an expedient tool for the modeling approach at hand for several reasons. First of all, it has established as the international standard modeling language for systems engineers. Coming as an adapted profile of the well-known and popular Unified Modeling Language (UML, [6]) for modeling systems of any kind, it is initially set up in a very generic manner. Due to that UML- and SysML-metamodels rely on the OMG-Standard Meta Object Facility (MOF, [7]), both are easily extendable by more specific metamodel in form of plugin language packages. This is also the reason for manifold available software platforms and emerging support of software interfaces for data transfer to CAD, CAE and simulation tools (i.e. via ISO 10303, the STEP standard family).

Consequently, SysML was applied as modeling language and its profile has been extended by additional elements and relations to enable modeling of technical systems in a functional manner with networking to the customer objectives. The main elements of the according metamodel with logical relations are illustrated in Figure 3.

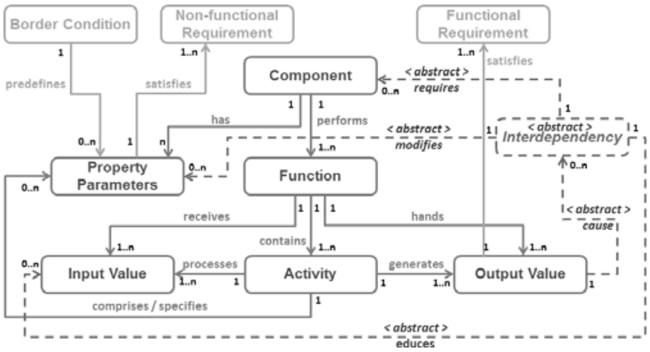

Figure 3: Metamodel for functional systems modeling

The main elements for modeling customer objectives (i.e. performance, costs etc.) and border conditions, for instance coming from purchased parts or environmental conditions are **Requirements** in SysML. The introduced metamodel extends this modeling element by **Functional requirements** for defining satisfying function outputs of systems. In fact, this is realized by setting either lower/upper or fix limits for function output values (i.e. min. torque, max. vibrations). **Non-functional requirements** define limits for non-functional property parameters of components like costs, recyclability or weight. **Border conditions** predefine unswayable conditions to be comprised in the product development (i.e. existing data interfaces, outside temperature etc.). All these restricting elements are depicted in orange color in Figure 3.

Functions can only be performed by a system component when it interacts with its environment. Hence, functions receive input values from interacting neighbor systems or system components and process them in one or more activities to generate output values. These are handed to other functions or to neighbor systems (i.e. torque, control commands etc.) in order to network the performed sub-functions within the overall systems' function-structure. Thus, also unintended output values like heat or vibrations can appear which may cause interdependencies with other component functions in form of educed input values. This fact is accommodated within the presented metamodel by means of the **abstract element "interdependency"** and its networking to related elements (red colored in Figure 3). First of all, an output value can cause such an interdependency, which consequently transmits the emerging outputs as input value to affected functions. Furthermore, the resulting effects of this interdependency (such as too strong vibrations, noise, heat, pollution, rattling, signals, data information etc.) can cause an exceedance of required limits and hence require new components or modified component parameters to reduce these emerging effects (i.e. implementing lubrication, sealing, heat isolation, damping, error handling code systems etc.). These new components performing the required side functions are integrated into the entire system in a new iteration step. Again, new output values are generated which again exert influence onto affected systems and – if necessary – the next iteration is triggered.

The implementation of the presented metamodel for functional modeling of interdisciplinary systems in a software tool is realized as a language subset of SysML. A remaining challenge in application of SysML is a missing guideline how to successfully integrate such an interdisciplinary product model in industrial development processes using a tool environment, as also stated in a survey of the MBSE Focus group of INCOSE [8].

Hence, AVL and IPEK have also developed a modeling guideline for a modeling language including the metamodel at hand, which is roughly characterized by the following activities depicted in Figure 4.

- Define customer objectives and main Use Cases
- Derive functional and non-functional requirements and border conditions
- Specify measurable desired parameters
- Specify system boundary and input & output values
- Define main function of entire system
- Decompose main function into subsystems and performing functions with input & output values
- Derive concrete system structure
- Transmit existing property parameters to discipline-specific tools (i.e. CAD, CAE, Simulation, CASE, MBD etc.)
- Receive property parameters from discipline-specific tools
- Derive dynamic system behavior (system states and state change sequences)
- Define and transmit Test-Cases for simulation tools
- Receive results from simulations of Test-Cases
- System validation by means of requirement satisfaction degree
- If necessary: modify parameters or add new subsystems to perform side-functions in order to fulfill requirements

Figure 4: Modeling Guideline

This guideline perceives as framework for modeling all interdisciplinary relevant aspects of a technical system and not as a definite modeling instruction. Depending on the application scope of the modeling language, it can be modified or extended by more specific modeling activities.

Due to that the modeling language is permanently improved and extended by further modeling elements, the scope for this paper is set to the first two major activities, the **definition of the system objectives** and **modeling of the functional structure** of a system. These activities build the basis for the following activities of developing the system structure (geometry, form, software code etc.) and dynamic system behavior.

Possible iterations are not only performed after having completed all previous activities, but rather at all times when interrelations appear. That is why this activity is also regarded here.

First of all, **customer objectives** are **defined** and modeled using Functional and Non-functional requirements. Possible objectives of technical systems are for instance performance, energy efficiency, durability or costs. Measureable parameters and priorities (i.e. by weighting the parameters) are added for a better traceability of the degree of target achievement within the tension field of development (cf. Figure 1 for automotive systems).

Afterwards, the **system boundaries** and system **interfaces** to neighbor (technical or non-technical) systems are **modeled** using structure diagrams like the Internal Block Diagram in SysML [5]. The modeling of functional system structures is done applying elements from the Contact-&Channel-Model (C&C-M) for SysML, which also has been developed at the IPEK [3]. Coevally, this model is improved and applied into an industrial environment for the first time. Using this C&C-M for SysML enables modeling of interfaces as different types of input and output values, which are there called Working Surfaces. Giving an example, interfaces of a vehicle to non-technical systems are the human-machine-interface (cockpit) or influencing weather conditions (temperature, humidity etc.). Interfaces to technical systems may appear for communication systems like GPS or radio, but also for radar sensor systems like adaptive cruise control (ACC). In case of developing a subsystem within a vehicle assembly (i.e. powertrain), manifold interfaces like chassis mounting, energy supply or drive torque transmission to the wheels appear.

Afterwards, the **main system function** is **defined** (i.e. "transportation of payload" for a vehicle) and specified through property parameters (i.e. "number of passengers [-]", "payload [kg]", "range [km/miles]" etc.).

In a next step, the **main function** of the system is **decomposed into subsystems** and performing functions (i.e. subsystem "powertrain", function "convert chemical energy from fuel to drive torque"), again specified by characterizing property parameters (i.e. "weight [kg]", "manufacturing costs [$]", "size [m]" etc.). Additionally, **input** values like "fuel flow [l/s]" and **output** values like "drive torque [Nm]" are **defined** and allocated to the according activities like "convert energy".

The resulting **sub-functions** with input and output values and the property parameters of the performing subsystems are directly **networked to** the **requirements** in order to verify the satisfaction degree (i.e. "min. acceleration from 0 to 60 mph = 8 sec." is networked to the drive torque, transmission ratios and to the vehicle mass amongst others). At this time, a **first iteration** can appear due to insufficient system properties like "efficiency factor" or "weight", which may cause modifications of property parameters (i.e. through lightweight design to reduce weight) or new subsystems (i.e. lubrication system to improve efficiency factor and hence drive torque). The emerging subsystems also gain requirements and property parameters and the development activities are run through again.

When the **system structure** is **derived** from the given information in the interdisciplinary system model, relevant **information** like property parameters as well as function structures (networked inputs, activities, outputs) is **transferred** to discipline-specific development tools. The emerging results are again transferred back to the interdisciplinary model in order to share the information with other affected disciplines. Coevally, simulation results and calculations can be applied for validation of system properties through balancing with required parameters and outputs. After having derived the entire system behavior through states and state changes or behavior sequences, a validation using defined test cases is done. As mentioned before, the latter activities of system structure and system behavioral design are not regarded in detail here. The approach at hand actually emphasizes on paving the way for improving the efficiency of these activities using the functional system model as information transfer platform.

A short application example of a hybrid powertrain demonstrates how the realization of the requirement "ability to recuperate electrical energy" is done by functional modeling of this system.

V. APPLICATION EXAMPLE: HYBRID POWERTRAIN

The application example starts with an extract of customer objectives and border conditions for a hybrid powertrain, depicted in Figure 5.

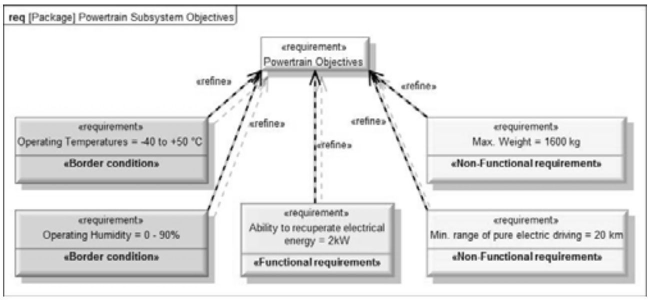

Figure 5: Extract of hybrid powertrain requirements

The **Requirement Diagram** shows several requirements, easily distinguishable by their differently colored appearance (brown: Border condition, orange: Functional requirement, yellow: Non-functional requirement). So the ability to recuperate electrical energy is modeled as functional requirement with defining a minimum recuperation power ratio of 2kW what equates a charge current of 10 Ampere at a Battery Voltage of 200 Volt.

After having defined major requirements, the system environment is modeled in an **Internal Block Diagram (IBD)**. A hybrid powertrain has manifold interfaces to its environment, wherefore the following diagram in Figure 6 only shows an extract of the real existing interfaces.

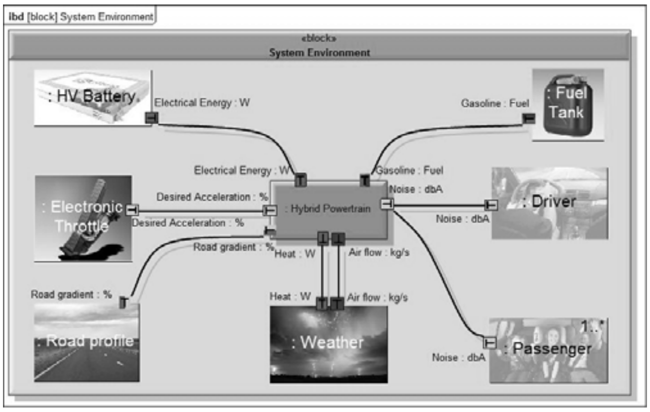

Figure 6: Hybrid Powertrain Environment

The diagram shows the system, which is to develop, in the middle, networked to several interacting technical systems like the high voltage (HV) battery, the fuel tank, or the electronic throttle signal. Also non-technical systems like the driver or the weather are depicted. The interfaces which in fact correspond to the input- and output-values are depicted in three different colors or combinations of those:

- i.e.Red = Material transfer

- i.e. Blue = Energy transfer
- i.e.Yellow = Information transfer

Beside the first rough information using these color-codes, the interfaces are named and have a “data type”, for instance the Noise transmission between Powertrain and driver/passengers is scaled in dbA, the acoustic pressure. Very often, these interfaces are simplified or a real interface is divided into multiple interfaces within the system model. This is intentionally done for clarity reasons or because the remaining interfaces are not needed in this model.

The next step is to model the functions and their input- and output values including networking them to a function structure in an **Activity Diagram**. This is done for an extract of the hybrid powertrain in Figure 7.

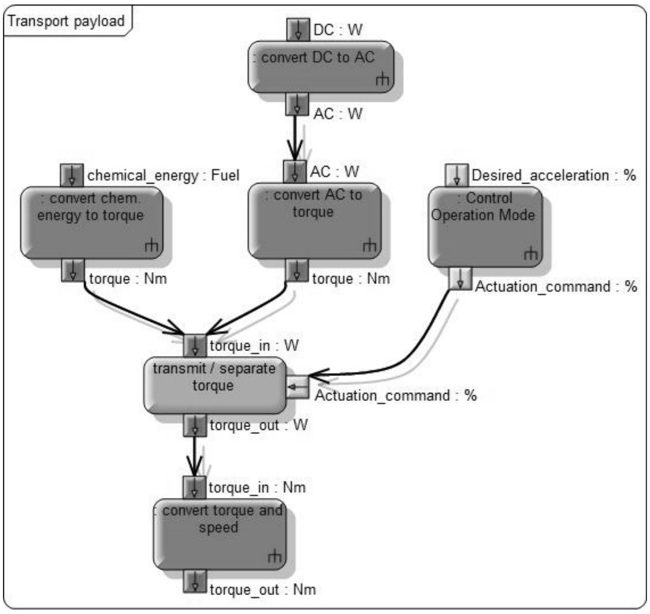

Figure 7: Function structure of hybrid powertrain

After having modeled the activities, the input- and output PIN’s in SysML, which equal the values, are added and finally networked. Now the structure of the performing technical components can be modeled in their own Internal Block diagram (cf. Figure 8).

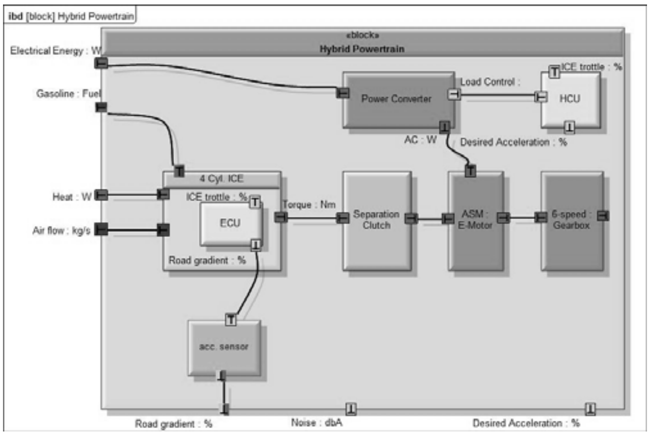

Figure 8: Hybrid powertrain structure with interfaces

The components are colored in order for an easier differentiation (i.e. control units are illustrated in yellow color). The interfaces are again colored according to their transmitted kind of information (material, energy, information). System structures are frequently very strong networked. Depicting them all in one diagram will quickly become very confusing. Hence, additional diagrams can be added for dividing the model into multiple aspects. For instance, Figure 9 only shows the communications between the control units, which is left out in the previous diagram.

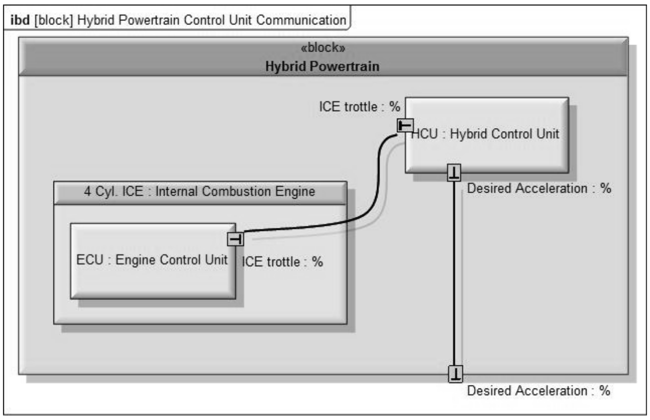

Figure 9: Control unit communication structure

The modeled system components can now be extended by property parameters, which in SysML are called Block Properties (Values). These properties can additionally gain units and default values. The last presented step in this application example is to model the functional structure of the hybrid powertrain at hand. For this purpose, a new Activity Diagram is created and the already existing activities from Figure 7 are assigned to the previously modeled system components. This results in the diagram shown in Figure 10.

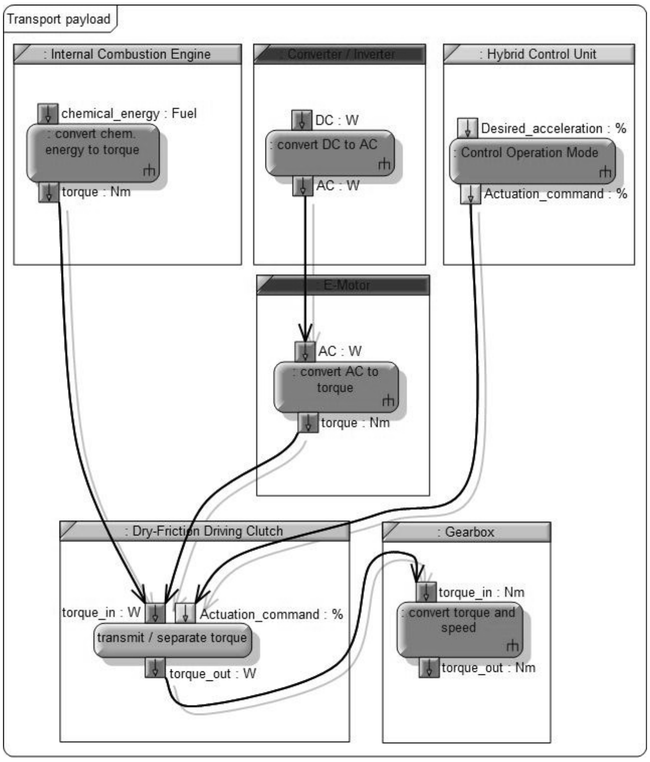

Figure 10: Component functions of the hybrid powertrain

Now the functional structure is mapped to performing components. Not depicted is the fact that also the activity input- and output values now are allocated to the interfaces (in SysML called Flow Port) of the structural elements (in SysML called Block).

The several elements of the interdisciplinary hybrid powertrain can now be allocated to the requirements. A simple example is done for the satisfaction of the functional requirement “Ability to recuperate electrical energy = 2kW” in Figure 11.

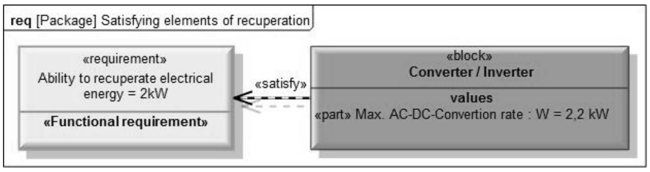

Figure 11: Satisfaction of Requirements

VI. CONCLUSION AND OUTLOOK ON FURTHER RESEARCHES

The introduced functional modeling approach for interdisciplinary technical systems combines following major benefits:

- Significantly improves development efficiency through saving costs and time.
- Clear specification of essential terms to be handled with in product development (i.e. function)
- Improvement of communication and collaboration of interdisciplinary development teams
- Clear networking between system components and performed functions
- Sustainable documentation through avoiding redundancies and consistency tests (i.e. alarms in case of unused or false networking of interfaces)
- Provides a modeling guideline for easier application in development process
- Comprises customer objectives in relation to realized systems and characterizing functions and properties
- Allows statements regarding target achievement state through measurable parameters
- Enables computer-based information transfer between discipline-specific tools

The next research task is the identification of methods to improve the usability and handling of SysML software frameworks in cooperation with tool providers and software engineers. Especially a structured clustering of structure and functional structure diagrams into different discipline-specific views will be focused within these researches. Coevally, the presented modeling approach will further be verified by application in several productive projects of AVL and the feedback of involved development engineers as model users will be comprised in advancing this approach.

Furthermore, the implementation of quantified parameters into SysML to enable statements about the quality of impacts of interrelations between sub-functions on the entire system functionality is assessed. For this purpose, also semantic technologies will be considered. The implementation of interfaces to PLM systems with integrated requirement management is another current research task (according to [9]).

REFERENCES

[1] Frischkorn, H.-G.: Keynote-Topic: IT in future automotive environment. In: Proceedings of the INFORMATIK 2003, Frankfurt, Germany, 2003.

[2] Abloeser, F. (June 7th, 2010), IBM Addresses Complexity in Automotive Systems [Online]. Available: http://www.prnewswire.com/news-releases/ibm-addresses-complexity-in-automotive-systems-95767844.html

[3] Albers, A.; Zingel, C.: Interdisciplinary Systems Modeling Using the contact & Channel-model for SysML. In: Proceedings of the International Conference on Engineering Design (ICED ‘11). Copenhagen, Denmark, 15-18th of august, 2011.

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

[5] Friedenthal, S. et al.: A Practical Guide to SysML – The Systems Modeling Language. 2008 (publisher: Morgan Kaufmann, Waltham, Massachusetts)

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

[7] Specification: OMG Meta Object Facility (OMG MOF™), Version 2.0. http://www.omg.org, 2006 (publisher: Object Management Group).

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

[9] Maletz, M. (2008): Integrated Requirements Modeling – A Contribution towards the Integration of Requirements into a holistic Product Lifecycle Management Strategy. PhD Thesis (publisher: Technical University of Kaiserslautern. ISBN-13: 978-3939432920)