INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED11
15 - 18 AUGUST 2011, TECHNICAL UNIVERSITY OF DENMARK

ANALYZING THE DYNAMIC BEHAVIOR OF MECHATRONIC SYSTEMS WITHIN THE CONCEPTUAL DESIGN

Frank Bauer[1], Harald Anacker[1], Tobias Gaukstern[1], Jürgen Gausemeier[1], Viktor Just[2]
[1] Product Engineering, Heinz Nixdorf Institute, University of Paderborn
[2] Control Engineering and Mechatronics, Heinz Nixdorf Institute, University of Paderborn

ABSTRACT

The increasing penetration of mechanical engineering by information technology enables considerable benefits. This is expressed by the term mechatronics, which means the close interaction of mechanics, electric/electronics, control and software engineering to improve the behavior of a technical system. The development of such systems is a complex and interdisciplinary task. Consequently a domain-spanning specification is required, which describes the system in total and builds the basis for all further communication and cooperation between the experts from the involved domains in the concretization. In order to validate the system, different tests are accomplished during the concretization phase. In this contribution we present how dynamics analysis may be integrated already during the conceptual design phase. For this purpose a simulation tool is used for the validation of the dynamic behavior of the system already on the basis of the principle solution. The refinements effected during the simulation are transferred back into the principle solution. This improves the provided information for the following domain-specific concretization.

Keywords: mechatronic, conceptual design, principle solution, solution pattern, dynamic analysis

1 INTRODUTION: DOMAIN-SPANNING CONCEPTUAL DESIGN AND DYNAMICS ANALYSIS OF MECHATRONIC SYSTEMS

The products of mechanical engineering and related industrial areas are often based on the synergetic interaction of mechanics, electrics/electronics control and software engineering. The term mechatronics expresses this interaction. The aim of mechatronics is to optimize the system’s behavior by using sensors, actuators and information processing. Sensors obtain information about the systems environment and the system itself. The processing of this data and the controlling of the actuators enable the system to adapt to the current situation. The design of such systems is a challenge. Existing methodologies for the development of mechatronic systems reached a high level; but they focus on specific tasks of the involved domains. There is too less consideration to the domain-spanning interactions. There is still missing a domain-spanning design method and supporting software tools integrating mechanical-, electrical-, control- and software engineering.

In accordance with existing methodologies, mainly the VDI guideline 2206 [1] the development process of mechatronic systems can be divided into two main phases: the domain-spanning conceptual design and the domain-specific concretization. Within the conceptual design, the basic structure and the operation mode of the system are defined. All results of the conceptual design are specified in the principle solution. The description of the principle solution has to include all necessary information for the following concretization, which takes place for the involved domains. These processes are carried out in parallel and aim a complete description of the system. The domain-specific models are therefore integrated into the final solution. Throughout the conceptual design the involved domains have to cooperate. For the specification of mechatronic systems, a lot of different modeling and specification languages are available, e.g. SysML [2]. These specification and modeling languages do not fully meet the requirements for a domain-spanning description of the system. For this reason, a new specification technique for the description of the principle solution has been developed [3]. The following aspects need to be taken into account: environment, application scenarios, requirements, functions, active structure, shape, behavior and system of objectives (Figure 1). The latter is only required for self-optimizing systems. The behavior is considered as group of partial models because there are various types of behavior, e.g. dynamic behavior of a multibody system, electromagnetic compatibility or heat transfer.

The mentioned aspects are mapped on computer by partial models. A software tool which can used to describe mechatronic systems using the specification technique is the Mechatronic Modeller. The Mechatronic Modeller offers a separate editor for each partial model and is based upon a metamodel. The principle solution is computer-internally represented as a data model, which is the instance of this metamodel [4]. At this design stage first analysis are possible. This includes computer-aided analysis of robustness, reliability, product-structuring or manufacturing costs. Further tests and analysis were conducted within the concretization phase, where the involved domains use specific methods and tools. As a consequence the validation of the specified principle solution is deferred to later point of the design process. In order to verify the information provided by the principle solution, a validation should be done as early as possible during the design process.

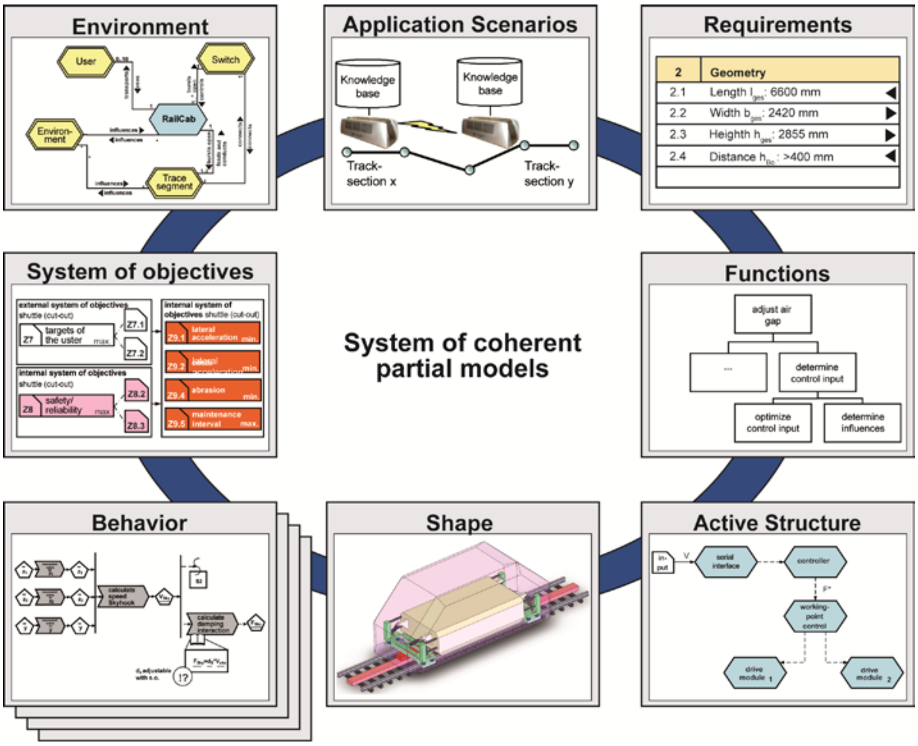

Figure 1: Aspects respectively partial models for the domain-spanning description of the principle solution of mechatronic systems [3]

A simulation-based design process including a dynamics simulation and analysis of a mechatronic system is completed in two abstraction steps. We illustrate these using the example of an active suspension module of a fully x-by-wire vehicle. First, a simple idealized model of the active suspension is built up. This model is used for dimensioning the passive suspension components and for the control design. The idealized model of the vertical dynamics consists of the two masses vehicle and wheel (Figure 2, left). A spring-damper element is used for the active suspension and the connection between wheel and ground. Each mass has only one vertical degree of freedom each. Due to a reduction of the vertical dynamics to only one wheel (quarter-vehicle), the axle kinematics being neglected, and an ideal actuator being used for an active adjustment of the spring pre-load c_A between the vehicle body and the wheel. The scope of the model is limited.

The dynamics control for the lifting motion is designed on the basis of this idealized quarter-vehicle model, with the controller design being based on the skyhook strategy [5]. This inserts a virtual spring and a virtual damper between a virtual coupling point in the “sky” and the vehicle body; along with an additional active spring-damper force between the body and the wheel. Figure 2 displays the simulation results of the passive vehicle behavior and of the closed-loop-controlled with an ideal suspension actuator. The red graph marks the desired behavior of the simulation.

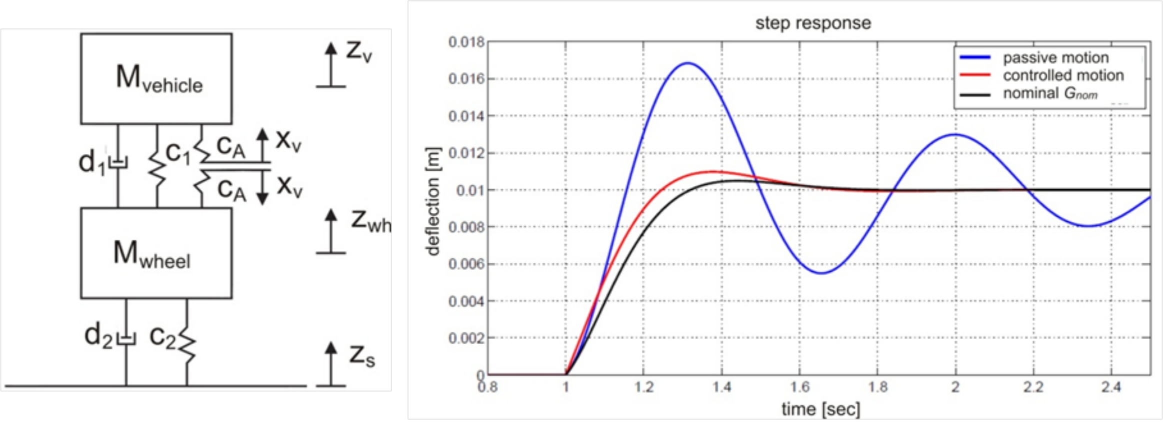

Figure 2: First analysis model and simulation results

In a second step a more detailed multi-body system model is built that takes into account the axle kinematics and the actuator system. This information is used for designing and fine-tuning the control strategy. Based on the requirements of the lifting range of the wheel suspension and on the stiffness of the passive vehicle-body spring, an idealized axle kinematics is designed. The necessary active force F_A is induced via a torsional moment M_A. To take into account the axle kinematics, a multi-body system model is built up whose torque M_A is commuted by means of an electric motor. In order to apply the active torque M_A on the suspension, the motor is coupled to the traverse link via a torsion spring. Thus the active torque can be preset via the motor-angle control which is realized by a cascade control and laid out on the motor model. Figure 3 shows the resulting simulation model which can be used in order to create first control strategies. Also the multi-body system defines the necessary kinematics for the suspension module [6].

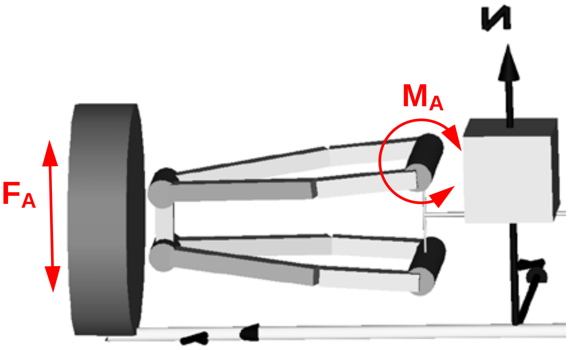

Figure 3: Multi-body system of the active suspension module

2 DYNAMICS ANALYSIS WITHIN THE CONCEPTUAL DESIGN PROCESS

In the following we present an approach which includes a dynamic analysis into the conceptual design process. The aim is to derive all required information for the dynamic simulation from the aspects of the principle solution. Furthermore the results of the analysis are transferred back and refine different aspects. An early validation of the systems dynamics supports the following domain-specific concretization. The domains control- and software-engineering benefit from a first dynamic model of the system, in order to start with domain-specific refinement. Also the dynamic affects the required kinematic and shape of the system which is elaborated within the mechanical-engineering. The electrical-engineering is targeted by the two other domains.

2.1 Design Process

The design of the principle solution (Figure 4) covers all partial models of the presented specification technique except from the system of objectives. In the following we consider advanced mechatronic systems which are not self-optimizing. The design process is divided into three steps; specify aim, synthesis and analysis. The first step starts with the clarification and definition of the task. This includes the aspects environment, application scenarios and requirements. Based on these aspects the systems functionality is defined in terms of a function hierarchy. The synthesis begins with the choice of possible solution pattern to fulfill the required functionality. We will describe the structure of solution pattern later in more detail. Each solution pattern can affect the structure as well as the behavior of the system. The structure of the system is modeled with the active structure. The initial behavior is modeled with states and activities. The partial model behavior – states defines the states of the system and the state transitions. The state transitions describe the reactive behavior of the system towards incoming events. The partial model behavior – activities describes the logical sequence of activities in the system. Especially, parallel executed activities and their synchronization can be described this way. The behavior marks the target for the later dynamics simulation. An initial geometry model is required for a dynamics analysis. Modeling the shape is the last step within the design of the principle solution as presented in Figure 1.

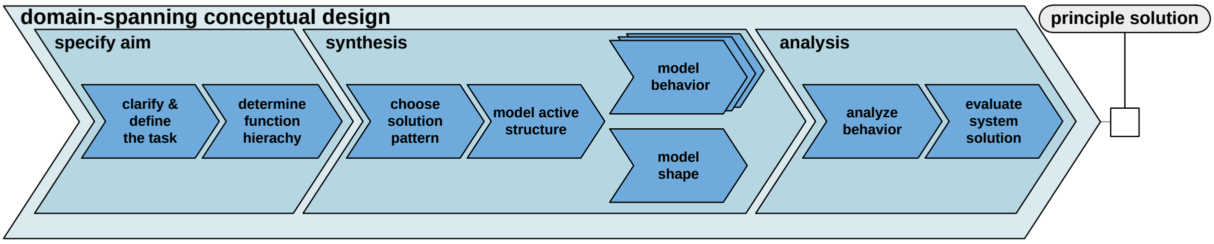

Figure 4: Design Process including an early validation

Due to the early design phase, the domain-spanning concept does not describe all aspects of a system in detail. Especially the number of precise physical parameters is limited. A simulation of the system at this development stage has to regard the provided level of information. Therefore a first analysis has to be done with models that consider this restriction. We refer to these types of models as idealized models, which only contain the necessary level of detail for a first validation. The analysis starts with the derivation of an idealized simulation model based on the active structure and the shape model. The system is simulated and the results will be analyzed with respect to other partial models. The simulation results are compared to the initially defined states and activities. It is checked if the dynamic model shows the predefined behavior. The parameters set within the simulation are compared to the requirements. In case that the simulation does not match the requirements the simulation model is adjusted, simulated and analyzed again. The adaptations within the analysis process are transferred back to the partial models of the principle solution. The active structure is enhanced when a new element is included into the simulation in order to achieve the desired behavior. This also causes an adjustment of the function hierarchy. Parameters set within the simulation model are added to the requirements list which is improved step by step. The behavior activities and states are refined by the simulation results. The geometric information assigned with the simulation is considered in a modified shape model. This process is repeated until the dynamic behavior fulfills the required dynamics necessary for the development task. This indicates the end of the conceptual design phase. In the following subsections we illustrate the design process for the already introduced active suspension module.

2.2 Design of the principle solution

Active suspension modules are used in vehicles to improve the driving characteristics in general. This includes for example a higher driving comfort for the passengers or a higher driving safety. The environment and an application scenario are described by a driving vehicle; where the ground caused disturbance on the wheel. Several requirements can be deflected for active suspension to achieve the outlined defined behavior. The overall function of the module is to keep the chassis in a rest position. A sub-function is the absorption of disturbance. A first simplified active structure, requirements and functions are presented in Figure 5. The whole module consists of a chassis, an active suspension and a wheel. Relations between system elements are represented by flows. Three flows are feasible: material, energy and information flow. In the example only energy relations between the system elements are modeled. Related to the active structure the behavior of the system elements can be characterized. Only two states are considered at this design step. The chassis can be in the rest position or in vibrations, which are caused by the disturbance. In the later no active suspension is applied. The activities are therefore, to transmit the disturbance to the active suspension, absorb them and transmit a lower disturbance to the chassis.

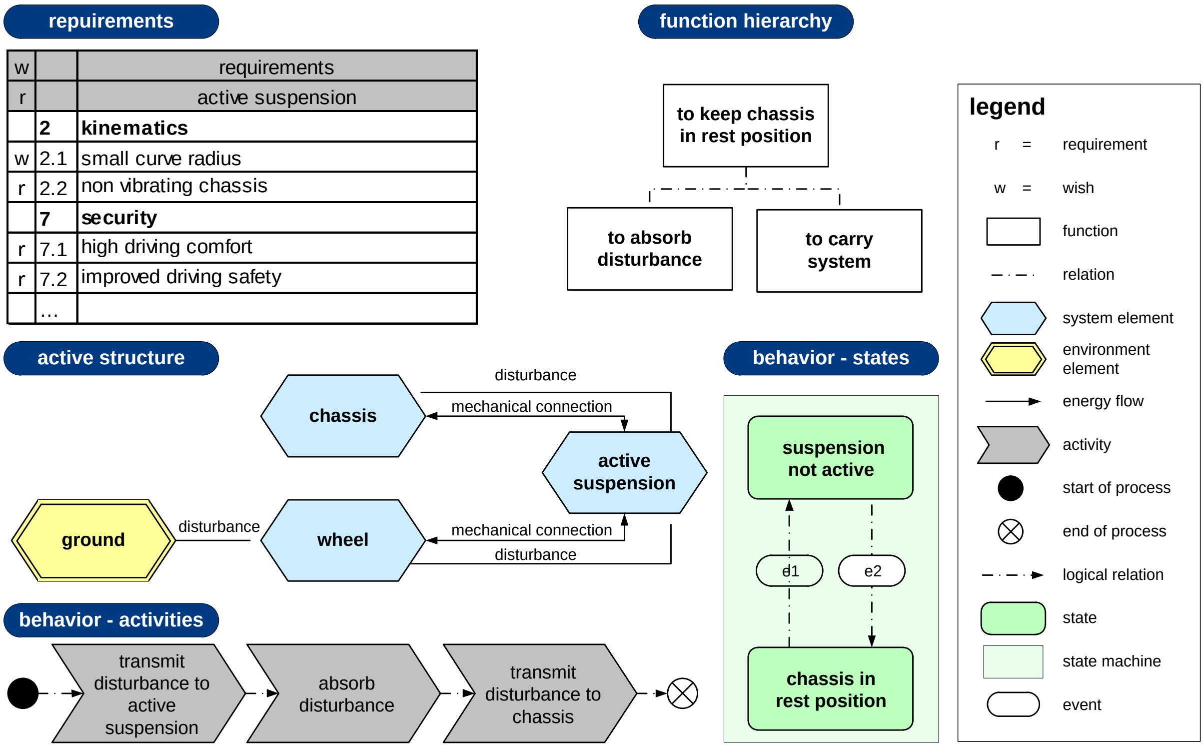

Figure 5: Principle solution for an active suspension module

A fundamental aspect of the design of mechatronic systems is the reuse of once successfully proven solutions in form of solution patterns. A pattern describes a context-specific problem and the core of the solution [9]. Furthermore a pattern is an established instrument to externalize and store the knowledge of experts. Generally the involved experts use their own terminologies during the development process. In order to deal with this challenge we use an adapted uniform specification of solution patterns, which was developed by DUMITRESCU ET. AL [10]. In the following we describe the specification of a solution pattern, which is structured in seven aspects (Figure 6).

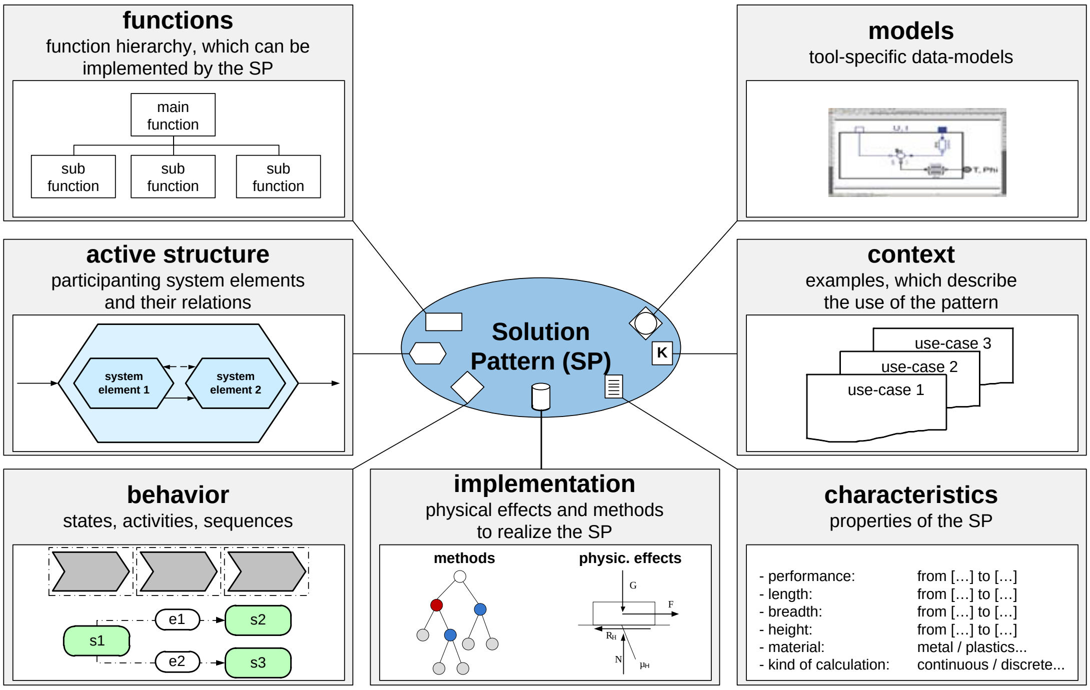

Figure 6: Aspects of Solution Pattern for the Design of Mechatronic Systems

In order to come up with the definition by Alexander, the developed specification of solution patterns for mechatronic systems describes the problem and all necessary aspects of the solution and its implementation. Within the described design process the solution patterns are used for two tasks: Firstly for modeling the principle solution. Secondly for derivating the idealized simulation model. The aspect functions express the problem description, which may be solved by the solution patterns. The description of the solution is separated into the aspects active structure (necessary system elements and their interrelation) and behavior (states, activities and sequences). In order to integrate the solutions into the design process, it is essential to clarify physical effects and methods for information processing. The aspect characteristics describes the significant different properties of the solution patterns. Once successfully applied solutions are generally attached to a specific context, so this aspect contains different use-cases. In addition to the specification of pattern by Dumitrescu et al. the aspect models is implemented. A model is a tool-specific mapping of the pattern for a specific problem or development task; in the presented design process it is a Modelica model. The simulation model of the whole system is build up from the single idealized models of the elements. Predefined models are selected to create a multi-body system which can be simulated.

2.3 Adjustment of the principle solution models

The result of the simulation is a new model, which describes the dynamic behavior of the system. This model is added to the behavior group of the principle solution. Moreover the other partial models have to be extended. The active structure shown in Figure 5 just includes a system element "active suspension". During the analysis this system element has been further refined; new system elements have to be added to the active structure. The multi-body system determines the necessary kinematics for the active suspension. The mechanical connections are mapped as energy flows between the system elements (Figure 7). The chosen combination of a electric motor, the torsion spring and the traverse link affects the behavior of the system. Further the activities are more detailed, compared to the initial model. The outlined sequence is valid for the different control strategies which can be applied. The state model consist now of three states. When the module is not in progress no active suspension is available. The two other states characterize different control strategies executed by the control unit. Although this example is quite simple, it shows the benefits of an early validation already within the conceptual design phase. The enhanced partial models build a better foundation for the following concretization phase. Especially for the control and software engineering a detailed description of the activities, states and events is useful. The three-dimensional simulation model can be used as a first shape model, which will be concretized within the domain mechanical engineering. Also specific solution elements will be selected during the concretization. The analysis of the required torques and motor dynamics provides information that is essential for selecting the appropriate actuators. These specific parameters will be added to the requirements list.

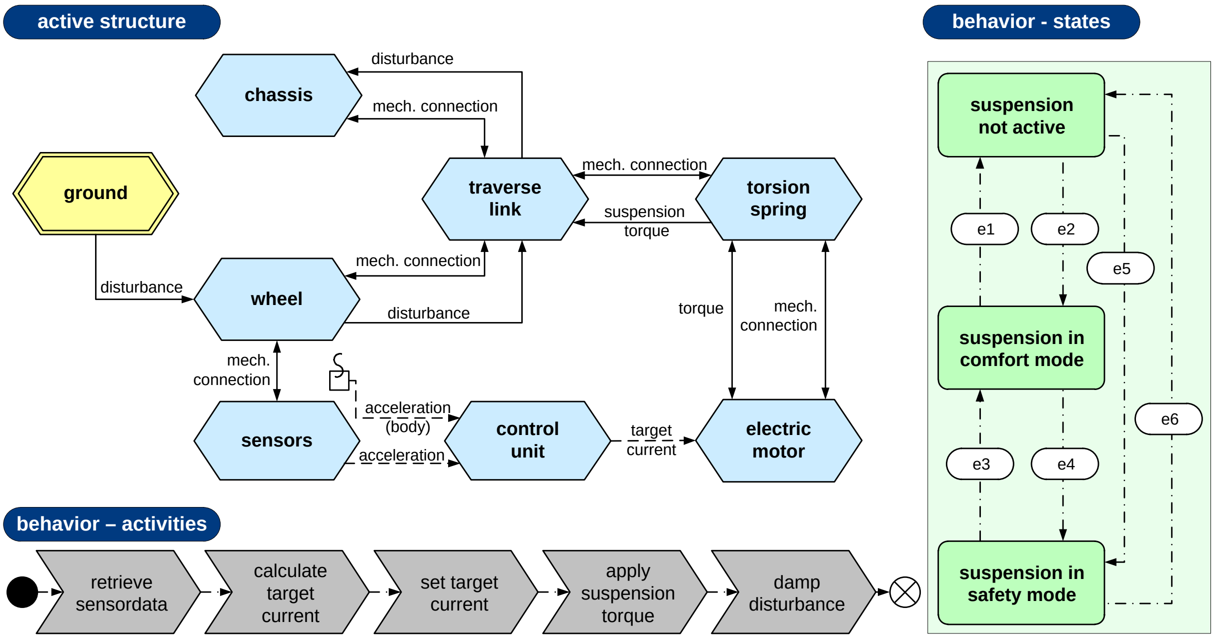

Figure 7: Adjusted active structure and behavior models

3 TOOL SUPPORT

The described design process is supported with two major software tools: The Mechatronic Modeller for modeling the required partial models of the principle solution and Dymola for the dynamics analysis. Dymola is an editor for the object-oriented physical description-language Modelica [7]. These tools are used during the conceptual design. Further a knowledge-based system is integrated to support the developer during the design process. The knowledge-based system contains a database of solution patterns and an ontology with domain specific and procedural knowledge. Originally “Ontology” is a philosophy discipline, used to structure entities and their relationships in certain area. This concept has been adapted to computer science in which an ontology refers to an intelligent data model. Today ontologies are a basic technology for semantic applications used to formalize and exchange knowledge. The Web Ontology Language (OWL) is the most used modeling language for ontologies. The basic elements from OWL are classes for collection of objects, individuals for specific objects and properties to model the different types of relations [8]. We use an ontology to model the relation between aspects of the specification technique, the solution patterns and the Dymola simulation model.

3.1 Modeling the principle solution

The design of the principle solution is supported by the Mechatronic Modeller. First the requirements list and the function hierarchy of the system is specified. The knowledge base assists the engineer during this conceptual task. The data base contains functions for the description of a mechatronic system. Within the ontology the dependencies between functions are modeled. For example the function “accelerate material” requires the function “convert energy” to produce the necessary acceleration energy. The stored information allows a computer-aided modeling of the function hierarchy. The mentioned dependencies support the engineer in adjusting or completing the function hierarchy. This formalization is required for the search and selection of possible solutions.

Similar to the functions, the solution patterns are stored in the database and structured by the ontology. The search is realized by a relation between functions and patterns. Based on the function hierarchy of the designed system the possible solution patterns are selected. For this task the requirements list is considered. A requirements list is normally just a text-based document written by the developer. With respect to a computer-understandable interpretation the requirements needed to be formalized. A first step is a template for the Mechatronic Modeller which includes basic requirements with necessary parameters valid for the mechatronic system. For example the maximum electric power supply. These parameters are compared with the characteristics of the patterns. Thus, patterns are selected which fulfill the required functionality and requirements. The active structure and behavior of the selected patterns are combined in the active structure and the initial behavior of the principle solution.

3.2 Analyzing the principle solution

Based on the partial models, that specify the principle solution, a semi-automated generation of a Dymola simulation model is achieved. This means that some parts of the simulation model can be generated automatically. Nevertheless the final assembly is up to the developer. Within Dymola a system is represented by components. A component could be a single element, like a sensor or a complex system which includes other components. Each solution pattern is attached to a Dymola component; an idealized model. Therefore each element within the active structure can be directly transferred into a Dymola simulation. The connections between components can be derived from the connections inside the active structure. Dymola offers different ports in order to model the connections between components. An energy port can be distinct into a mechanical or electrical port, which can again split up into one-phase and three-phase. Also the ports are subdivided by incoming and outgoing ones. The material, energy and information flows within the active structure also consider the direction. For a dynamics simulation information about the geometry has to be added. This information is provided by the solution patterns and a rough shape model of the specified principle. The developer uses this for the final assembly of the Dymola model and run the analysis.

4 CONCLUDING REMARKS

The presented approach combines the domain-spanning conceptual design of mechatronic systems and an early validation of the desired behavior. The result is a validated principal solution of the system which marks the starting point for the following concretization phase. The validation is realized by a dynamics simulation. During the design process the developer uses the Mechatronic Modeller for the domain-spanning description of the mechatronic system. Partial models specify functionality, structure, behavior and shape of the system. The behavior is modeled with activities and states. Based on this information an idealized simulation model is derived. The analysis phase is done within the tool Dymola. The simulation results are compared with the required system behavior and the simulation model is adjusted in order to achieve it. The adjustments are transferred back to the partial models of the principle solution to keep consistency.

Our future work covers the formalization of the aspects of the principle solution. For example, the required geometric aspects for a physical simulation have to be considered. In addition, we will also focus on software engineering, achieving a transformation from the evaluated principle solution to domain-specific models of the domain software engineering. Finally the mentioned tools need further concretization. The algorithms for the selection and combination of solution patterns have to be concretized.

ACKNOWLEDGEMENT

This work was developed in the project “ENTIME: Entwurfstechnik Intelligente Mechatronik” (Design Methods for Intelligent Mechatronic Systems). The project ENTIME is funded by the state of North Rhine-Westphalia (NRW), Germany and the EUROPEAN UNION, European Regional Development Fund, “Investing in your future”.

REFERENCES

[1] Verein Deutscher Ingenieure (VDI): *VDI-Guideline 2206 – Design Methodology for Mechatronic Systems*, Beuth Verlag, Berlin, 2004

[2] Weilkiens, T.: *Systems Engineering mit SysML/UML*. dpunkt.verlag, Heidelberg, 2. Auflage, 2008

[3] Gausemeier, J.; Frank, U.; Donoth, J.; Kahl, S.: *Specification Technique for the Description of Self-Optimizing Mechatronic Systems*. Research in Engineering Design, Vol. 20, No. 4, Springer-Verlag, London, 2009

[4] Gausemeier, J.; Dorociak, R.; Pook, S.; Nyßen, A.; Terfloth, A.: *Computer-aided cross-domain modeling of mechatronic systems*. In: Proceedings of the 11th International Design Conference (DESIGN 2010), Dubrovnik, 2010

[5] Gevatter, H.-J.; Grünhaupt, U.: *Handbuch der Mess- und Automatisierungstechnik im Automobil*. Springer- Verlag, Berlin, 2005

[6] Nachtigal, V.; Jäker, K.-P.; Trächtler, A.: *Development and Control of a Quarter-Vehicle Testbed for a Fully Active X-by-Wire Demonstrator*. In: Proceedings of 9th International Symposium on Advanced Vehicle Control (AVEC 2008), Kobe, Japan, 2008

[7] Tiller, M.: *Introduction to Physical Modeling with Modelica*. Kluwer Academic Publishers, Nowell, 2001

[8] Hitzler, P.; Krötzsch, M.; Rudolph, S.; Sure, Y.: *Semantic Web – Grundlagen*. Springer- Verlag, Berlin, 1. Auflage, 2007

[9] Alexander, C.; Ishikawa, S.; Solverstein, M.; Jacobson, M.; Fikshdahl-King, I.; Angel, A.: *A Pattern Language*. Oxford University Press, New York, 1977

[10] Dumitrescu, R.; Anacker, H.; Gausemeier, J.: *Specification of Solution Patterns for the Conceptual Design of Advanced Mechatronic Systems*. In: Proceedings of International Conference on Advances in Mechanical Engineering (ICAME 2010), Selangor, Malaysia, 2010

Contact: Prof. Dr.-Ing. Jürgen Gausemeier
Heinz Nixdorf Institute, University of Paderborn
Product Engineering
Fuerstenallee 11
D-33102 Paderborn
Germany
Phone: (0049) 5251 606267
Fax: (0049) 5251 606268
E-mail: Juergen.Gausemeier@hni.uni-paderborn.de
www.hni.uni-paderborn.de/en/pe/