Supporting Design Decisions in Interdisciplinary Product Development

B. Mensing*, C. Stechert**, M. Hagner*, J. G. Sánchez Ruelas**, U. Goltz*, T. Vietor**

 TU Braunschweig, Institute for Programming and Reactive Systems, mensing@ips.cs.tu-bs.de
TU Braunschweig, Institute for Engineering Design, c.stechert@tu-braunschweig.de

Abstract: Though interdisciplinary development projects are ubiquitous today, only few systematic support is provided for supporting designers concerned with such projects. Especially the support for design decisions is a crucial issue here since considering aspects from several disciplines at the same time is cumbersome. To overcome this lack of support, we propose an integrated approach covering the complete development process from requirements up to the final system design. Therefore, we present a generic procedure to analyze the fulfillment of goals from each discipline (here: software and mechanical engineering) and incorporate a cycle for iterative improvements to meet the product's design goals. This cycle consists of the identification of missed goals and especially of according counteractions based on the analysis of the system structure. Their impact on the development is used to provide a systematic decision support for designers in order to choose the most effective actions. To show the feasibility of our approach, we apply the presented methodology on a real-world case study. For the examined product development from the field of compound materials, all steps of our approach are exercised in a realistic environment.

Key words: Interdisciplinary Development, Decision Support, Conflict Resolution, Requirements Analysis, SysML

1. Introduction

Nowadays, in almost every branch and market, interdisciplinary development projects can be found. In the development of automation devices at least a mechanical structure, control hardware, and software have to be developed and integrated into a consistent total system while keeping track of the project management requirements (e.g., milestones, costs). However, due to their personal background (education, work experience) communication between project members can be problematic. Sometimes the same words have different meanings in the diverse disciplines. The way of thinking may vary: Mechanical engineers tend to think of functions and function carriers thus finally 3D hardware components of the system providing properties like geometry and material. Software engineers think in terms of input and output values, processed and stored at certain speeds. Because of the different demands, the different disciplines use own process models and supporting tools and are organized in different departments. Moreover, the different ways of thinking sometimes lead to a reservation: "I don't understand him and I even don't want to."

The problem occurs during integration of the subsystems into one total system. Although each subsystem was optimized, the total system does not necessarily work properly or does not satisfy all the stated system requirements. Rework is needed to adjust the system. The project plan is violated, milestones are not kept, cost

targets are missed, or the customer is displeased. Since this problem is well known, enterprises developed lots of control mechanisms during project runtime (gates), cross-sectional task-forces, and integration testing as early as possible. However, these approaches are just made to prevent the project from crashing. In order to gain synergetic effects and develop a consistent total system suitable for the defined requirements a new approach is necessary. Our approach can be used in an overall design process, utilize a common and standardized modeling, and provide generic and transferable methods and tools for analysis and synthesis. These methods and tools provide assistance for identifying and resolving conflicts by exploring relations among targets and components of the projected system. Furthermore, analysis results are used further in order to verify the fulfillment of requirements. Moreover, analysis and synthesis methods are incorporated into a loop of iterative overall system optimization, converging to an advanced system setup.

Our approach is useful especially in "new" disciplines where experts from different knowledge domains working together on a topic for the first time. For example, in the development of electric cars bringing high voltage systems into the car results in new challenges concerning package, safety, driving dynamics, etc. For each domain, the application and the necessity to communicate with experts from other domains is new. Similar difficulties appear during the development of handling or measurement devices for the production of new materials. The TU Braunschweig is partner in establishing "Forschungscampus" Open Hybrid LabFactory. Together with partners from industry, research is done to proceed with the use of new and lightweight materials with a focus on hybrid parts. Hybrid parts are made out of several materials but produced and handled as one part during the production process. For such hybrid parts new design rules and knowledge bases have to be developed. Moreover, the production process and all necessary handling and measuring devices have to be analyzed and developed from scratch. In this context, the paper uses the development of a wood plastic composite (WPC) scanner as an example of an interdisciplinary development project to evaluate the described methods.

2. Model-based Requirements Management Including Analysis

In earlier work the authors analyzed development processes of different knowledge domains with a focus on the management of requirements [19]. It was shown that although many different approaches are known and described in literature (see [16] for details) differences between the domains are relatively small. The main difference is in how to externalize knowledge, i.e., which modeling approach is used, which terms and definitions have which meaning, or where are the system boundaries.

The description of the system to be developed is often made in stating specifications. Specification documents are made up of requirements. For complex systems, requirements are often documented in tables. Some software is able to connect tables and specification documents. The specification document, as a formal and legally binding document, contains no open questions and is provided as a final version to team members and subcontractors. During early stages of product development, often requirements cannot be stated absolutely clear and values are not known. These open questions have to be clarified during the first weeks or months of the project. Moreover, some questions can just be answered through analysis and tests in later development phases. Therefore, developers often use their own requirements lists alongside with specification documents to focus on their specific area or to use them as "working papers". These lists include very specific and detailed requirements as well as restrictions, knowledge, due dates, available budget, and open questions. As almost every engineer has access to a spreadsheet

program, such as Microsoft Excel, this is often the first choice for the "informal" documentation of requirements. For an interdisciplinary development project, however, it is not possible to consolidate such locally maintained lists and to avoid redundancies and contradictions. Even if all developers would use the same table based requirements management system some drawbacks are apparent:

- the lack of clearness, because of big tables,
- the lack of consistency, because developer will start to use their own lists again if clearness is not given,
- the lack of transferability, because a list is difficult to connect to other (analyzing/developing) software.

One approach to overcome these drawbacks is the use of an uniform modeling language for system development. The Systems Modeling Language (SysML) was invented to provide a common basis for the development of complex systems [12]. In the systems engineering philosophy a system consists of objects and relations between them [4]. For complex systems the objects can be described as subsystems. In such a way it is possible to come from an abstract to a detailed view of the system. Together with the use of different viewpoints, i.e., different partial models, experts from different domains are able to work together [17]. In SysML, on an abstract system level for example hardware and software, components can be described as blocks, functions can be modeled as activities, and requirements can be modeled in requirements diagrams. An extension to SysML for an improved requirements management was developed by the authors [15], formalized in a profile, and is already used in several research projects [18, 19, 13, 10].

The first step is to formulate goals and targets. Then the situation has to be analyzed, i.e., who are stakeholders, what are neighboring systems, which use cases from the whole life-cycle have to be considered, which factors of what market are influencing the product. Next is to formalize requirements and to document them in the model. Requirements on system level are identified and related to overall project targets. On the other hand, requirements are allocated to subsystems that should add a share to their satisfaction. Thus, the overall targets are satisfied if the requirements from multiple disciplines are satisfied. In addition to these inter-partial-model-relations, some intrapartial-model-relations are important, e.g., geometric interfaces between hardware components or functional interfaces between software components. For the requirements model, traces between requirements can be given. These show interdependencies describing system level (vertical) or support/conflict (horizontal).

Basically, the aim of system modeling is to assist and to support the designer. Thus, the authors developed analysis tools to interpret the model, find new dependencies, and support decisions. VBA-based tools are generating tables from the model to show existing conflicts and to identify new possible goal conflicts [15]. Additionally, the requirements fulfillment of a special concept and its uncertainties related to new technology and future customer behavior can be evaluated and reported [13].

On the software side, requirements fulfillment is evaluated by model-based analysis of non-functional properties, which has become more important in the past years. The goal is to enable early-development stage analysis to have a feedback. One of the most important development languages is UML. To adapt UML to the non-functional analysis domain, UML profiles are used. The OMG proposed the MARTE (Modeling and Analysis of Real-Time and Embedded Systems) [11] that contains stereotypes and tagged values.

Based on the possibility to model the structure and the behavior of a system and to parameterize it using UML profiles, different approaches have been proposed to use the system model as a basis for different analyses [2, 14].

In the past, our approach was to employ different views for different aspects to help the developer to concentrate on the non-functional property he/she is currently working on. We have introduced dedicated UML

views covering scheduling [7], power consumption [5], and thermal behavior [8] which are used for corresponding analyses.

Figure 1 and Figure 2 show a Scheduling Analysis View. In the first figure the structure of the system is described. It contains two CPUs (marked with the <<saExecHost>> stereotype) and a bus system (marked with the <<saCommHost>> stereotype). The lower row of classes represents physical hosts which are connected to schedulable resources (upper row) by the <<allocated>> stereotype. Schedulable resources contain tasks or communication tasks that are executed on the corresponding physical host (methods marked with the <<saExecStep>> or <<saCommStep>> stereotype).



Figure 1: Example of a Scheduling Analysis View

Figure 2 shows a workload situation. Workload situations describe the dependencies between tasks using activity diagrams. The example in Figure 2 defines that task *getData()* has to be finished before the communication task *send()* can be started.



Figure 2: Example of a behavior description of a Scheduling Analysis View

The Scheduling Analysis View does not necessarily only cover software task or busses/CPUs as resources. It is possible to define mechanical parts as resources on which mechanical tasks can be executed, too. No additional extension of the UML profile or the Scheduling Analysis View is necessary [19]. The Power Consumption Analysis View and the Thermal Analysis View work in a similar way. The diagrams have the same structure; only the stereotypes are different. However, all these diagrams contain tasks, resources, their connections, and their parameters. To perform the analyses based on the described views, model transformations are necessary. In case of the scheduling analysis, a transformation to a specialized tool (SymTA/S [9]) was realized [6]. For the thermal and the power consumption analysis, new algorithms were developed [5, 8].

Additionally, a Cost Analysis View is introduced to consider the financial factor (Figure 3). The structure is similar to the views defined above and contains values describing the system's costs. The difference is the set of stereotypes used to annotate the model. For each hardware and software part of the system, the prime costs and the integration costs are captured. Alike for the other views presented, an algorithm for the cost analysis was developed. A related approach was presented by Axelsson in [3], where UML models are used to model the

system structure. These models are enriched with cost information and total costs are calculated taking into account uncertainties, but are too fine-grained and therefore not applicable in out setting.





Apart from these technical issues SysML/UML modeling is also applicable to support process management. A product development process consists of various tasks that have inputs and outputs. Inputs and outputs of development tasks can be CAD parts, parameters or the like. Some tasks need the output of other tasks as an input. Tasks can be made up of subtasks. Using SysML and the systems engineering philosophy a complex development process can be developed from abstract to detail (Figure 4). Every task can be described as an activity with input and output pins. Activities are related to each other with control and object flows, whereas CAD parts etc. use object flows and pins. In addition, pins can be related to blocks, i.e., a CAD part is the output of a development task and another (detailed) model of the according component that is modeled as a block in the other partial model. This gives us the possibility to find domain spreading iteration cycles in order to newly structure the process according to new conditions/restrictions appearing during development, and to analyze and optimize the total system with formal methods and tools, all basing on the same modeling language.



Figure 4: Simplified view on the development process and subprocess kinematic synthesis

The different tasks in this development process and the way they will be executed are defined based on the agreed time and costs during the planning phase. Nevertheless most of the tasks can be executed on more than one way, having a different time and cost for each way (e.g., an additional engineer can support a design task to come up with an earlier result, in this case a headcount needs to be added). There are tasks like "production of the parts" were different execution ways (production techniques) will affect not only time and cost but also other requirements like "number of parts that can be produced". In the planning phase these tasks are defined based on experience, but during the execution phase it is difficult to visualize the consequences of a task delay and even more difficult to calculate the best possible process adaptation that can maintain the milestones with the minimum cost and the less influence on other requirements.

3. Evaluation Analysis Results

Our approach addresses the combination of analysis results from different disciplines to ensure the fulfillment of the overall goals and to exploit the optimization potential (e.g., regarding costs). This has significant influence on the decision procedures. Thus, our approach encompasses the analysis of complex relations among goals to support the selection of an advanced system set-up. Therefore, we consider different viewpoints to produce a simple and consistent knowledge base, use domain crossing analysis tools, and create a systematic decision support for a holistic system optimization.

At the very start of the development process, it is necessary to identify risks for the development in terms of conflicting targets, since these hamper the optimization procedure and influence design decisions. The existing conflict analysis tool [15] evaluates the system on the basis of traces that were modeled between requirements. We assume that an expert knows conflicting and supporting relations between the defined requirements in his/her domain basing on experience or physical context. In addition, the expert knows some relations to requirements from other domains because of experience. As requirements are allocated to blocks we are able to write own sets of requirements for each block or to show the allocations in a matrix. This gives the designer a support to review the multidisciplinary set of requirements of one component and find possible conflicts or redundancies. Furthermore, requirements are related to use cases. Thence, a set of requirements for a special use case can be generated and analyzed. Now that all requirements are formulated and first relations among requirements are modeled, the conflict analysis starts. The tool is able to find conflicts that follow from other conflicts: If A is in conflict with B and B supports C then there is a chance that A is in conflict to C. However, this approach provides just possible new goal conflicts [15]. If this conflict is really existent, it has to be decided by the experts in discussing the issue.

The conflict analysis up to now has the drawback of generating only a list of hints where an intelligent developer could find a conflict. For large products, this list grows to an enormous size and contains a vast amount of false positives which make the retrieval of real conflicts a time consuming task. Another drawback is that the detection of conflicts relies on the modeling of problematic correlations in the system model. For the detection of conflicts not captured in the model, especially if these range across the borders of disciplines, further effort is necessary. Moreover, the designer could be assisted if suggestions were made how to resolve a conflict.

For advanced conflict detection, the SysML model is analyzed more precisely. Since relations between requirements and parts of the product are represented in the model, these can be used to detect more complicated conflicts even if the conflict itself was not explicitly modeled. Though intelligent developers are concerned with predicting conflicts, many may be missed due to the systems' complexity. Thus, we automatize this task by examining the graph structure of the SysML model. For each element with stereotype <
block>> representing a concrete part of the final product, the dependent requirements and targets are collected such that interdependencies become apparent. These can be reported in a clearly arranged illustration depending on the developers' background (e.g., domain, experience) and automatically forwarded to the developer including the problematic component and contact persons to discuss the problematic aspects.

Another possibility to detect conflicts not explicitly modeled is the automatic use of experience from earlier related products. At first, only generic patterns are useful to identify problematic relations between requirements. Most of these patterns depend on physical or technical dependencies which can be predefined. An example for

such a pattern is the matter of fact that a faster CPU may consume more energy and produces more heat than a slower CPU and has higher costs. If now requirements exist concerning the same part of the system stating that the CPU must not exceed a certain temperature and at the same time a determined throughput is claimed, a conflict is likely to exist even without any further knowledge about the future system. The more projects of a similar variety are conducted, the more specialist knowledge can be accumulated and reused such that the conflict detection becomes better. Combined with the analysis of the model's structure introduced above, this technique reduces false positives and recognizes more conflicts. However, expert knowledge is quite unerring if projects are very similar. If projects vary too much or new technologies are introduced, "expert knowledge" could give false hints and hinder innovation. For this reason, a classification will be developed to decide what knowledge from earlier projects can be transferred to a new project.

After conflicts have been considered in this way, the optimization can start with the objective of designing a system fulfilling the requirements as best as possible while at the same time being optimal regarding the optimization criteria. An overview of our supported development procedure is given in Figure 5. The figure shows the general process of iteratively optimizing a system in order to ensure both the fulfillment of requirements and a preferably optimal setup regarding the given targets. The process is implemented in a loop consisting of analyses as introduced in Section 2 and a chain of further activities to improve the system design. This loop is carried out until an adequate system design is found.



Figure 5: Supported development procedure

The first step in this procedure is the analysis of several system properties as described in Section 2. If the resulting values do not satisfy the claimed requirements or further optimization is necessary, a list of possible activities is created. This is done as follows: The SysML diagram is reconsidered and the components violating the requirements are identified. For these components, the reason for the violation has to be found, which can either be done automatically based on technical correlations or manually depending on expert knowledge. As for conflict recognition, a learning algorithm is used which evolves also in this case such that more and more critical reasons for missed requirements can be identified automatically. As a result, either changes in the single components are proposed or even the replacement of parts by other parts fulfilling the requirements better. To decide on the right conflict solving method the conflict type is important. We distinguish logical, physical, technical, technological, and economic conflicts. These conflicts can be solved by variation on the functional (e.g., function integration [20], redundancies, changing order) or design level (e.g., number of parts, degree of freedom). Some conflicts can just be solved by consolidation strategies (e.g., compromise, configuration). Technical conflicts can often be solved be considering the "innovation strategies" of TRIZ [1], although lots of experience is necessary to adopt the strategies to a concrete development problem. To assist the developer in using these strategies they are integrated into SysML. To do so, first the requirements affected by a characteristic will be specified with a stereotype from a list of possible characteristics. On the conflicting requirements the algorithm reads the stereotype and reports the innovation principles that support the given conflict, displayed in a matrix.

The next step is the ordering of optimization actions. The generated list of actions is possibly large and may additionally contain conflicting activities. Furthermore, their influence on the targets' fulfillment varies and new problems are likely to be introduced, too. Thus, an analysis of the single actions' impacts on the system and of conflicts with other actions has to be done. Therefore, the relations between requirements are used again to detect which other requirements are affected if an action is taken. Another important constituent of this step is the impact of changes on factors as the total costs or the development process itself. Using this, an ordering is introduced to find an adequate tradeoff between, e.g., higher costs for better components or longer development time due to improved algorithms working better with poor hardware. The output of this analysis is weighted based on these data such that a prioritized list of actions exists from which the designer chooses the required actions. Due to this ordering, assistance is provided for the designer to select the most effective actions for optimization.

The optimization itself consists of different adaptations of parts of the system and depends on the specific setting. Possible optimizations are, e.g., the use of different mechanical or electronic parts, the application of other scheduling algorithms, or changes in the tasks' program code.

4. Case Study

In this section we demonstrate the presented approach on a case study where we survey the development of a wood-plastic composites-scanner (see Figure 6). We show how the requirements model and the analysis views can be used for specification and validation of the non-functional requirements.



Figure 6: The wood-plastic composite scanner

Wood-plastic composites (WPCs) are composite materials made of wood fiber/flour and plastic. The WPC scanner is a device for continuous detection of structural defects in WPC shelves, operated at the production line after a cooling line. The measurement is done by ultra-sonic pulses that are sent by the shelf. With a mobile arm (lever), the ultrasonic transmitter and receiver are moved over the shelf to scan the entire surface.

The embedded device is situated next to the production line in a box. There are a number of different nonfunctional requirements:

• Scheduling/performance requirements: The timing is very important. The ultra-sonic signal can only be send every 5 ms. Otherwise, the signals will influence each other and make the measurements incorrect.

However, the measured values have to be analyzed in time to set the alarm if a WPC is damaged. Here, a simple and a more complex algorithm for failure detection are available.

- A thermal requirement is that the temperature of the used CPU must not be higher than 100 °C as the CPU is not able to work properly at this temperature. The CPU has a thermal safe mode, where it shuts down if 100 °C are reached. However, in such a case the whole production line has to be stopped which is very costly.
- Power consumption and production cost requirements are not specially defined. The overall requirement here is to optimize these properties by fulfilling the other requirements.
- The acceleration of the lever should be high to allow a uniform distribution of measuring points.
- The lever should move with a high speed to allow a good coverage of measuring points.
- The oscillations of the lever should be as low as possible, because measuring while the lever is oscillating would cause undesired reflections. These reflections would hamper the analysis of the WPC.
- Production costs of all subsystems and development costs should be as low as possible and the projected start of production (SOP) should be kept.



Figure 7: Requirements diagram of the main requirements of the main WPC scanner subsystems

The requirements of the system, all subsystems, and the project requirements are modeled in a SysML-Model (Figure 7). Requirements are allocated to their subsystems (block) and satisfy-relations are modeled to the superior targets. In addition, a use case "detection process" refines the requirements and helps to find the right values for a quantitative specification. Relations between requirements are modeled as traces, e.g., a good scheduling traces to high acceleration. If the kinematic system is far too slow for the data processing system, however, an extremely good scheduling would cause high costs but no visible benefits in time saving.

The components of the system are displayed in a block definition diagram (Figure 8). Main subsystems are kinematic system, rack and data processing system. The kinematic system consists of the motor carrier that is made up out the mobile arm, i.e., upper, middle and lower lever. Middle lever has a geometric interface to the ultrasonic transmitter and upper and lower lever to upper and lower receiver of the detection system.

Carrying out the conflict analysis shows us that a high acceleration generates oscillations of the lever which is bad for the realization of a good scheduling. One conflict solving strategy could be to realize more structural damping by bringing more mass into the lever. However, this would corrupt the good acceleration. Using the separation strategy additional dampers could improve the oscillation decay but would also increase production costs. Another strategy is to improve the detection algorithm in such a way that it will work during acceleration. However, this could lead to a scheduling problem. A detailed analysis is necessary to support the developers in the decision on what strategy is the most promising (see procedure in Section 3).



Figure 8: Block definition diagram of the main WPC scanner subsystems

After an analysis of all requirements using development models in an early design stage, the acceleration is found to be a problem. Because of the high acceleration and the additional effort the detection algorithm has to handle, the deadlines of the software tasks are missed. Consequently, some measurement values are not considered and defects of the WPC may not be found. One action to solve this problem is to use a higher DVS (Dynamic Voltage Scaling) mode of the system's CPU (change the frequency; consequently, tasks are executed faster). The impact of this action is a changed scheduling. Additionally, the power consumption and the temperature of the CPU are different. After a rating of this action and a comparison with other actions, it is decided to use the approach of changing the DVS mode.

In the next analysis of the power consumption, the scheduling, and the thermal behavior, the maximum temperature of the CPU is found too high (compared with the upper bound temperature of 100 °C). The scheduling and the power consumption analysis results are satisfying with respect to the requirements. However, as there are negative analysis results, the procedure's next step is to observe possible actions. One possible action is to use a better cooling. This action has an impact on the temperature behavior as well as on the production costs. After this decision is made, the procedure is started again.

The two analyses that have to be repeated are the temperature analysis and the production cost analysis. An excerpt of the Cost Analysis View is illustrated in Figure 9. Besides the CPU ("ControlCPU"), there is a new element in the system description: the "Cooler", a new passive cooler necessary to lower the maximum temperature. The cooler's price is relative low. However, the cost for the integration was considered higher.

After the integration of the cooler, all analyses are successful and the WPC scanner is developed successfully.





5. Conclusion

In this paper, we have presented an integrated approach covering the complete development process from requirements up to the final system design. We have proposed a generic procedure to analyze requirements and, in case of negative analysis results, give proposed actions and their impact. Afterwards, the proposed actions are be rated and executed. Based on the executed action and their impact, the procedure shows the developer which analyses have to be repeated.

Future work is to optimize the development process by using scheduling analysis methodologies for increasing the productivity. A scheduling approach can help to prioritize development tasks and to give feedback about the meeting of deadlines. If things change in the development process (e.g., the development of a subpart is delayed), the use of modern scheduling approaches can help to change the order of development tasks to find the new optimum. Thus, the difficulty of visualizing the consequences of a task delay during the execution phase can be addressed in order to calculate the best possible process adaptation that can maintain time, cost and priorities.

Moreover, the current model based requirement management approach can be enhanced by explicitly modeling the way requirements are satisfied by components. The satisfaction of a requirement is often allocated to a component, but it is not the block itself that satisfies the requirements. Rather, the satisfaction arises from the functions which the block fulfills. It could be argued that the functions are just further specifications of the requirements, and as objects they can be categorized as such. Nevertheless, when keeping the functions as a different category of requirements the system optimization can be further supported. A direct application is the visualization of how different requirements and components meet in a single function; this can lead to functional integration opportunities, or functional splitting to avoid conflicts.

6. Acknowledgment

The authors want to thank NetCo Professional Services GmbH for the possibility to evaluate the presented approach on the case study.

7. Citations

[1] Altschuller, G.S. (1984) Erfinden: Wege zur Lösung technischer Probleme. VEB Verlag Technik, Berlin.
[2] Arpinen, T.; Salminen, E.; Hännikäinen, T.; Hännikäinen, M. (2011) MARTE Profile Extension for Modeling Dynamic Power Management of Embedded Systems. Journal of Systems Architecture, In Press, Corrected Proof.
[3] Axelsson, J. (2006) Cost Models with Explicit Uncertainties for Electronic Architecture Trade-off and Risk Analysis, In Proceedings of the 16th International Symposium of the International Council on Systems Engineering (INCOSE).

[4] Daenzer, W.F.; Huber, F. (2002) Systems Engineering – Methodik und Praxis. Verlag industrielle Organisation.

[5] Hagner, M.; Aniculaesei, A.; Goltz, U. (2011) UML-based Analysis of Power Consumption for real-time Embedded Systems. In 8th IEEE International Conference on Embedded Software and Systems (IEEE ICESS-11).

[6] Hagner, M.; Goltz, U. (2010) Integration of Scheduling Analysis into UML based Development Processes through Model Transformation. In 5th International Workshop on Real Time Software (RTS'10) at IMCSIT'10.

[7] Hagner, M.; Huhn, M. (2008) *Tool support for a Scheduling Analysis View*. In Design, Automation and Test in Europe (DATE 08).

[8] Hagner, M.; Kolatzki, S. (2012) *UML-based Thermal Analysis of real-time Embedded Systems*. In Power, Energy, and Temperature Aware Real-Time Systems, 33rd IEEE Real-Time Systems Symposium.

[9] Henia, R.; Hamann, A.; Jersak, M.; Racu, R.; Richter, K.; Ernst, R. (2005) *System Level Performance Analysis* - *the SymTA/S approach*. IEEE Proc. Computers and Digital Techniques, 152(2).

[10] Kuchenbuch, K.; Nehuis, F.; Stechert, C.; Vietor, T. (2012) *Analysis of Goal Conflicts in the Vehicle Layout with respect to different Vehicle Environments*. In: Proceedings of the 21st Aachen Colloquium Automobile and Engine Technology 2012, Volume 2

[11] OMG Object Management Group. (2009) UML profile for modeling and anal- ysis of real-time and embedded systems (MARTE).

[12] Object Management Group (2012) *Systems Modeling Language*. http://www.omgsysml.org/, last download 14.03.2013, 2013.

[13] Sanchez Ruelas, J.G.; Stechert, C; Vietor, T; Schindler, W. (2012) *Gestión de Requerimientos para el Desarrollo de Nuevas Arquitecturas de Vehículos Electricos*. In 7º Congreso Iberoamericano de Innovación Tecnológica y Desarrollo de Productos.

[14] Shorin, D.; Zimmermann, A.; Maciel, P. (2012) *Transforming UML State Machines into Stochastic Petri Nets for Energy Consumption Estimation of Embedded Systems*. The Second IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT 2012).

[15] Stechert, C. (2010) Modellierung komplexer Anforderungen. Verlag Dr.-Hut, München.

[16] Stechert, C.; Franke, H.-J. (2009) *Managing Requirements as the Core of Multi-Disciplinary Product Development*. In CIRP Journal of Manufacturing Science and Technology 1(3), Elsevier B.V., Amsterdam, pp. 153-158.

[17] Stechert, C.; Franke, H.-J. (2009) *Requirements Models for Collaborative Product Development*. In Proc. of 19th CIRP Design Conference, pp. 24-31.

[18] Stechert, C.; Franke, H.-J.; Vietor, T. (2010) *Knowledge-Based Design Principles and Tools for Parallel Robot*. In Robotic Systems for Handling and Assembly. STAR 67, Springer Verlag, Berlin.

[19] Stechert, C.; Hagner, M.; Vietor, T.; Goltz, U. (2011) *Requirements Management From An Interdisciplinary Point Of View.* In Proceedings of IASDR2011.

[20] Ziebart, J.R. (2012) Ein konstruktionsmethodischer Ansatz zur Funktionsintegration. Verlag Dr.-Hut, München.