Empirical Validation of a Requirements Engineering Process Guide

Jörg Leuser, Nicolas Porta
Group Research & Advanced Engineering, Daimler AG
P.O. Box 2360, 89013 Ulm, Germany [*joerg.leuser, nicolas.porta*]*@daimler.com*

Armin Bolz, Alexander Raschke
Institute of Software Engineering and Compiler Construction, Ulm University
89069 Ulm, Germany
*[armin.bolz, alexander.raschke]**@uni-ulm.de*

Abstract

Current requirements engineering (RE) practice in industry lacks the availability of an integrated guide that defines and supports the requirements engineering part of embedded system development. Therefore, the RE part of a development project is not as structured and defined as desired. This paper presents results of an experiment validating a RE guide developed by a joint research project between industry and academia. In the experiment carried out for 14 weeks, embedded development projects using the guide were compared to projects following a state-of-the-practice process. The results suggest the guide to be suitable for mastering the RE part of an embedded system development.

Keywords: RE process, systems engineering, embedded systems, process validation, multi-stage validation, experiment, REMsES

1. INTRODUCTION

Requirements engineering (RE) processes for embedded systems in industry are not as systematic as desired. Most companies have their own approach to RE. This is due to the fact that there is no established, complete, integrated, and scalable RE approach for developing embedded systems which is also well documented for end users. Many guides are intended for RE specialists – but in industry, systems are often specified by developers as the domain experts and not RE specialists. The ongoing research project REMsES¹ has the goal to deliver an integrated, suitable RE guide which describes harmonized methods, artifacts and process steps for embedded systems RE. RE for embedded systems has to address some points that are not as important as in interactive systems RE: real-time requirements, and hard resource limitations. In this paper, we describe an experiment used as a validation step of the REMsES guide, later on referred to simply as 'guide'. In the experiment, an interim version of the guide was tested for its suitability and practicability in specifying embedded systems. The use of the interim version allows incorporating the findings of the experiment into the final guide. The experiment was conducted in cooperation between Daimler AG and Ulm University and took place at the university. During the experiment of 14 weeks, the groups of participants specified embedded systems referring either to a state-of-the-practice process or a process documented in the guide.

Contribution: For *industry*, we present evidence that the process and methods described in the REMsES guide are usable in an embedded context. For the *research community* we introduce a multi-stage validation approach to validate processes and their accompanying documentation.

Outline: In Sect. 2 we give an overview over the REMsES project whose guide was validated, the conducted experiment is described in Sect. 3. In Sect. 4 we present the results and how they affect the further validation of the guide.

¹This research was partly funded by the German Federal Ministry of Education and Research (BMBF), grant 01 IS F06 D.

2. THE REMSES PROJECT

The REMsES research project brings together academia and industry. The partners from industry are Daimler AG and Robert Bosch GmbH, the partners from academia are Technische Universität München and University of Duisburg-Essen.

The project's goal is to provide a suitable guide for RE of embedded systems. By the term 'suitable guide', we understand a guide effective for the intended users and their tasks. The intended users of the guide are developers. Therefore, the descriptions of the elements have to be well understandable by non RE experts. A further requirement for the guide is the need for 'hands on' guidance to be useful in the daily work of the users. To achieve this, this new guide tries to combine already established methods with newly developed or adapted methods (see for example [9], [7]) into an integrated and guided approach to RE.

This paper does not report the contents of the guide itself (for an overview see [8], [11]) although the ideas behind the guide and therefore an overview over the contents will be given.

Creating the guide itself is only the first step. The second step is its validation in order to convince industry that the guide is suitable for their needs. The validation approach is described in Sect. 2.2.

2.1. The Guide

The guide was developed with different organizations and projects in mind. As organizations and development projects are very different from each other, the guide offers a master-set of artifacts and process steps which is meant to be tailored to meet the needs of the distinct projects. The tailoring is facilitated through the explicitly defined relationships between the artifacts and process steps as well as through the chosen representation of the guide. Realistic examples support the understandability. Both content and chosen presentation will be described in the following sections. The experiment described in Sect. 3 aims to validate both content and presentation.

Content: The two main components of the guide are artifacts and accompanying process steps. In simple terms, artifacts are named and defined information containers which together form a complete specification of the system to be built. The process steps form a process which guides the creation of the artifacts. Additionally, possible integration of the artifacts and process steps into typical surrounding processes (e.g. change management, and original equipment manufacturer/supplier coordination) are described in the guide to facilitate its use in industry.

The requirements engineering methods which are part of the guide include for example goal modeling, scenario/use case modeling, and behavior modeling. Although the methods are not completely new, the goal is to adapt the approaches and describe them within the guide so they are well understandable and usable for the intended users - the developers - and suitable for the intended context of embedded systems.

To manage the complexity of embedded system development projects, the introduced artifacts are classified into three different abstraction layers (rows) and content categories (columns) (see Fig. 1). Using this classification leads to a separation of concerns between the different artifacts. A more in-depth view of the proposed abstraction layers and a comparison to other proposed structures (e.g. to the project mobilSoft [12]) can be found in [10].

FIGURE 1: Abstraction layers as defined by REMsES

While the artifacts describe what information is needed and how it should be documented, the accompanying process steps describe how the artifacts can be created, derived, or changed.

FIGURE 2: Screenshot of the exported guide

The description of the process steps includes hints about where to find the needed information, for example from certain stakeholders or other artifacts. Relationships between the different artifacts are defined both explicitly (e.g. 'stakeholder characterization' refines 'stakeholder list') and implicitly by input-output relationships of process steps.

Presentation: The guide is created using the Eclipse Process Framework² (EPF). The standard export format of the tool is a website (for the presentation structure, see Fig. 2) which contains the content of the guide including the overall process. Generating websites has the advantage of being easily deployable, easy to use (e.g. through the full-text search), to update, and to integrate in existing tools (e.g. in form of hyperlinks). For the process specialists, EPF provides collaboration possibilities, allows the reuse of already existing process fragments, and allows to easily tailor the guide and its process to different development projects. We see an advantage in having a tailored guide that only contains the information needed by a development team over having a monolithic process descriptions.

2.2. Validation Approach

The project's goal to deliver a suitable guide requires ample validation. While aspects like understandability of the guide's descriptive texts could be assessed by reviews performed by the industrial and academic partners, a more thorough approach was needed to demonstrate the suitability and practicality of the guide to industry. Contrary to process improvement approaches, the goal is not to improve just a single instance of the process and its documentation but to improve them in a more general way. To validate the different aspects of the guide, a multi-stage validation approach was planned. One of those steps is the described experiment.

The validation stages: As a first check, the guide's authors tested their methods for suitability against *sample specifications* of embedded systems provided by industry. Those specifications are part of the guide as examples.

A second step in the validation approach were ample *reviews*. The reviews focused on consistency, readability, and understandability of the defined artifacts and process steps.

The third step is the *experiment* which for the first time tested the proposed guide in a long-term realistic setting which is described in Sect. 3. For the experiment, the guide was tailored as it would be done for a real development project. Currently the results are used to improve the

²www.eclipse.org/epf

interim version of the guide. With the improved version, three more validation steps are in progress or planned:

Parts of an existing *real-world specification* of an embedded system are being *re-specified* using REMsES methods and following the process steps described in the guide. The resulting specification will be compared to the originally available one.

As a next step, *industrial workshops* with developers – the intended users of the guide – are planned. The developers will be trained and try to solve common tasks from their daily work using the guide. The feedback given by the participants will be used to further improve the guide.

Finally, the introduction of the guide to *real world pilot projects* in industry is intended. The industry partners plan to cover different parts of the guide. Those projects will be supported by the REMsES members and lessons will be learned and incorporated into the guide.

While obviously each of the validation steps separately cannot cover all aspects of the guide, the combined steps together cover the most relevant aspects of the guide. The advantage of splitting the validation into different validation-stages is to allow single steps to be more focused and to use early validation results to improve the guide while it is still being developed. In the following section we present the experimental part of the multi-stage validation.

3. EXPERIMENTAL APPROACH

As described in Sect. 2.2, the experiment aims to explore the suitability of the guide's methods as a precursor for further validation steps. To analyze the effects of the guide's process and methods, the most obvious approach would be to clone a system development project, to control the surrounding variables, and to measure differences to another process.

In an industrial context, however, this is not feasible (or at least very difficult) for two main reasons: Firstly, the cloning of a process for research purposes only is too expensive. Secondly, the surrounding variables are very difficult to control, due to organizational change and changing requirements of the project's stakeholders.

Contrary to the industrial context, the cloning approach is feasible in academic research: Here we do have the possibility to run the same project using different processes in a controlled experiment (see [1] for detailed insight into the experimental approach taken in the cooperation between Daimler AG and Ulm University). Houdek [4] demonstrated the practical value of such software engineering experiments for a target environment.

The participants of the experiment were students of Ulm University. They were free to choose between the course with the experiment and similar unmarked courses. The subjects specified and implemented embedded systems from the automotive domain either using the guide process or a state-of-the-practice RE process (SotP). The details are described in Sect. 3.2.

3.1. Hypotheses

The main goal of the experiment was to show the suitability of the methods described in the guide. We based the concept of suitability on the fulfillment of the following characteristics, formulated as two main hypotheses (MH):

- **MH1:** The process steps and artifacts in the guide are *comprehensible*.
- **MH2:** The process steps *guide* the development of the required artifacts *well*.

Additionally to the main goal of the experiment, an exploratory comparison to a SotP was included. Although further validation steps will focus on such comparisons (see Sect. 2.2), the experiment can give first indications towards which outcome to expect. To emphasize that this question is of lesser importance here, the following hypotheses are labeled as secondary hypotheses (SH):

- **SH1:** The usage of the guide improves the *specification quality* compared to the SotP.
- **SH2:** The *specification effort* with the guide process is higher than the specification effort with the SotP.
- **SH3:** The *implementation quality* of a project following the guide is better than the implementation quality of a project following SotP.
- **SH4:** The better the *specification quality* the less *implementation effort* is required.

To allow drawing conclusions about the stated hypotheses, metrics were used as well as questionnaires which were answered by the students. The conclusions about the main hypotheses were largely based on the questionnaires' results while the raised metrics contributed mainly to the secondary hypotheses. More details can be found in Sect. 4.

On the metrics' side we focused on defects and effort concerning the different development phases, student groups, and artifacts. Table 1 gives an example for the measurement of the specifications quality.

Quality of the specification:	
Object: Specification	Purpose: Comparison (SH4)
Metrics:	
Defects: #classified defects found in specification review (SotP and guideline)	
Defect density: #critical defects / #document pages	
Overall quality: Expert evaluation of the specification according to standardized checklist	
Variation factors:	
Granularity of the defect documentation	

TABLE 1: Example of specification quality metrics

3.2. Experiment setup

It is a general issue that the evaluation of one's own developed process often turns out to be better than another one. One reason for this effect is probably the deeper knowledge of the self developed process. To avoid this effect, Ulm University which was in no way involved in the development of the guide, was asked to conduct the experiment.

The idea for the experimental setting is to develop the same systems twice: Some groups use the REMsES process and its guide and some other groups build the systems following the SotP process. Since the main focus of the experiment is to validate the guide and not the comparison of two RE-processes (see Sect. 3.1), four groups were to use the guide and only two groups were to follow the SotP. Thus, more data points for the validation are obtained.

The original guide process was tailored in a way that students were able to carry out all required steps within 14 weeks (the duration of one semester) but the used artifacts still fit together and build upon each other. Additional limitations were given by the experiment setup. For example dealing with the constraints given by laws made no sense in this set up. Therefore the corresponding steps were removed from the original process. The names of the process steps being part of the tailored process can be seen as labels on the x-axis in Fig. 4.

As our SotP, we tailored a process commonly used at Daimler to the boundary conditions of the experiment. The process is described in rough terms in [3]. As a documentation basis for the specification, we opted for a tailored version of the Volere Template [5]. Like the guide, this established template provides user guidance on how a system has to be specified.

A three day training course was conducted in advance of this experiment to teach the students the methods requested by the approaches and the used tools to balance the different knowledge levels. Besides this measure, teams each consisting out of two students were formed by the supervisors, in order to avoid too big differences in knowledge and motivation. This division was first of all based on a questionnaire on the knowledge of each student at the beginning of the training course and second on the results the students developed during the training. It is also very helpful creating teams to react on possible drop outs. Furthermore, the small team size reduces the impact of group dynamics which can strongly influence the results.

To achieve a better quality of the experiment's results two different embedded systems have to be developed. The systems chosen were a controller for a central locking system of a modern car (called MachZu) and a controller for the light system (including interior light, headlights, etc.) of a car (called Lumiere). These two systems were already used multiple times in prior experiments and proved themselves being of the same complexity, having a comparable number of requirements and leading to comparable specification and implementation effort. Therefore we treated the system not as differentiating factor.

FIGURE 3: Experiment setup (time line not to scale)

The students are provided with a feature list to ensure the comparability of the developed systems at the end (see Fig. 3). Each team specifies one of these systems and at the end of the specification process the complete specification is reviewed by another team.

Based on the corrected and improved specification the original team creates test cases for their system before the specifications are swapped again for the implementation phase (which is conducted to check the quality of the specifications) across the teams. This exchange guarantees that the implementation is built only upon the information contained in the specification artifacts not including the facts not mentioned in the artifacts but probably still existing in mind.

The implementation itself is done in Telelogic Statemate³. This tool offers the possibility to create state charts that can be executed. Additionally, a graphical user interface can be integrated to make simulation and testing easier and more comfortable. Finally, the created systems are tested using the earlier created test cases.

The reason why we conducted the experiment with four guide groups and only two SotP groups is that the main focus of this experiment was not the comparison of two processes but the investigation how comprehensible and helpful the guide is. This circumstance is also reflected in the main hypotheses (see Sect. 3.1).

For any questions concerning the experiment (besides technical ones), each team got so called time tickets corresponding to 30 min consultation time per week. This time ticket system simulated a management-to-developer relationship where management also only has restricted time for questions. With this constraint the students were forced to concentrate their questions in an effective way.

3.3. Threats to validity

For every experiment one important task in the experiment design is dealing with the validity of conclusions drawn. For this reason we identified several threats to validity and defined measures in order to minimize the risks of obtaining incorrect conclusions on the stated hypotheses.

By internal validity we address the extent to which the design and analysis may be comprised by the existence of confounding variables (or other sources). The external validity aims on the extent which the hypotheses capture the objectives of the research and to the extent to which the drawn conclusions can be generalized [6].

One important aspect is the question whether the measurement of a process executed by students allows assumptions on the same process performed by professionals. In our opinion this is the case for the experiment at hand.

At the industrial side, domain but not RE experts develop the specifications. Additionally the students do have sufficient know-how in requirements engineering due to at least one obligatory software engineering project in their earlier studies.

In literature about validation through software engineering experiments Hannay and Jørgenson [2] argue it has to be distinguished between *structural* and *situational* artificiality. While structural

³www.telelogic.com/products/statemate

FIGURE 4: Median rating of process step descriptions

artificiality is the very essence of every experiment (control variables to isolate the outcomes of purposeful interventions) situational artificiality relates to the experimental ingredients like subjects, treatments, settings and materials. At the time situational artificiality is often seen purely as a necessary evil in software engineering experiments - by contrast we share the authors' view that it can be a chance to get better results compared with the attempt of just copying reality.

In conclusion we regard the students' comprehension of the process steps and artifacts as good as those of professionals. Additionally, the situational artificiality opens the chance to work with stronger motivated subjects.

Table 2 gives an overview of further identified threats to validity, the argumentation followed and the measures taken.

Potential threats to validity (internal [I] or external [E])	Measures [M] / argumentation [A]
Unfair comparison between processes and descriptions as Volere is a generic approach which is not adapted to the embedded context (I)	<ul style="list-style-type: none"> The Volere template was adapted and the SotP process was tailored to fit in the given context (M) Stronger guidance through REMsES guideline forms part of the observed object in the experiment (A)
Inadequate skill of test persons (students don't have enough experience in RE) (E)	<ul style="list-style-type: none"> Specifications in automotive context are developed by domain experts (not by RE experts) (A) Students do have experiences in RE due to obligatory development project within their earlier studies (A)
Systems are too small and therefore not realistic (E)	<ul style="list-style-type: none"> Systems are small, but similar problems artificially introduced: <ul style="list-style-type: none"> System was spread in multiple electronic control units (M) Fix simulated system environment used (M) Network-resources restricted (M)
Unequal groups (stronger and weaker groups) endanger the conclusion of the experiment (I)	<ul style="list-style-type: none"> Groups allocation by the advisors after training workshop (M)
Too little groups not realistic (I)	<ul style="list-style-type: none"> Specifications in automotive context are sometimes developed by very little teams (A)
Groups implement mental model instead of the specification (I)	<ul style="list-style-type: none"> Two systems developed: Switch between groups (implement the system of the other group) (M)

TABLE 2: Threats to validity and measures taken

4. RESULTS

The results are based on two different kinds of data sources: data provided directly by the participants and data collected by the supervisors. On one hand we used questionnaires together with review protocols, error protocols and time ticket logs. Those were promptly recorded into spread sheets by the participants during the experiment. On the other hand, we collected metrics from the students' artifacts.

Evidence about suitability of the REMsES guide as described in Sect. 3.1 was gained by questionnaires. The questionnaires had closed questions for each artifact and each process step. For examples see the hypotheses' results later in this section. While in the following we report

only the results of the closed questions, each of those was accompanied by multiple directed open questions to obtain more detailed information. The closed questions base on Likert scales with four options. For analysis purposes the options were encoded from '1' (worst) to '4' (best).

In order to compare the quality of the different specifications and implementations we used two structured checklists based on a rating scheme commonly used in the institute and iteratively adapted over the last years. Each checklist consisted of certain categories such as correctness, consistency, complexity, layout or comprehensibility. Every category itself contained a list of various questions that let us rate the aspects of the given category reaching from '--' up to '++'. For example, the category 'model correctness' of the implementation checklist contains questions like:

- Is the Statemate model executable at first go?
- Is the Statemate model free of non-determinisms?
- Is the Statemate model free of infinite loops?

The category 'model comprehensibility' contains questions like:

- Are the identifiers clearly defined?
- Does the state-layout/grouping contribute to the model-readability?
- Were the (model)elements used consistently to model similar or equal functionalities (concerning subcharts or parallel states)?

We assign a score to each category by aggregating the individual ratings. By weighting these scores we obtain the quality scores of the specification and the implementation.

4.1. Evaluation of hypotheses

Regarding our hypotheses we were able to draw the following conclusions:

MH1: Our first hypothesis stated that the process steps and artifacts are comprehensible. For the 17 artifacts and 21 process steps, the question '*How understandable was the guide section?*' resulted in a median score of 3 which was labeled 'well understandable'. Only the process step 'define system-vision' rated slightly worse (median of 2,5) and the process step 'comparison of local and system wide function group scenarios' rated better (median of 4, labeled 'very well understandable').

MH2: Our second hypothesis stated that the process steps guide the development of the required artifacts well. To evaluate that point we asked the question '*Were you able to create the output artifacts by following the guide's steps?*'. The median answers to this questions are shown in Fig. 4. Except for two outliers all other 19 steps were rated with a median of at least '3'. Furthermore, we asked for occurred obstacles or additionally necessary development steps, but only minor issues were reported. We conclude that the process steps guide the development of the artifacts well.

Since our main hypotheses were supported by the experiment results, we see the guide to be suitable. Therefore the guide is ready for further validation steps.

Due to the small number of groups involved in the experiment it is not possible to employ powerful parametric statistics (e.g. t-Tests) as the required normality checks cannot be performed. Instead we used the non parametric Mann-Whitney-U-Test. Concerning our secondary hypotheses we found the following:

SH1: Our first secondary hypothesis stated that the usage of the guide improves the specification quality compared to the SotP. As described before, a checklist was used to asses the quality of the different specifications. This checklist incorporates the results of the different metrics described in Table 2. The resulting quality scores are shown in Fig. 5. The U-test showed on the significance level of .1 that the specification quality of the REMsES approach is better than the quality of the

SotP approach. We are convinced that replications of the experiment will support our view that this hypothesis can be accepted.

FIGURE 5: Checklist based specification quality scores

FIGURE 6: Distribution of effort

SH2: Our second secondary hypothesis stated that the specification effort when using the guide would be larger than the approach using SotP. As shown in the 'Specification' component in Fig. 6, we found evidence that this is in fact the case, but it is not statistically significant. It is obvious that group 3 needed more time for the specification of their system compared to the other groups. This is due to the fact that they made a critical mistake in their specification which was found during the review and needed fixing. With no conclusive data, this hypothesis should be evaluated again in an industrial context where more surrounding constraints (e.g. needed coordination) influence the specification time more than it was the case in the experiment context.

SH3: Our third secondary hypothesis stated that the implementation quality of a project using the guide would be better than the implementation quality of a project that follows the SotP. The implementations (given as Statemate models) were rated by the use of a checklist as described above which led to a quality score. The different categories showed only minor differences. The models developed with the guide had marginally better grades in the category 'comprehensibility' whereas the models developed with the SotP had better grades in the category 'Statemate architecture'. We used the Mann-Whitney-U-Test on the overall scores and could not detect any significant differences. With the given results we are not able to corroborate the hypothesis SH3.

SH4: Our fourth secondary hypothesis stated that the better the specification quality, the less implementation effort is required. We found a correlation of -0.68 between specification quality (see Fig. 5) and implementation effort (see Fig. 6, 'Implementation' component). A statistical significance on the .05 level can only be stated when the outlying group 3 is omitted. In this case the correlation is -0.89. In our experiment we agree with this hypothesis.

4.2. Further findings

We limited ourselves to a number of hypotheses but were aware that other useful data could be collected during the experiment. To facilitate the collection of this additional data we actively gave the participants many opportunities to report their experiences. One reported problem was error fixing in REMsES as it leads to multi-layered specifications. Errors that were made in artifacts on the first abstraction layer but found in later phases often lead to changes in multiple artifacts. Although the guide defines the connection between different kinds of artifacts, more traceability information is needed for better and more efficient impact analysis. The guide points to possibly affected artifacts but the user still has to find the correct ones. For such a case, explicit links into the artifacts and between different artifacts would facilitate such a task. This is especially true as the specifications generated during the experiment are still relatively small compared with real world specifications. However, to efficiently do this, a sophisticated tool support is required.

A phenomenon not anticipated was the approach some of the implementing groups took to handle specification errors on the lower functional level (see Fig. 1) that were not found during the review. If for example the detailed function group scenarios were erroneous, the groups ignored them and implemented the system level use cases instead. This effect occurred only in groups using the guide as the resulting specifications were more detailed.

5. CONCLUSION

The experiment's results indicate that the guide edited by the REMsES project is suitable for specifying embedded systems. The positive results of the experiment (e.g. the observed improved specification quality) provide confidence in industry to be able to execute the described additional validation steps successfully. A final conclusion about the suitability of the guide should only be made after those further steps.

We conclude that it works well to let the participants record a time log during the experiments. However, to be able to explain unaccountable time amounts (e.g. the longer time needed by group 3), experiment advisors should additionally record problems identified during the experiment. The feedback of the participants confirms that having realistic and ample examples for artifacts and process steps available in the guide simplifies the application of methods as many questions concerning the methods could be answered without asking the advisors.

Outlook: While the experiment showed that in general the guide is suitable for supporting the specification of embedded systems, it also revealed some weaknesses. Those weaknesses are currently being addressed to allow proceeding to the next validation steps.

REFERENCES

- [1] Gessenharter, D., Merten, A.-M., Raschke, A. & Porta, N. F. (2007), Experiences on using software experiments in the validation of industrial research questions, in 'Proceedings of the IWSM - Mensura', pp. 279 – 288.
- [2] Hannay, J. & Jørgensen, M. (2008), 'The Role of Deliberate Artificial Design Elements in Software Engineering Experiments', *IEEE Transactions on software engineering* pp. 242–259.
- [3] Heumesser, N. & Houdek, F. (2004), 'Experiences in managing an automotive requirements engineering process', *IEEE International Conference on Requirements Engineering* pp. 322–327.
- [4] Houdek, F. (2003), *Lecture notes on empirical software engineering*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, chapter External experiments: a workable paradigm for collaboration between industry and academia, pp. 133–166.
- [5] James Robertson, Suzanne Robertson (2008), 'Volere Requirements Resources'. www.volere.co.uk.
- [6] Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El & Rosenberg, J. (2002), 'Preliminary guidelines for empirical research in software engineering', *IEEE Transactions on Software Engineering* 28(8), 721–734.
- [7] Penzenstadler, B. & Koss, D. (2008), High confidence software reuse in large systems, in 'Proceedings of International conference on Software Reuse (ICRE'08)', Lecture Notes in Computer Science, Springer, pp. 52–63.
- [8] Penzenstadler, B., Sikora, E. & Pohl, K. (2009), 'Guiding requirements modelling in the embedded systems domain with an artefact reference model', to appear in *RefsQ'09*.
- [9] Pohl, K. & Sikora, E. (2007), COSMOD-RE: Supporting the co-design of requirements and architectural artifacts, in 'Proceedings of the 15th IEEE International Requirements Engineering Conference (RE 2007)', pp. 258–261.
- [10] REMsES Konsortium (2007), 'D-2.2: Grobes Produktmodell inklusive der Abstraktionsebenen zur Strukturierung und Modellierung von Anforderungen'. www.remses.org.
- [11] REMsES Konsortium (2008), 'www.remses.org', Website, last visited 12/12/2008.
- [12] Wild, D., Fleischmann, A., Hartmann, J., Pfaller, C., Rappl, M. & Rittmann, S. (2006), 'An Architecture-Centric Approach towards the Construction of Dependable Automotive Software', *Proceedings of the SAE 2006 World Congress*.