A Formal Systems Engineering Approach in Practice: An Experience Report

Wolfgang Böhm, Maximilian Junker, Andreas Vogelsang
Technische Universität München

Sabine Teufl
fortiss GmbH München

Ralf Pinger, Karsten Rahn
Siemens AG Braunschweig

ABSTRACT
This paper reports on a successful research transfer project executed in collaboration between Siemens AG, fortiss GmbH and Technische Universität München. The goal of the project was to evaluate if the SPES modeling framework (SPES MF), which has recently been developed by an industrial and academic consortium, and which is implemented within the tool AutoFOCUS3, can be directly applied to a real-life, productive, industrial system. To achieve this, we performed a case study, in which we created models for requirements and functionality for a part of a Siemens train automation system. The results indicate that the SPES MF can indeed be beneficially used in this context. Furthermore, by applying such a structured modeling approach, we were able to reveal several issues in the original requirements specifications. In this paper, we report on our experiences in setting up and performing such kind of a collaboration between industry and academia. We discuss the success factors as well as problems that we encountered during the project.

Categories and Subject Descriptors
D.2.2 [Software Engineering]: Design Tools and Techniques

Keywords
Software engineering, model-based development, industry collaboration, case study, AutoFocus3, requirements

1. INTRODUCTION
Seamless model-based development, starting from requirements, promises to increase productivity and improve the quality of the developed systems [4]. The real benefits of the models take effect if they are used throughout the whole development process in a seamless way. For instance, requirements are the inputs for an initial system design and for test case generation. This workflow requires a deep integration of requirements, system design, and tests in an integrated product model.

Recently, the Technische Universität München (TUM) and fortiss GmbH (fortiss) participated in a consortium of more than twenty partners from academia and industry that jointly developed the SPES Modeling Framework [12] (SPES MF) to support seamless engineering of software for embedded systems. The SPES MF is an artifact-oriented and model-based approach, which defines artifact types and their relationships for different engineering concerns and different levels of system granularity. While the SPES MF has been applied to some exemplary artificial systems (see for example [7, 14]), it remained an open question, whether the approach is applicable to a system of realistic size and complexity and if it still fulfills the claimed benefits in this context.

In a collaborative project between TUM, fortiss and the Rail Automation business unit of Siemens AG (Siemens), we applied the SPES MF to a real-world productive system developed by Siemens, asking the research question: "Can the SPES MF, as a scientific approach, together with a mature tool support be applied directly to a real-world productive system?"

In order to answer this question, we set up a case study, where we used the tool AutoFOCUS3 [8, 11] to develop a comprehensive model of the system requirements and functions for a part of a productive train automation system. The application of the SPES MF to the system revealed several points for discussion regarding the original requirements, which were of interest for both the industrial and the academic partners.

In this paper, we report on our experiences with this collaboration. In particular, we describe and evaluate the project setup and the case study that was performed and identify success factors and constraints that we encountered. We also highlight some technical findings that we discovered during the project.

The rest of the paper is structured as follows. We outline the goals and setup of the collaboration in Section 2. In Section 3, we give an overview of the study object, the SPES MF and the results we achieved. In Section 4, we evaluate the collaboration setup from an academic as well as from an industrial point of view, and finally we draw a conclusion and give an outlook in Section 5.

1This work was part of the project SPES_XT, funded by the German federal ministry of education and research.
2http://af3.fortiss.org
2. COLLABORATION

2.1 Goals of the Collaboration

Academic Perspective. The academic partners pursued three main goals with the project.

- Validation of the SPES MF as a Development Method. As stated above, the SPES MF has been applied to several exemplary systems from industry and academia (e.g., [7, 14]). However, evaluating how well the approach performs, when used with the requirements specification of a productive system would be a major step towards introducing the approach into industrial practice.
- Validation of AutoFOCUS3 as Tool Support for the SPES MF. AutoFOCUS3 was developed to enable research on tooling concepts for model-based development as well as on pragmatic issues with using formal verification and synthesis based on semantically rich models. The tool supports modeling in all phases of development from requirements engineering to code generation and deployment together with a wide range of analysis techniques. The collaboration reported in this paper was the first case study, where AutoFOCUS3 has been used to implement the SPES MF. We wanted to evaluate if the modeling and analysis features available in AutoFOCUS3 are adequate for modeling systems of productive size with respect to the SPES MF development method.
- New Research Topics. Situations, where the SPES MF or AutoFOCUS3 do not provide adequate solutions are possible directions for further research. By applying the development method to the industrial case study, we aimed at identifying such future research topics.

Industrial Perspective. As a single-source supplier and system integrator, Siemens combines all the expertise necessary for sustainable solutions in all areas of rail transportation. Thus, system architectures, the use of platforms and the integration of newly developed components are important for implementing different customer requirements, innovating products and increasing the level of automation and automated services for customers. New ways for the description of system architectures, which are able to integrate different views on different levels of abstraction can help to support the integration of new features in existing solutions for rail automation. The main goal of Siemens was to assess the combination of tools and methods provided by the academic partners and to evaluate new concepts in an industrial setting.

2.2 Setup of the Collaboration

In the following, we outline the collaboration setup that we chose to achieve the project goals.

Case Study Based on a Mature Product. We decided to use an already existing mature product as case example instead of a new system that is currently being developed or will be developed in future. Because of the availability of high quality specification documents, we decided to (re-)model a part of an existing product that has already been introduced into the market. Leveraging Siemens’ experience with the system and its development, we expected fruitful discussions about the appropriateness of modeling alternatives.

No Adaptations. For the project, Siemens provided their original specification documents as input. We decided that the modeling should be done strictly based on these input documents. We also decided to neither adapt the input documents to better fit the development method, nor to modify or extend the development method to cope with the specifics of the case study. From this setup we expected answers to the questions:

- Can the available methods be applied successfully and where is further improvement of the methods necessary?
- To which extent can the existing specification documents be used without adaptations as input for the SPES MF development method?

Based on the SPES MF and AutoFOCUS3. It was a requirement of Siemens to demonstrate the advantages of a seamless model-based approach for the development of embedded systems by also providing seamless tool support. We decided to use AutoFOCUS3 for all modeling and analysis tasks as the tool was readily available at TUM/fortiss. As an additional benefit we were able to evaluate AutoFOCUS3 with respect to its capability of supporting the SPES MF development method.

Modeling by Method Experts. We decided that the actual modeling work was done by method and tool experts (i.e. the academic partners) due to pragmatic reasons. We wanted to avoid time consuming introductory seminars for the method and the tool. The necessary domain expertise was introduced in workshops and regular phone conferences. These regular meetings were also used to continuously discuss intermediate results and get feedback from the domain experts. We were aware that this collaboration setup does not allow to evaluate the feasibility of the modeling approach and AutoFOCUS3 in the actual organizational context of Siemens.

3. STUDY

This section describes the case study performed within the project. We start with a description of the examined system and the SPES MF. Afterwards, we discuss the implementation of the study in terms of how the SPES MF was applied to the case. Finally we report on the findings that resulted from the execution of this study.

3.1 Study Object

Case Example. Trainguard MT (TGMT)\(^3\) is an automatic train control system for metros, rapid transit, commuter and light rail systems. It is a communication based train control (CBTC) system with high-resolution train localization and bidirectional continuous data communication between the train and the wayside systems. By providing moving block train separation, optimal usage of the infrastructure is guaranteed. TGMT provides a large number of protection and automation functions for railway operation and uses components on the wayside and on-board the trains.

The TGMT system concept is based on a cyclical exchange of position report telegrams sent from trains to the wayside subsystem and on movement authority telegrams sent from the wayside subsystem to the trains. Telegrams are standardized records that are digitally transmitted and usually

\(^3\)http://sie.ag/1aHfPlJ
used for remote control and for control purposes in the system automation.

One purpose of TGMT is to control and protect passenger transfer at platforms. Therefore, TGMT provides a function to control platform screen doors (PSDs) for the protection of passengers in metro systems. PSDs are installed at the platform and can be implemented as full-height or half-height doors. Figure 1 shows a typical half-height platform screen door installation.

For the realization of PSD control, TGMT has an interface to the wayside doors and to the train doors on-board the train. The opening and closing of train doors and PSDs has to be synchronized. To guarantee passenger safety the following protective mechanisms are part of the PSD function:

- The train doors as well as the PSDs are only allowed to be opened if the train is at standstill.
- The train doors as well as the PSDs are only allowed to be opened on the correct side.
- The PSDs are only allowed to be opened when there is a train in the correct position at the platform (the train doors have to match the related PSDs).
- Only those PSD sections are allowed to be opened that match the train length.
- During passenger transfer (open doors) the train must not move.
- If a PSD at a platform is unintentionally opened, no train is allowed to approach the platform.
- If there is a malfunction of the train doors, the PSDs must not open.

To model the PSD functionality, Siemens provided four documents which were taken directly from the PSD development: a high-level system requirements specification (59 pages), a more detailed system architecture specification (299 pages), a performance specification (57 pages) and a glossary (42 pages). The requirements documented in the system architecture specification were linked to the requirements of the high-level system requirements specification, indicating a notion of refinement. In addition, the system architecture specification contained a partitioning of requirements to wayside components and on-board components. The performance specification provided additional requirements and calculations, for example regarding timing constraints. Finally, the glossary explained most technical terms and abbreviations used in the other documents.

Development Method. In the research project SPES, the SPES Modeling Framework was developed to enable seamless model-based development for embedded systems [12]. The SPES MF focuses on artifacts that are created during the development of embedded systems. The framework does not impose a concrete development process in which the artifacts are created. The SPES MF structures artifacts according to two orthogonal dimensions, the SPES viewpoints and the SPES layers of granularity.

The SPES viewpoints introduce a conceptual framework for architectural descriptions with the goal to reduce complexity by considering only relevant information of the system under development according to one particular development view (cf. IEEE Std. 1471 [9] and its current successor IEEE Std. 42010 [10]). A viewpoint can be characterized as a structured specification that supports the definition of such a view of the system. The specification of a viewpoint consists of the stakeholders’ concerns (e.g. specifying the system architecture) that are addressed by the view together with conventions for creating that view (e.g. the underlying ontology, the ontological relationships to other views, and rules for evaluating the quality of the corresponding views). The SPES MF differentiates between the following four viewpoints:

- The SPES Requirements Viewpoint addresses the structured documentation and analysis of requirements
- The SPES Functional Viewpoint addresses the structured documentation and analysis of system functions and their behavior (Functional Architecture)
- The SPES Logical Viewpoint addresses structured documentation and analysis of the logical solution (Logical Architecture)
- The SPES Technical Viewpoint addresses the structured documentation and analysis of the technical solution.

To further reduce the complexity of the engineering process, the SPES MF introduces layers of granularity, where a coarse-grained engineering problem is decomposed into a number of more fine-grained engineering problems following the principle of divide and conquer, i.e. the composition of the fine-grained solutions is a solution for the coarse-grained engineering problem. Whenever a coarse-grained engineering subject is decomposed into a number of fine-grained engineering subjects, a new layer of granularity is created. Hereby, the system is decomposed into smaller and less complex parts. Since the number of such layers depends on the properties of the individual engineering context of an embedded system, the SPES MF does not define a fixed number of granularity layers.

Tooling. AutoFOCUS3 is an open source research tool developed by the research institute fortiss. The main motivation behind the development of AutoFOCUS3 is to evaluate tooling concepts and pragmatic aspects about model-based development, formal specification and analysis, and design space exploration techniques. It provides a broad range of modeling concepts at different levels of formalization for different phases of development ranging from requirements [13] to platform architecture and deployment. Specifically, it provides support for most of the modeling artifacts described in the SPES MF.

3.2 Study Execution

In the study we modeled the PSD door functionality ac-
Figure 2: Overview over the artifacts that were created, as well as the engineering activities and analyses that were performed.

cording to the SPES MF. We focused on modeling requirements and the functional architecture of the PSD system. An overview over the artifacts that we created is shown in Figure 2.

As described above, the SPES MF introduces the concept of granularity layers, where decomposition is used to define subsystems. In our case study the top granularity level defines the complete PSD function as the system under development, which is decomposed into an on-board and a wayside subsystem. This decomposition was already given by the specifications and therefore we treated this fact as a design constraint during our modeling work. As a consequence, the next level of granularity, consisted of two subsystems defining two engineering paths with the on-board unit and the wayside unit as the systems under development.

In the following, we describe the modeling of the PSD function in AutoFOCUS3.

**Description of the domain.** The domain knowledge of the PSD system is described by a glossary, which includes all major concepts of the domain. For that purpose we analyzed the terms within each requirement. Domain and system terms were identified and, whenever possible, the definitions given by the case study documents were included into the AutoFOCUS3 model.

**Textual Requirements.** After defining the relevant terms of the domain, the requirements were transferred from the case study documents word-by-word into AutoFOCUS3. Each requirement was annotated in the AutoFOCUS3 model with a link to its location in the original source document, thereby linking the original documents to the AutoFOCUS3 models.

In the original documents, requirements from the architecture specification provided links to requirements of the system requirements specification. These links indicated that the requirements of the architecture specification are derived from the more abstract requirements of the system requirements specification. We modeled these relations as refinement links [2] in AutoFOCUS3. Relations between requirements that did not denote refinements were modeled as undirected trace links. For all trace links AutoFOCUS3 offers views to comprehend and navigate the relations between requirements.

**Formalization of Requirements.** The formalization of requirements was done in two steps (similar to [3]):

1. Formalization of the syntactic structure of a requirement in terms of a syntactic interface (i.e. we extracted stimuli and reactions mentioned in the requirements and modeled them as ports with an associated data-type). This was achieved by creating an AutoFOCUS3 component for each requirement with input and output ports.

2. Formalization of the desired behavior stated in a requirement by adding an interface behavior specification to the syntactic interface. We decided to use the specification patterns introduced by Dwyer et al. [6], which are directly supported by AutoFOCUS3 and can be translated to CTL formulas, in order to specify the behavior of a requirement.

The specification patterns, more precisely the CTL formulas derived from them, specify input/output traces of system behavior that need to be reflected by the final implementation.

**Functional Architecture.** In order to derive a specification for the PSD function within the TGMT system, we had to specify a system behavior that is consistent with the requirements. While requirements, in general, are unstructured and incomplete descriptions of desired interactions between a system and its context, a specification aims at a complete and structured description of system behavior that is consistent with the requirements. As system specifi-
Figure 3: The *train door function*, a function of the on-board subsystem, consists of three subfunctions. Communication between (sub)functions (encircled) solely originates from relations between requirements and is not the result of architectural considerations.

Examinining the formalized requirements, we found that some requirements affect the same output and some requirements reference each other (i.e. the output of one requirement is an input of another requirement). We exploited these structural dependencies in order to specify a set of functions that can be traced to the corresponding requirements. We call the resulting set of functions and their relations the *functional architecture* of the system. For the PSD function, we developed a functional architecture for each subsystem (on-board and wayside).

The mapping of requirements to functions was established by their interface definition. The interface of a (formalized) requirement must be part of the interface of the function that the requirement is mapped to. Requirements that affect the same output are mapped to the same function that integrates all the requirements.

Figure 3 shows the *train door function*, a function of the on-board subsystem, that consists of three subfunctions. Internal communication between the subfunctions results from relations between the mapped requirements, i.e. a requirement mapped to one subfunction takes an output of another requirement mapped to another subfunction. It is important to note that communication between functions solely originates from relations between requirements and is not the result of architectural considerations.

We provided a behavior specification for each (sub-)function. In contrast to the requirements, where we used specification patterns translated to CTL formulas for the formalization of their behavior, we now used specification techniques that are complete and executable. In AutoFOCUS3, there are different types of executable specification techniques. In this project we have mainly used code specifications and automaton specifications.

Verification. Functions are linked to the set of requirements from which they were constructed. Each of these requirements has a formalization that corresponds to a subset of the input and output ports of the function. Thus, we were able to verify if the behavior of the function is consistent with the requirements formalization.

The verification technique used by AutoFOCUS3 is model checking (for details see [1, 5]). AF3 translates a function to a labeled transition system and a formalized requirement to a CTL formula. NuSMV$^4$ is used by AF3 to check whether the functional architecture is a model for the formalized requirement. In cases, in which a counterexample was found by the model checker (i.e. a simulation run that violates the requirement), it was possible to animate the counterexample in the AF3 simulator.

3.3 Findings

We were able to model most of the requirements using AutoFOCUS3. An exception were real-time requirements, which need special treatment because of the underlying time-discrete semantics of AutoFOCUS3. These were intentionally left out in this project and will be addressed in a follow-up project. The integration of a glossary in the modeling tool facilitated understanding the domain. As the academic partners were not experts in the domain of rail automation, the lack of domain knowledge of important concepts could be compensated by the integrated glossary to a certain extent.

One of the main findings was the rather big gap between system requirements (RS) and requirements originating from the system architecture specification (AS), as different types of design decisions, namely *scoping* and *concretization*, were taken in one step when moving from RS to AS. By concretization, we mean a refinement of high-level requirements...
defined over the system interface without adding architectural decisions (black-box view). By scoping, we refer to refined requirements due to the inclusion of architectural decisions (i.e., breaking the system down into sub-systems). In order to enable tracing and refinement from system requirements to the architecture specification we needed to reduce the gap between the two levels of requirements. Therefore, we introduced a new set of requirements, which we called intermediate requirements. These intermediate requirements connect RS and AS requirements, by having the same scope as the RS requirements, i.e., they take a black-box view onto the system, while containing all concretization information of the AS requirements (see Figure 4). By introducing intermediate requirements we were able to specify a formal notion of tracing and refinement from system requirements to the architecture specification as a necessary design step to allow consistency checks between the system requirements and the architecture specification.

Verification of the formalized requirements revealed some issues in the specifications, which stimulated discussions between the different stakeholders. For instance, we found that some requirements lacked information or assumed certain properties of the context that were not explicitly formulated. Those pieces of information could be added to the requirements in order to make them self-contained. Under specification of high level requirements, which is a valid technique to keep requirements small and readable (by only describing the reaction of the system for a certain input situation and not for all situations) was handled by using specification patterns as formalization technique instead of state machines.

We successfully verified about 75% of the formalized low level requirements. Verification failures were due to either incomplete and inconsistent requirements or tool issues. In the former case, a failed verification pointed to requirements that needed to be discussed with the stakeholder and thus provided an indication of possible quality issues.

The functional architecture could, to a large degree, be derived from the formalized requirements in a straightforward manner. The structure of the functional architecture was defined by grouping requirements together and identifying internal communication. As requirements were uniquely mapped to functions, the behavior of corresponding functions could also be created without much effort. This led to an executable system specification.

Most of the effort in the project was spent on requirements modeling. This included the creation of the glossary, the formalization of requirements and the extraction of intermediate level requirements. However, this effort facilitated other activities. For example, formalization of requirements greatly sped up the creation of the functional architecture, as we could extract its structure directly from the requirements.

4. EVALUATION

In general, the project was perceived as very beneficial for both, industrial and academic partners. In the following, we evaluate the collaboration with respect to achievement of the academic and industrial goals given in Section 2 and highlight the success factors and constraints that influenced the achievement of these goals.

4.1 Achievement of Goals from Academic Perspective

Validation of the Development Method. The seamless model-based approach proved to be largely applicable to the case study. We were able to model functional requirements in natural language, formalize a great portion of them (ca. 80%) and extract a functional architecture from the formalization. When we did not formalize a requirement this was for one of two reasons: a) the requirement described no behavior or was formulated too abstractly to formalize it, or b) the requirement included timing behavior, which was out of scope for this project. Through modeling, we found peculiarities in the requirements that were investigated in detail together with the domain experts.

Moreover, we were able to model the system of the case study without the need to extend the SPES MF. AutoFOCUS3 was able to support the SPES MF for our case. The modeling and analysis facilities were suitable for the modeled PSD system. We were able to verify about 75% of the formalized low level requirements. Verification here means to ensure that the specification fulfills all (formalized) requirements attached to it. In particular, the formal verification checks between the formalized requirements and the functional architecture were performed within seconds, giving instant feedback to the developer about the impact of any change to requirements or architecture.

Besides supporting the SPES MF, additional functionality provided by AutoFOCUS3 proved beneficial. In particular, highlighting domain terms greatly alleviated the process of understanding for non-domain experts. Furthermore, tracing functionality for linking requirements, on the one hand, to their location within the original requirements documents and, on the other hand, to the corresponding functions, improved clarity and supported validation of the design. Integrating the verification in AutoFOCUS3 in a user-friendly manner, for example by using patterns for specification instead of temporal logics and simulating the counterexample, allowed us to present the techniques in a way that they can be understood by practitioners.

New Research Topics. While applying the SPES MF, we found several interesting new research directions with respect to the development process originally applied by Siemens. Most prominently, we identified a methodical gap...
between system requirements and the architecture specification, which we filled in the model by the introduction of intermediate requirements as described in Section 3.3. We believe that such intermediate requirements can in general be helpful for systematic requirements refinement. An interesting question is if and how the definition of intermediate requirements can be automatically supported. A further research topic that emerged is the construction of a functional architecture from requirements by means of their formalized interfaces. A systematic method had so far not been explicitly formulated. In the case study we made a first attempt. However it would be interesting to see if this can be applied in other contexts.

4.2 Achievement of Goals from Industrial Perspective

Modeling of TGMT’s platform screen doors functionality using the tools and methods presented in this paper indicates a promising potential for improvement of the development processes. Proving refinement relations between different abstractions of the architecture gives the opportunity to discover implicit design decisions that have been taken due to engineer’s experience with earlier versions or with predecessor systems. Making implicit decisions explicit gives the opportunity to reconsider old design decisions, thus giving more opportunities for the evaluation of the system and increasing quality.

Integration of different views of the same system strongly supports keeping these views consistent and thus reduces sources of errors or misunderstandings stemming from inconsistent descriptions of the system. Nevertheless, the maturity of tools used for the description needs to be sufficient for an industrial application including long-term support over the complete life-cycle of a product which can easily exceed 30 years for a typical product in rail automation.

It turned out that the SPES MF development method could be applied to the existing and unchanged system descriptions that we used as input for the modeling. This is an important insight, as these documents were originally created with a different process and development methodology in mind and were not specifically designed for the application of model based development. On the other hand, we had to deal with the fact that the documents were written by domain experts and thus were sometimes hard to understand by the academic partners. The inclusion of a glossary into AutoFOCUS3 helped a lot to deal with this problem.

Looking ahead, we see even more opportunities than those addressed during this project. Formal modeling gives us the opportunity to early detect design flaws, check artifact consistency, and automatically generate test cases. In addition, dependability and timing analysis as another aspect seems to be promising from an industrial use case.

4.3 Success Factors

In the following, we describe the success factors that were crucial for achieving the goals.

Cooperation with product experts. The involved persons from industry were experts in the train automation system. For example, the system architect was part of the project team. This facilitated understanding the domain and the system. Regular phone conferences and workshops allowed to continuously discuss results and competently review the models that were created.

Modeling of an existing system. Our study object originated from a mature and widely deployed system. The participating industrial experts had good knowledge about its functionality and especially about its subtleties. This allowed the academic partners to focus on applying the development method rather than on defining the system behavior, which was fixed to a large extent. In addition, the domain experts were able to assess whether the resulting models were realistic and valid. This was a big advantage compared to other case studies, where we had to deal with systems that had not been built so far and thus required significantly more effort to understand and develop due to unclear requirements.

Glossary. The availability of an extensive glossary facilitated understanding the domain. After importing the terms to AutoFOCUS3, the glossary could be used seamlessly in the tool. Without that integrated glossary, understanding and formalizing the requirements would have been much harder.

Quality of documents. The specification documents provided by Siemens were very mature, comprehensive and well structured. Only few linguistic inconsistencies were found. The textual requirements could easily be extracted and imported into AutoFocus3. Thus, the documents provided a good starting point for modeling. We found that most requirements were self-contained, which facilitated later verification.

Mature Tool Support. Modeling and discussing the results by means of a mature tool contributed to the credibility and transparency of our findings. Only by the help of the tool we were able to get quick feedback and confidence on the correctness of our formalizations. The stability and usability of AutoFOCUS3 additionally enabled to actually use the tools in meetings and to investigate modifications to the model “on the fly”.

4.4 Constraints and Disadvantages

Modeling done by method and tool experts. The major part of the modeling work was done by the academic partners, who were experts for the SPES MF and AutoFOCUS3 but not in the rail automation domain. A disadvantage of this collaboration setup was that we could hardly obtain any assessments about tool usability by nonexperts or any realistic effort estimations. On the other hand, letting the domain experts do the modeling would also not have produced realistic assessments about modeling efforts or tool usability because in a realistic context, developers would have been trained beforehand in using tools and methods to be applied.

Restriction to a sub-function. Due to resource constraints, we decided to model only one sub-function of the TGMT system, namely the PSD control. However, the documents provided by Siemens described the whole TGMT system. A challenge arising from that was to extract the relevant information associated to the PSD sub-function which was spread throughout the documents. For some functionalities, for example control of the train operation mode, it was not immediately clear whether a given function should be part of the system under development or be taken as an external parameter (i.e. as a function in the context). This had also implications for design decisions, which could have influenced the validity of the study results.
5. SUMMARY AND OUTLOOK

In this paper, we reported on a collaborative project between the Rail Automation business unit of Siemens AG, the Technische Universität München and fortiss GmbH. The main goal of the project was to evaluate the SPES development approach and its tool integration with AutoFOCUS3 in the context of an industrial setting. The collaboration was set up as a case study project, where the method and the tool were demonstrated by (re-)modeling a productive system on the basis of real specification documents provided by Siemens.

After the collaboration both, academic as well as industry partners, concluded that the modeling approach and its tool integration are well suited to model a productive system in the context of rail automation.

In the paper, we identified important success factors of the collaboration that contributed to the achievement of the goals. Amongst them, (re-)modeling of an existing system instead of a fictional example, the high quality of input documents, and a cooperation with product experts were considered the most influential success factors.

Besides the success factors, we also identified constraints and disadvantages regarding the type of collaboration. Most significantly, we were not able to create realistic assessments about modeling efforts and tool usability because most of the modeling work was done by the modeling and tool experts. It would be interesting to conduct a similar experiment, where developers get trained in tools and methods and then start modeling the system.

In a follow up project, we plan to extend the modeling to also include a logical architecture as well as a deployment of the logical architecture to control units. Another area of interest are timing aspects of the PSD system and the generation of schedules that minimize end-to-end latencies.

Acknowledgments

We would like to thank Andreas Bauer, Henning Femmer, Georgeta Igna, Diego Marmsoler, Dongyue Mou, and Daniel Ratiu for their work and valuable comments on this paper.

6. REFERENCES