Applying the SPES Modeling Framework

A Case Study from the Automotive Domain

Jennifer Brings, Julian Bellendorf, Kevin Keller, Markus Kempe, Noyan Kurt, Alexander Palm, Marian Daun

paluno - The Ruhr Institute for Software Technology,
University of Duisburg-Essen, Essen, Germany
{jennifer.brings,marian.daun}@paluno.uni-due.de
{julian.bellendorf,kevin.keller,markus.kempe,
noyan.kurt,alexander.palm}@stud.uni-due.de

Abstract. [Context & motivation] Model-based engineering, and model-based requirements engineering in particular, has commonly been valued in the automotive domain. Hence, model-based engineering methodologies have been proposed for the engineering of automotive systems, such as the SPES modeling framework, which has been positively evaluated in the German embedded industry. **[Question/problem]** However, the increasing interconnectivity of automotive systems raises new challenges for their development in general and for requirements engineering in particular. Existing approaches to model-based engineering of embedded systems might only be partially suitable for developing such highly connected embedded systems. **[Principal ideas/results]** To investigate the applicability of existing approaches for developing of such systems, we applied the SPES modeling framework, a framework for continuous model-based engineering of embedded systems, in a case study. As case example autonomous driving on controlled-access highways was chosen. **[Contribution]** This paper contributes preliminary results from our ongoing case study and provides first insights into the needs for adaptation of model-based engineering frameworks to cope with the challenges resulting from the increased interconnectivity of cyber-physical systems.

Keywords: Cyber-physical system, model-based development, case study

1 Introduction

It has been shown that model-based engineering is an appropriate means to deal with the growing complexity of safety-critical embedded systems such as those found in automobiles (cf. e.g., [1–3]). To aid their development, engineering methodologies such as the SPES modeling framework [4] have been proposed. The SPES modeling framework aims at continuous model-based engineering of embedded systems, including closely integrated model-based requirements engineering. The SPES modeling framework has already been applied to case examples and evaluated in the area of embedded system development [5–11].

Copyright 2017 for this paper by its authors. Copying permitted for private and academic purposes.

However, as embedded systems become more and more cyber-physical the question arises, whether or to what extent model-based engineering methodologies such as the SPES modeling framework are applicable to such highly-interconnected systems. To investigate if the SPES modeling framework is suitable for the development of cyber-physical systems (CPS), we are conducting a case study using an interconnected highway-driving assistant as case example. This paper reports on the setting of the case study and gives insights into first findings regarding the need to adapt model-based engineering frameworks to cope with highly-interconnected CPS, particularly, from a requirements point of view.

2 The SPES Modeling Framework

The SPES modeling framework [4] was created to support the continuous model-based engineering of embedded systems in various application domains (e.g., automotive industry, avionics, energy, health care, industry automation). Its artifact-centric nature allows for engineering artifacts, i.e. models to be created depending on the individual needs without prescribing a rigid process for creating them. To this end, the framework defines four viewpoints: the requirements viewpoint, the functional viewpoint, the logical viewpoint, and the technical viewpoint; thus allowing for separation of concerns. The viewpoints predefined within the framework address the concerns of one or more stakeholders commonly found in embedded system projects, but viewpoints can be added and discarded as needed for the project at hand. Additionally, the framework supports the definition of granularity layers as needed, based on the particular demands of a development process. These granularity layers allow for using abstraction mechanisms to reduce complexity. Fig. 1 illustrates the frameworks viewpoints and granularity layers.

The requirements viewpoint [12] focuses on the context of the system under development (SUD) as well as on fundamental behavior and functions the SUD has to provide. The developed models for the requirements viewpoint commonly serve as a basis for further engineering artifacts (e.g., functional design, logical and technical architectures). In particular, the requirements viewpoint contains models about the goals of the SUD, context models highlighting system border, context and context border, and scenarios pertaining to the SUD.

The functional viewpoint [13] specifies the system's functionality in a detailed way. In this viewpoint the system functionality defined in the requirements viewpoint is refined into more fine-grained implementable functions. Additionally, the function behavior and the interfaces between system functions and functions of other systems are specified.

The functional viewpoint is closely connected to the logical viewpoint. The logical viewpoint [14] focuses on the decomposition of the system into logical components. This is commonly achieved by partitioning all defined system functions to logical components, which will later on be deployed to the same electronic control unit. Hence, this viewpoint serves as a bridge towards the technical viewpoint, as important architectural decisions are made.

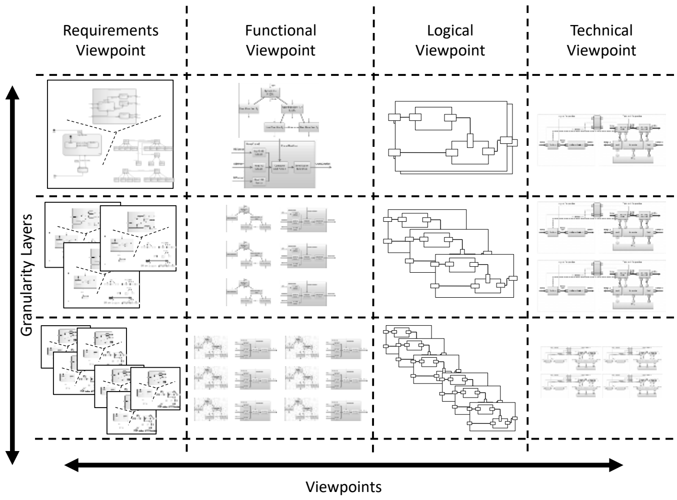

Fig. 1. SPES Modeling Framework

The technical viewpoint [15] incorporates hardware features, as the technical architecture is specified in detail. This viewpoint focuses on the deployment of the logical components defined in the logical viewpoint to the hardware components.

The SPES framework does not prescribe a path through viewpoints or granularity, permitting engineers to choose their own path through viewpoints and granularity layers of the framework as required. This allows for SPES conform development processes to be tailored to different domains (e.g., automotive industry, avionics, industry automation) and to different companies.

3 The Case Example

For years automotive embedded systems have taken over tasks that used to be the driver's responsibility. It will not be long before autonomous cars will be a common sight on streets. Meanwhile new cars are increasingly being equipped with driver assistance systems that partially automate driving in certain situations such as parking or driving on controlled-access highways. While previous case studies have evaluated certain aspects of the SPES modeling framework (e.g., the exemplary use of single viewpoints, the use of granularity layers in one viewpoint or the transition from one viewpoint to another), our ongoing study investigates the application of the SPES modelling framework to the case example of an autonomous highway driving system (AHDS) across two granularity layers in all viewpoints.

Autonomous highway driving systems can take over the driving task from the vehicle's driver while on an access-controlled highway. With the aid of other automotive embedded systems such as the adaptive cruise control, the lane changing assistant, the brake system, etc. the AHDS can coordinate the vehicle's speed, lane choice, react to dangers etc. just like a human driver would. Beyond that the AHDS is capable of exchanging information with other equally equipped vehicles in its vicinity. The exchange of information such as road, weather, and traffic conditions allows the AHDS to adapt its behavior accordingly and thus prevent accidents and traffic jams. This connectivity allows the AHDSs to form dynamic networks at runtime.

4 First Results

So far our ongoing case study has yielded several interesting results regarding the applicability of the SPES modeling framework to CPS. The SPES modeling framework seems to a large extend capable of dealing with the challenges posed by highly-connected systems. All relevant aspects of the AHDS can be captured using the methodological framework and appropriately documented in a model-based fashion.

However, while interdependent relations exist (e.g., a context instance used in a scenario must also be documented in a concrete context model) between the different artifacts (particularly within the requirement viewpoint) for all kinds of systems, the number of dependencies seems to increase for highly-connected CPS. For instance, the AHDS does not only perceive its environment by sensors, but it also relies on information from additional systems which enter and leave the context of the ADHS independent of each other. Consequently, this manifests itself in an increased number of context systems the AHDS interacts with. The identification of a new system in the context of the AHDS does not only affect the context models in the requirements viewpoints but also other models that depict parts of the context such as scenario models.

Additionally functionality of the systems entering and leaving the context is used by the AHDS to fulfill its own goals and its own functionality and behavior is altered due to the specific operational context. Identified changes in the context result in multiple revisions to nearly all other requirements models. As these revisions can again force new revisions it can become difficult to keep track of all necessary changes and the current state of work for each model. Hence, it seems beneficial to restrain the development process within the SPES modeling framework, specifically within the requirements viewpoint, in such a way that models are more stable and do not need to be changed that often. For instance, it seems advantageous to not start iteratively developing context, goal, and scenario models as commonly suggested in goal-scenario-based requirements engineering, but to advance one model as far as possible before creating the next model.

Another issue arising in the context of cyber-physical behavior is the treatment of properties which are the same within in the AHDS and the context. CPS often interact in networks that contain other CPS of the same type. Specifying the AHDS and documenting its context leads to duplicates which are notorious sources of inconsistencies and thus problematic in software engineering. As one potential solution, a scenario-centric engineering methodology might aid the development within the requirements viewpoint. In doing so, scenarios describing some system interactions can be reused to also describe context behavior and vice versa. Therefore, we found the collaborative aspects of the AHDS to be best modeled using message sequence charts

[16], as they allow to reference system behavior exhibited by context entities that can also be exhibited by the AHDS. An example is shown in Fig. 2. Even though the MSC *Alert Driver* documents behavior of the AHDS (here the *AHD-System*), context systems (here the *Other AHD-System*) need to exhibit the same behavior, which can be modeled as a reference. Hence, parts of the behavior of a context instance can be described by the same behavior as specified for the AHDS itself. As message sequence charts are commonly used for scenario descriptions and also allow for detailed specification of the complete interaction-based behavior of the AHDS under consideration of context aspects, we assume that the development of detailed scenario descriptions at first and their completion can provide a fairly stable basis for the definition of other aspects relevant to requirements engineering.

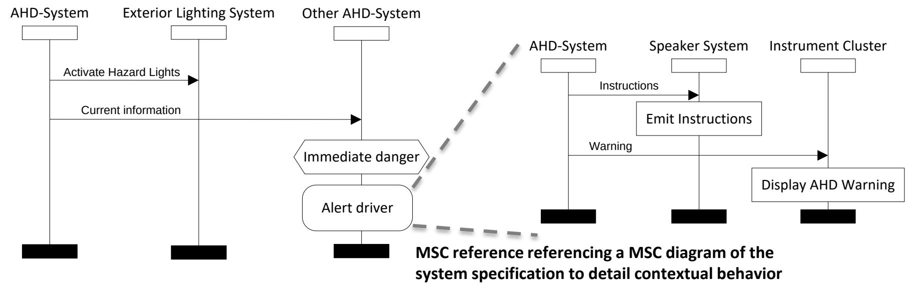

Fig. 2. Message Sequence Charts: Alert other driver (left), Alert Driver (right)

5 Conclusion and Future Work

The SPES modeling framework offers a structured approach to modeling not only embedded systems, but also CPS. In this paper, we reported on first findings regarding challenges posed for model-based engineering frameworks. We identified problems resulting from an increased number of dependencies. While interdependent relations exist for all types of systems and, hence, potentially pose a problem for the development using the SPES modeling framework in general, the highly connected nature of CPS, however, seems to exacerbate the problem of keeping all artifacts consistent. Furthermore, we identified the need to cope with redundancies caused by properties which are system as well as context properties in a structured manner. Hence, future work will have to deal with the integration of existing traceability approaches to keep track of affected artifacts and model transformation approaches that can propagate changes. As we already identified message sequence charts specifications as potential anchor models to ensure consistency and manageability of model-based requirements engineering for CPS, we intend to investigate benefits and potential disadvantages in more detail during the ongoing case study.

6 References

1. Schätz, B.: Model-Based Development: Combining Engineering Approaches and Formal Techniques. In: Formal Methods and Software Engineering, 6th International Conference on Formal Engineering Methods, pp. 1–2. Springer (2004).
2. France, R., Rumpe, B.: Model-driven Development of Complex Software: A Research Roadmap. In: 2007 Future of Software Engineering. pp. 37–54. IEEE Computer Society, Washington, DC, USA (2007).
3. Schmidt, D.C.: Guest Editor's Introduction: Model-Driven Engineering. IEEE Computer. 39, 25–31 (2006).
4. Broy, M., Damm, W., Henkler, S., Pohl, K., Vogelsang, A., Weyer, T.: Introduction to the SPES Modeling Framework. In: Model-Based Engineering of Embedded Systems. pp. 31–49. Springer (2012).
5. Wagner, T., Wehrstedt, J.C., Löwen, U., Jäger, T., Fay, A., Schuller, P.: Application and Evaluation in the Automation Domain. In: Model-Based Engineering of Embedded Systems. pp. 137–155. Springer (2012).
6. Fockel, M., Heidl, P., Höfflinger, J., Hönninger, H., Holtmann, J., Horn, W., Meyer, J., Meyer, M., Schäuffele, J.: Application and Evaluation in the Automotive Domain. In: Model-Based Engineering of Embedded Systems. pp. 157–175. Springer (2012).
7. Bender, O., Hiller, M., Girod, M., Strobel, C., Waßmuth, M., Dieudonné, L.: Application and Evaluation in the Avionics Domain. In: Model-Based Engineering of Embedded Systems. pp. 177–196. Springer (2012).
8. Fasse, F.-W., Glomb, C., Grünbauer, J., Heuer, A., Klaus, M., Kuntschke, R., Laskowski, M., Weyer, T.: Application and Evaluation in the Energy Domain. In: Model-Based Engineering of Embedded Systems. pp. 197–214. Springer (2012).
9. Heinze, H., Kallow, K., Lackner, H., Sadeghipour, S., Schlingloff, H., Tahirbegovic, S., Wiesbrock, H.-W.: Application and Evaluation in the Healthcare Domain. In: Model-Based Engineering of Embedded Systems. pp. 215–230. Springer (2012).
10. Pohl, K., Broy, M., Daembkes, H., Hönninger, H.: Experiences of Application in the Automation Domain. In: Advanced Model-Based Engineering of Embedded Systems. pp. 225–239. Springer (2016).
11. Pohl, K., Broy, M., Daembkes, H., Hönninger, H.: Evaluation of the SPES XT Modeling Framework. In: Advanced Model-Based Engineering of Embedded Systems. pp. 263–271. Springer (2016).
12. Daun, M., Tenbergen, B., Weyer, T.: Requirements Viewpoint. In: Model-Based Engineering of Embedded Systems, The SPES 2020 Methodology. pp. 51–68. Springer (2012).
13. Vogelsang, A., Eder, S., Feilkas, M., Ratiu, D.: Functional Viewpoint. In: Model-Based Engineering of Embedded Systems. pp. 69–83. Springer (2012).
14. Eder, S., Mund, J., Vogelsang, A.: Logical Viewpoint. In: Model-Based Engineering of Embedded Systems. pp. 85–93. Springer (2012).
15. Weber, R., Reinkemeier, P., Henkler, S., Stierand, I.: Technical Viewpoint. In: Model-Based Engineering of Embedded Systems. pp. 95–106. Springer (2012).
16. International Telecommunication Union: Message Sequence Chart (MSC). (2011).