International Journal on Software Tools for Technology Transfer
https://doi.org/10.1007/s10009-024-00753-2

GENERAL

Special Section: ABZ 2020/2021

An automotive case study

Alexander Raschke[1] · Dominique Méry[2]

Accepted: 13 May 2024

1 Introduction

Modern cars have become driving computers that provide a variety of safety and comfort functions. However, most automotive systems do not have a central computer, but typically consist of a set of communicating electronic control units (ECUs).

There are several reasons for this: On the one hand, one wants to avoid a single point of failure, and on the other hand, ECUs installed close to sensors or actuators save wiring harnesses and therefore weight. Finally, the limited space and power supply in vehicles restrict the possible technologies. One disadvantage of this distributed architecture is the increased communication effort between the ECUs. One advantage, however, is that more and more (and increasingly complex) functions can be implemented in software. This in turn changes the possibilities for and needs of quality assurance of systems from pure (mechanical) testing to formal methods. However, due to the increasingly complex interrelation of different systems, it is also necessary to recognize and control possible interactions between the different, supposedly independent functions at an early stage of development.

This special issue is dedicated to the formal specification and verification of an automotive case study presented at ABZ 2020. However, this conference had to be canceled due to the unclear situation at the beginning of the spread of COVID-19 and its classification as a pandemic. It was finally held one year later as an online-only conference. The collection of supplemented and extended contributions to the case study "Adaptive Exterior Light and Speed Control System" [5], which describes the functionality of two systems in a modern vehicle, is presented in this volume.

✉ A. Raschke
alexander.raschke@uni-ulm.de

D. Méry
dominique.mery@loria.fr

1 Institute of Software Engineering and Programming Languages, Ulm University, Ulm, Germany

2 LORIA, University of Lorraine, Nancy, France

First, a short introduction to the systems described in the case study is given before the five contributions, utilizing four different formalization approaches, are briefly summarized.

2 System overview

This section briefly introduces the case study. For a detailed description, please refer to the requirements document, which can be found in the proceedings of ABZ 2020 [10] or online at https://abz2020.uni-ulm.de/case-study.

The case study describes two subsystems that are only loosely coupled: The "Adaptive Exterior Light" and the "Speed Control System". This division makes it possible to model only one of the two subsystems. A special feature of this case study is the systems' configuration options. While in real life, countless variants must be managed via software product lines, in this case we limited ourselves to just a few, such as left/right-hand traffic or the market for which the vehicle is designed, impacting some requirements.

For each subsystem, a detailed description of the user interface (e.g., steering column lever, cruise control lever, various (rotary) switches and pedals) as well as the sensors and actuators is given. The requirements document focuses on functionality and therefore abstracts all communication and hardware details. To this end, abstract signals are introduced that can be read to obtain the values supplied by the sensors or can be set to activate actuators.

2.1 Adaptive exterior light

The exterior light system (ELS) of a modern car no longer simply switches on and off the headlights, but integrates (among others) the following (comfort) functions:

- **Low beam headlights:** Control of the low beam headlights combined with a daytime running light and coming-leaving-home functionality, including parking light.
- **Adaptive high beam:** Control of the high beam headlights with an adaption concerning the driving speed.
- **Cornering light:** Control of additional headlights that illuminate the cornering area separately when turning left or right.

Published online: 16 May 2024

- **Turn signal:** Control the driving direction indicators, including tip-blinking and hazard warning light system.
- **Emergency brake light:** Warns following drivers by a flashing brake light in case of an emergency brake.

In addition to the 40 functional requirements enumerated for this subsystem, eight requirements for the behavior in case of over- or under-voltage were defined to guarantee the lighting system's most important functionality for as long as possible.

2.2 Speed control system

The speed control system (SCS) can control acceleration and braking until the vehicle comes to a standstill and offers functions such as (adaptive) cruise control, distance warning, emergency braking assistance and automatic enforcement of the maximum speed recognized on traffic signs:

- **Cruise control:** The vehicle automatically maintains a set speed, independently of the distance to other vehicles. Here, the driver is in charge of maintaining a safe distance.
- **Adaptive cruise control:** The vehicle maintains the distance to the preceding vehicle, including braking until a complete standstill and starting from a standstill.
- **Distance warning:** The vehicle warns the driver visually and/or acoustically if the vehicle is closer to the car ahead than allowed by the safety distance.
- **Emergency brake assist:** The vehicle decelerates to a complete standstill in critical situations.
- **Speed limit:** The vehicle does not exceed a set speed.
- **Sign recognition:** The vehicle sets the speed limit automatically according to the recognized signs.
- **Traffic jam following:** The vehicle accelerates from a standstill when the preceding vehicle departs.

This subsystem was described in 39 listed requirements. Four additional safety requirements limit safety behavior when the distance to the vehicle in front cannot be correctly determined.

2.3 Improvements of the textual specification

Based on the experiences gained with previous case studies [2, 3, 9], two appointments were offered where questions could be asked to clarify ambiguities, misunderstandings, or inconsistencies in the textual requirements.

The questions asked during these meetings and various e-mail correspondences with individual authors resulted in several extensions and improvements to the requirements document, which are available on the website mentioned above including a change history. Validation sequences were also created to enable the comparison of the created formal models with the intended functionality. They are also available on the ABZ 2020 website.

In this respect, the formalization of the requirements has already helped to identify inconsistencies and ambiguities in the textual requirements, regardless of the specific modeling (and verification) approach used.

3 Overview of contributions

To make it easier for readers to compare the different modeling solutions to the case study, we have devised a common structure that all contributions should follow:

Introduction introduces the methods and tools used and provides relevant information about the applied method that is needed for non-experts to understand and validate the models. In addition, any distinctive features of the described approach shall be given.

Requirements and modeling strategy explains if the structure given by the requirements document was used or if it was reorganized into a different one and how the structure of the formal model is related to it. This section also shall answer the following question: Does traceability exist between the formalization and the requirements? How is the variability of the requirements represented in the model? What are the most important properties addressed by the presented solution? Finally, any features of the requirements not addressed by the solution shall be mentioned in this section.

Model details provide insights into how the formalization of the requirements was approached. This includes the description of the modeling styles used and key snippets of the models following the structure outlined above. An interesting question is, how readable and understandable might the model be for non-experts, and how the authors have tried to increase readability? An important part of this section is how time constraints were modeled.

Validation & verification describes strategies and tools used to validate and verify the model (e.g., results, degree of automation, etc.) and whether and how the provided validation sequences were used. Also, changes to the model that resulted from the validation or verification phase shall be provided in this section. The answer to the question of how the verification capabilities of the chosen technology influenced the modeling itself provides interesting insights.

Other observations can be used to explain any ambiguities, limitations, or flaws the authors identified in the reference document, or suggestions for improving the requirements and/or the method and tools that would help apply them to the case study. In this section, the authors were asked to explain how their solution could support the derivation or verification of a software implementation of the described system, the execution of the model, the translation to an executable, or testing an executable.

Comparison outlines the main differences between the described solution and the other case study solutions published in the ABZ 2020 proceedings.

Based on this structure, the following five contributions are presented in this special issue:

The first two papers "*An Event-B Model of an Automotive Adaptive Exterior Light System*" and "*Modeling of a Speed Control System using Event-B*" by Mammar et al. [7, 8] consider the subsystems separately. In both papers, the authors used similar approaches to create Event-B models, refine them, and validate by the animation capability of ProB. The verification of the system was done by proving all obligations produced by the tool Rodin. This contribution focused on modeling temporal (dependencies on variable changes) and timing constraints (e.g., something happens 3 seconds after another event) by introducing an artificial time. This was necessary because Event-B does not directly support LTL, and the model was too large for external tools like ProB. In addition, the authors describe in detail which inconsistencies or ambiguities were discovered in the requirements document in the various phases (modeling, validation, verification) and how they were resolved.

The third contribution, "*Validating Multiple Variants of an Automotive Light System with Alloy 6*" by Cunha et al. [4], presents an Alloy 6 model of the exterior light subsystem. Alloy 6 is the successor of Electrum and also supports mutable relations and LTL constraints. The main goal of this contribution was to validate the variability in the ELS by exploring different strategies. For this purpose, the authors developed a tool to translate the provided validation sequences into Alloy 6 and back that can be reused for other signal-based systems.

"*A Journey with ASMETA from Requirements to Code: Application to an Automotive System with Adaptive Features*" by Arcaini et al. [1] uses the ASMETA toolset to create an abstract state machine (ASM) model of both subsystems. The applied modeling process is — as suggested by the ASM theory — iterative and refinement-based, starting with an abstract ground model. A MAPE-K feedback loop was adopted to model the adaptive control features. Like the contribution mentioned above, time was abstracted by introducing functions that notify that a certain amount of time has passed. Regarding the variability, flags were used to indicate a specific configuration.

The fifth paper "A Verified Low-Level Implementation and Visualization of the Adaptive Exterior Light and Speed Control System" by Krings et al. [6] provides a different approach: Instead of defining abstract models in a formal specification language, they developed a running implementation in C according to the MISRA guidelines used for safety-critical systems. Validation was done by strict test-driven development and verification using CBMC for model checking. Although this approach does not offer the mathematical precision and correctness of formal methods, it can be seen as a baseline for comparing formal approaches with classical software development.

4 Conclusion

This introduction briefly overviews the ABZ 2020/21 case study system. It explains the desired, uniform structure of the submissions and briefly presents them.

The contributions presented in this special issue enrich the set of case studies developed within the ABZ community and related methods. We believe that they will help readers to assess the strengths and weaknesses of the various formal methods used by the contributors.

Acknowledgements We want to thank the authors for their patience during the long time it took to produce this special issue. We would also like to thank the reviewers for their willingness to volunteer their time for the review, especially during the challenging pandemic period. Special thanks also go to Frank Houdek, who sacrificed a lot of his free time to patiently answer the many questions regarding the requirements document in detail.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Arcaini, P., Bonfanti, S., Gargantini, A., Riccobene, E., Scandurra, P.: A journey with ASMETA from requirements to code: application to an automotive system with adaptive features. This issue
2. Boniol, F., Wiels, V.: The landing gear system case study. In: Boniol, F., Wiels, V., Ait Ameur, Y., Schewe, K.D. (eds.) ABZ 2014: The Landing Gear Case Study, pp. 1–18. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07512-9_1
3. Butler, M., Raschke, A., Hoang, T.S., Reichl, K.: Introduction to special section on the ABZ 2018 case study: hybrid ERTMS/ETCS level 3. Int. J. Softw. Tools Technol. Transf. **22**(3), 249–255 (2020). https://doi.org/10.1007/s10009-020-00562-3
4. Cunha, A., Macedo, N., Liu, C.: Validating multiple variants of an automotive light system with electrum. This issue
5. Houdek, F., Raschke, A.: Adaptive exterior light and speed control system. In: Raschke, A., Méry, D., Houdek, F. (eds.) Rigorous State-Based Methods, pp. 281–301. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-48077-6_24
6. Krings, S., Körner, P., Dunkelau, J., Rutenkolk, C.: A verified low-level implementation of the adaptive exterior light and speed control system. This issue

7. Mammar, A., Frappier, M.: Modeling of a speed control system using event-B. This issue
8. Mammar, A., Frappier, M., Laleau, R.: An event-B model of an automotive adaptive exterior light system. This issue
9. Mashkoor, A.: The hemodialysis machine case study. In: Butler, M., Schewe, K.D., Mashkoor, A., Biro, M. (eds.) Abstract State Machines, Alloy, B, TLA, VDM, and Z, pp. 329–343. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-33600-8_29
10. Raschke, A., Méry, D., Houdek, F. (eds.): Rigorous State-Based Methods. Lecture Notes in Computer Science (LNCS), vol. 12071. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-48077-6