ATO OVER ETCS: MODEL-BASED SYSTEM AND ARCHITECTURE ANALYSIS WITH ARCADIA AND EVENT-B

ROBERT ESCHBACH*
Railway Technology, ITK Engineering GmbH, Germany

ABSTRACT

The European Railway Traffic Management System (ERTMS) is intended to replace incompatible national rail traffic management systems in Europe and thus simplify cross-border rail traffic. A part of ERTMS is the European Train Control System (ETCS). ETCS is an automatic train protection system and can collaborate with an automatic train operation system (ATO). ATO can control and monitor the braking, traction and door system of a train. This collaboration is called ATO over ETCS. In this paper we describe the experiences gained in the integrated application of the model-based systems and architecture engineering method ARCADIA and the formal method Event-B to the system requirements of ATO over ETCS. A central part of the system requirements is related to the operational modes, mode transitions and mode properties of the ATO onboard unit (ATO-OB). Mode properties are system requirements that must be satisfied whenever the ATO-OB enters a mode or stays within a mode. Especially modes, in which the ATO-OB automatically drives the train, are of utter importance. The main goal of the analysis was to check consistency and completeness of the system requirements related to modes, mode transitions and mode properties. Within the ARCADIA phase "System Analysis" a so-called "Mode/State Model", a special ARCADIA model, was systematically derived from the system requirements. In order to guarantee consistency, especially to guarantee that mode properties will not be violated by mode transitions, a formal Event-B specification was derived and formally analyzed. This analysis approach identified several critical inconsistencies of the system requirements.
Keywords: ATO over ETCS, MBSE, formal methods, ARCADIA, Event-B.

1 INTRODUCTION

The European Railway Traffic Management System (ERTMS) aims at the replacement of incompatible national railway traffic management systems in Europe. A central part of ERTMS is ETCS. In this paper we will focus on the collaboration of ETCS with ATO called ATO over ETCS. The system requirements for ATO over ETCS have been specified within the Shift2Rail research project X2RAIL-1. Further information can be obtained from `https://projects.shift2rail.org` (accessed July 12, 2022). The system requirements are specified in SUBSET-125. (Technical specifications for ETCS are published in the Control Command and Signalling Technical Specification for Interoperability hosted by the European Rail Agency. These specifications are grouped into several uniquely numbered subsets.) As can be seen in the reference architecture of ATO over ETCS depicted in Fig. 2 the ATO is divided in ATO onboard unit (ATO-OB) and ATO trackside unit (ATO-TS). A main part of the system requirements is related to ATO-OB operation modes, mode transitions and mode properties. Mode properties must be satisfied whenever the ATO-OB enters a mode or stays within a mode. They can be modelled as invariants of the ATO-OB. Modes, in which the ATO-OB automatically drives the train, are critical and hence important. The main goal of the presented analysis was to check consistency and completeness of the system requirements related to modes, mode transitions and mode properties.

* *ORCID: https://orcid.org/0000-0002-2789-1873*

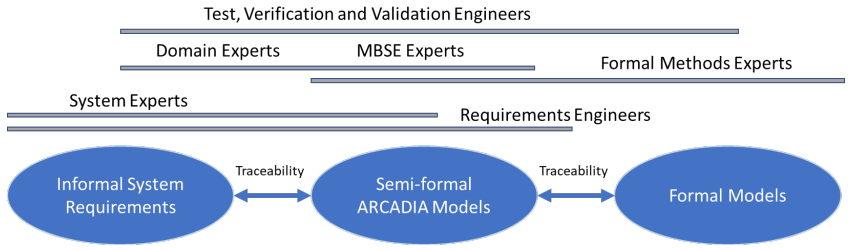

Figure 1: Development roles and scope.

In order to achieve this goal, an analysis basis was defined suited for different stakeholders, especially for the development roles involved in the overall system and software development (the engineering team). This basis was used to enable the analysis of completeness and consistency issues on different formality levels. The analysis involves different kinds of work items: informal requirements, semi-formal ARCADIA models and formal models (more precisely Event-B models). The objectives of the analysis are the following: The analysis basis should be understandable to all stakeholders and should be suitable for their expert knowledge and engineering skills,so that expert knowledge of all stakeholders has the highest influence on the overall system quality. Different work items with different degrees of formality are needed in order to address different types of completeness and consistency issues which require different analysis depths. The higher the degree of formality, the deeper is the formal analysis, but the more expert knowledge related of formal methods is needed. Usually, only a small part of the engineering team is able to peform this kind of formal analysis. Several critical issues regarding consistency and completeness were found in the formal model and could be transferred to the semi-formal ARCADIA models successfully. The formal analysis revealed 23 ambiguities and missing requirements. The anaysis approach and the obtained results will be presented in more detail in the following.

The paper is structured as follows: In Section 2 the underlying analysis approach will be presented. In Section 3 the semi-formal analysis consisting of reviewing system requirements, establishing a bidirectional traceability, analyzing the activation of system functions, verification and validation activities will be explained in more detail. Section 4 focusses on the formal analysis which consists of the identification of components (system, environment), the derivation of so-called function and variable tables, the creation of an Event-B specification together with its verification and validation. Obtained results and experiences gained in the application of the analysis approach are given in Section 5. Concluding remarks, the main results, and some perspectives related the presented study will be summarized in Section 6.

1.1 Related work

Basile et al. [1] present the rationale for designing the formal methods demonstrator for the 4SECURail project. The project 4SECURail is a research project funded by the Shift2Rail Innovation Programme (IP). Thereby they consider the usefulness of the application of formal methods from the infrastructure managers perspective. They show how SysML state machines can be used for execution and simulations and how they can be translated into formal Event-B specifications that can be formally analyzed using model checkers or theorem provers. Graves and Bijan [2] embed SysML into a formal logic to maintain an evolving consistency during design activities. Design decisions may depend on assumptions

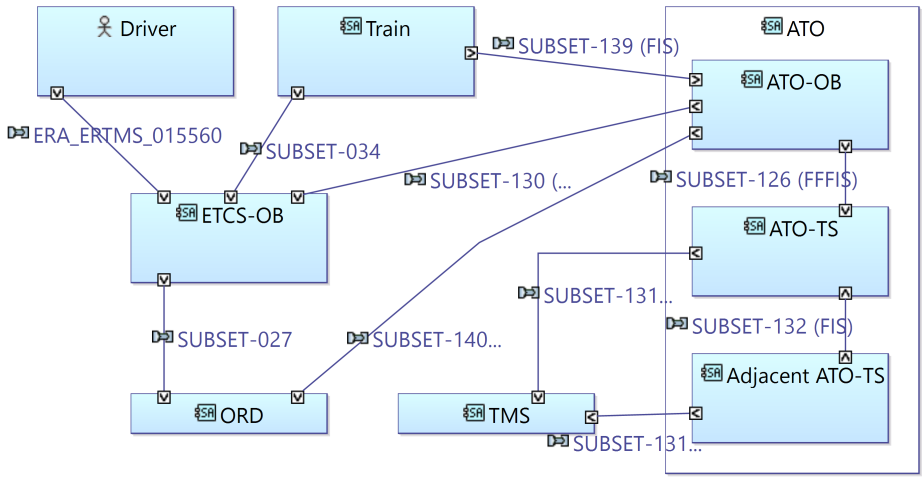

Figure 2: ATO over ETCS reference architecture.

which may introduce inconsistencies during a typical model-based development process. The embedding supports developer by formally verifying consistency when adding new mode assumptions and thus avoid potential rework. Pétin et al. [3] present a methodology how to use formal methods on top of SysML models to formally prove safety properties of heterogeneous safety-critical systems involving software, mechanical, electrical, and pneumatic components. Ahlbrecht and Durak [4] present a holistic application of Model-Based Systems Engineering, System-Theoretic Process Analysis, and formal methods in the area of Urban Air Mobility. The overall goal is to guarantee safety-by-construction within an agile development process. The approach is applied on a simplified Collision Avoidance System architecture. ter Beek et al. [5] present a detailed report on the convergence of FM related studies carried out in the Shift2Rail projects X2Rail-2 and ASTRail. In both projects a systematic survey of the state of the art of formal methods application in railway industry and related best practices was carried out. Basile et al. [6] present the results of a questionnaire which was part of the analysis phase of ASTRail research project. One of the most important results is that classical B and Event-B with the respective tools Atelier-B, ProB and Rodin were used most often in projects. Fitzgerald et al. [7] the authors present an approach to formal model-based systems-of-systems engineering based on different and complementary notations for functional, interaction and architectural aspects.

2 ANALYSIS APPROACH

The overall goal of the model-based system and architecture analysis was the creation of an analysis basis suited for different stakeholders and development roles involved in the overall system and software development as required by railway standards like [8], [9] or [10] (see Fig. 1). This analysis basis consists of the following interconnected work items:

- informal system requirements (as stated for example in SUBSET-125),
- semi-formal ARCADIA models like Mode/State models, sequence diagrams and models for functional chain analysis, and
- formal models as basis for formal verification related to consistency and completeness issues.

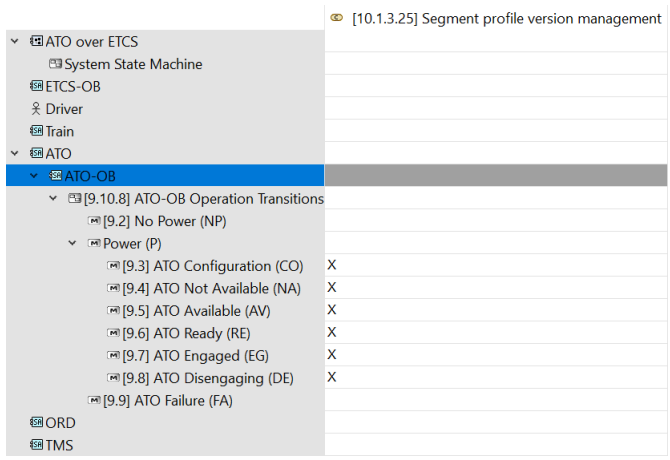

Figure 3: ARCADIA capabilities matrix (excerpt).

Different development roles with different scopes are usually involved in the model-based system development as depicted in Fig. 1. System experts with their system expertise as well as domain experts can support requirements engineers to formulate and analyze system requirements on an informal level. The MBSE experts are needed to extract semi-formal ARCADIA models from the system requirements. A bidirectional traceability between system requirements and ARACDIA models enables the information transfer between requirements and models. For example, inconsistency or incompleteness issues identified in requirements can be related to models and model elements using the traceability information. Vice versa, issues identified in models can be related to system requirements. Test, verification and validation engineers can make use of requirements and models in order to create relevant documents like test plan, test report, verification plan, verification report, validation plan, and validation report. Furthermore, system requirements, especially those expressed by tabular specification methods, or specified as ARCADIA Mode/State models and sequence diagrams are useful for the derivation of appropriate system test cases.

3 SEMI-FORMAL ANALYSIS

The semi-formal analysis consists of the following steps:

- Review system requirements
- Establish bidirectional traceability
- Analyze system function activation
- Verify system properties
- Validate against stakeholder expectations

The analysis approach started with a review of the system requirements related to modes, mode transitions and mode properties of the ATO Onboard Unit (ATO-OB). An ARCADIA Mode/State Model was systematically extracted from the system requirements and a bidirectional traceability between requirements and model elements was established. Modes as well as transition conditions are enriched with traceability information. For example, mode “[9.6] ATO Ready (RE)” refers to Section 6.1 of SUBSET-125, which specifies the system requirements of ATO over ETCS. The transition from mode “[9.6]

```
TRACE [9.10.8, 5>-p5-]
   - section 9.10.8
   - transition condition 5
   - priority p5
CONDITION
  AND
   - Driver has selected 'ATO Engage'.
   - ETCS-OB is in AD Mode.
   - TBL is in neutral or traction position.
```

Figure 4: Transition condition "ATO ready" to "ATO engaged" (excerpt).

ATO Ready (RE)" to mode "[9.7] ATO Engaged (EG)" is enabled whenever the transition condition 4 is satisfied.

Example The traceability condition of transition from mode "ATO Ready" to "ATO Engaged", shown in Fig. 4, refers to Section 9.10.8 of SUBSET-125. The flag `5>-p5` refers to transition condition 5 specified in Section 9.10.8, which, whenever enabled, is carried out with priority 5 ("p5").

The system requirements also specify which functions can be activated a mode. To analyze these activation requirements a so-called capability matrix, a special ARCADIA model, was extracted from the system requirements and analyzed w.r.t. completeness and consistency (see Fig. 3).

The Mode/State Model can be used to verify system properties as well as to validate the model against stakeholder expectations.

4 FORMAL ANALYSIS

The systematic formalization transforms in several steps system requirements as well as the extracted ARCADIA Mode/State Model into a formal Event-B specification. The formalization approach consists of the following steps (see Fig. 5):

- Identify components
- Derive function tables
- Derive variable table
- Derive Event-B specification
- Validation
- Verification

4.1 Identify components

In step *Identify components* components (system, environment) specified in SUBSET-125 are systematically identified. Since most of the requirements in SUBSET-125 are related to the ATO-OB the decision was made to choose the ATO-OB as system. As can bee seen in the reference architecture (see Fig. 1) the following environmental components were identified: Driver, Train, ETCS-OB, ATO-TS, ORD. In step *Derive function tables* conditions and actions related to modes like *ATO Engaged*, *ATO Disengaging* of the ATO-OB and its associated mode transitions and transition conditions (see Fig. 4) will be identified

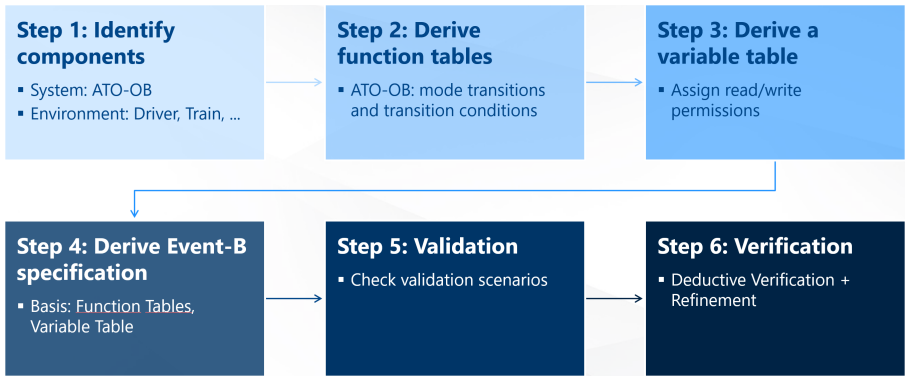

Figure 5: Formal analysis.

and extracted. All conditions and actions were placed in a special table called function table. Function tables were proposed by Dave Parnas (see [11]). Subsequently, traceability information links the requirements with elements of the function tables. A coverage analysis has been performed to ensure that all relevant requirements have been covered in the function tables.

4.2 Derive variable table

In step *Derive variable table* variables in the conditions and actions were identified so that conditions and actions can be specified formally. After analyzing the requirements types (like bool or enumeration types) were assigned to the identified variables. This step involves the analysis of relevant interface specifications SUBSET-139, SUBSET130, SUBSET-126, ... (see reference architecture Fig. 1), especially the interface between ATO-OB and ETCS-OB and the interface between ATO-OB and train, respectively. For each variable and for each component (system, environment) read/write permissions were assigned. traceability information related to both the interface and system requirements and events of the Event-B specification were added.

4.3 Derive Event-B specification

In step *Derive Event-B specification* an Event-B specification was derived on the basis of function tables and identified variables. Each Column of a function table was represented as an event in the Event-B Specification. Mode properties were formalized as Event-B invariants. The derived Event-B Specification has 25 variables, 30 invariants and 41 events.

Example In Fig. 6 the formal Event-B transition from mode "[9.6] ATO Ready" to mode "[9.7] ATO Engaged" ([9.6] and [9.7] are traceability information related to SUBSET-125). The transition can also be found in the ARCADIA Mode/State model.

4.4 Validation

In step *Validation* the use scenarios specified in SUBSET-125 were executed step-by-step with the ProB tool (see for example [12] for an application of ProB in the railway domain).

```
ato_ob_mode_re_6
any
  ato_ob
where
  ato_ob ∈ ATO_OB
  ato_ob_mode(ato_ob) = RE
  ato_ob_operational_conditions(ato_ob) = TRUE
  ato_ob_engagement_conditions(ato_ob) = TRUE
  ato_ob_etcs_ob_in_ad_mode(ato_ob) = TRUE
  ...
then
  ato_ob_mode(ato_ob) = EG
  ...
end
```

Figure 6: Event-B Transition from mode “ATO Ready” to mode “ATO Engaged”.

For example, Section 9.10.2 of SUBSET-125 describes the nominal scenario in which the ATO will be engaged by the driver for each so-called journey segment.

4.5 Verification

In step *Verification* theorem prover of Rodin, SMT solvers and ProB plugins were used for the formal verification. Deductive verification (see proof tree depicted in Fig. 7) was the main proof technique used in the formal verification. Several critical safety properties could be verified by refinement steps and additional constraints on the environment. For example, it could be formally verified that the ATO-OB is never in mode *ATO Engaged (EG)* or *ATO Disengaging (DE)* (the only modes in which ATO is engaged), whenever the driver never selects “ATO Engage”.

The formalization concept and formal analysis approach has been described in more detail in Eschbach [13].

5 RESULTS

Different stakeholders are usually involved in the overall systems engineering process. All of these stakeholders have different views on the system to be designed. All of them use different languages to speak about the system to be designed and to understand the corresponding system properties. Model-based systems and architecture engineering helps to bring all of these different views and languages together by providing common models as a basis for communication and alignment between different stakeholders. Since this common basis should be understandable to all stakeholders these models often lack formality, but formality is needed to ensure completeness and consistency in a mathematical sense. In the project described in this paper the formal method Event-B was used to specify a mathematical model and to formally verify properties of this model. The overall idea was to create a formal model closely related to some relevant ARCADIA models in order to ensure their completeness and consistency. A bidirectional traceability relates the semi-formal ARCADIA models with the formal model and allowed a transfer between the semi-formal and the formal layer. Especially issues regarding consistency and completeness found in the formal model could be transferred to the semi-formal ARCADIA models. The formal analysis revealed 23 ambiguities and

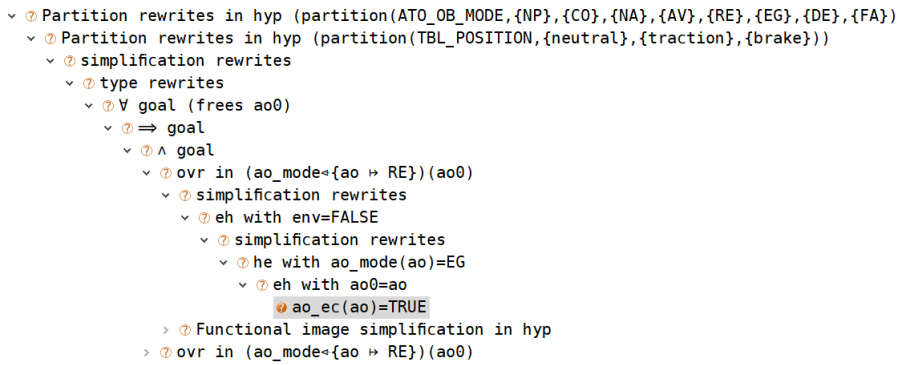

Figure 7: Deductive verification: Proof tree shown in Rodin.

missing requirements. Furthermore, a critical mode invariant violation has been discovered and was transferred to the ARCADIA models, successfully.

6 CONCLUSION

In this paper we have presented the experiences gained in the application of a model-based system and architecture analysis based on informal system requirements, the ARCADIA method and the formal method Event-B. We have shown how a bidirectional traceability between system requirements, ARCADIA models and model elements, and Event-B specifications can be used to involve different stakeholders and technical experts within the overall model-based system and software engineering development. Different levels of formality and a strong traceability between these levels helps to transfer knowledge and insights between stakeholders, technical experts and other roles involved in the overall system and software development. The formal analysis of the mode transition requirements revealed 23 ambiguities and missing requirements and will be continued in the future. Further ATO functions related to journey and segment profiles will be addressed. The current focus is on Grade of Automation Level 2 (GoA2), but will be extended the highest level of automation, namely Grade of Automation Level 4 where more sophisticated sensor technology and innovative algorithms are required.

ACKNOWLEDGEMENT

I thank the anonymous reviewers for their valuable suggestions to improve the paper.

REFERENCES

[1] Basile, D., ter Beek, M.H., Fantechi, A., Ferrari, A., Gnesi, S., Masullo, L., Mazzanti, F., Piattino, A. & Trentini, D., Designing a demonstrator of formal methods for railways infrastructure managers. *Leveraging Applications of Formal Methods, Verification and Validation: Applications*, eds T. Margaria & B. Steffen, Springer International Publishing, volume 12478 of *Lecture Notes in Computer Science*, pp. 467–485, 2020.

[2] Graves, H. & Bijan, Y., Using formal methods with SysML in aerospace design and engineering. *Annals of Mathematics and Artificial Intelligence*, **63**(1), pp. 53–102, 2011.

[3] Pétin, J.F., Evrot, D., Morel, G. & Lamy, P., Combining sysml and formal methods for safety requirements verification. *22nd International Conference on Software and Systems Engineering and Their Applications*, p. 11, 2010.

[4] Ahlbrecht, A. & Durak, U., Integrating safety into mbse processes with formal methods. *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, pp. 1–9, 2021.
[5] ter Beek, M.H., Borälv, A., Fantechi, A., Ferrari, A., Gnesi, S., Löfving, C. & Mazzanti, F., Adopting formal methods in an industrial setting: The railways case. *Formal Methods: The Next 30 Years*, eds M.H. ter Beek, A. McIver & J.N. Oliveira, Springer International Publishing, volume 11800 of *Lecture Notes in Computer Science*, pp. 762–772, 2019.
[6] Basile, D., ter Beek, M.H., Fantechi, A., Gnesi, S., Mazzanti, F., Piattino, A., Trentini, D. & Ferrari, A., On the industrial uptake of formal methods in the railway domain: A survey with stakeholders. *Integrated Formal Methods*, eds C.A. Furia & K. Winter, Springer International Publishing, volume 11023 of *Lecture Notes in Computer Science*, pp. 20–29, 2018.
[7] Fitzgerald, J., Bryans, J. & Payne, R., A formal model-based approach to engineering systems-of-systems. *Collaborative Networks in the Internet of Services*, eds L.M. Camarinha-Matos, L. Xu & H. Afsarmanesh, Springer: Berlin, Heidelberg, pp. 53–62, 2012.
[8] EN 50155:2021 Railway applications – Rolling stock – Electronic equipment, 2021.
[9] EN 50657:2017 Railways applications – Rolling stock applications – Software on board rolling stock, 2017.
[10] EN 50128:2011/A2:2020 Railway applications – Communication, signalling and processing systems – Software for railway control and protection systems, 2020.
[11] Parnas, D.L., Inspection of safety-critical software using program-function tables. *Linkage and Developing Countries, Information Processing '94, Volume 3, Proceedings of the IFIP 13th World Computer Congress*, Hamburg, Germany, 28 Aug.–2 Sep., IFIP Transactions, pp. 270–277, 1994.
[12] Hansen, D., Leuschel, M., Schneider, D., Krings, S., Körner, P., Naulin, T., Nayeri, N. & Skowron, F., Using a formal B model at runtime in a demonstration of the ETCS hybrid level 3 concept with real trains. *Abstract State Machines, Alloy, B, TLA, VDM, and Z*, eds M. Butler, A. Raschke, T.S. Hoang & K. Reichl, Springer International Publishing, pp. 292–306, 2018.
[13] Eschbach, R., Formalizing and analyzing system requirements of automatic train operation over ETCS using Event-B. *Rigorous State-Based Methods*, eds A. Raschke & D. Méry, Springer International Publishing, volume 12709 of *Lecture Notes in Computer Science*, pp. 137–142, 2021.