Chapter 1

Introduction

The thesis starts with a brief introduction of automotive systems and the importance of embedded software in modern automobiles. An outline of challenges in the development of embedded automotive software is given. Next, the topics of the thesis, the key issues in the development and the contributions of this thesis are stated. The chapter closes with an outline of the thesis and a list of publications, in which aspects and intermediate development stages of the presented work are published.

1.1 Motivation

The history of automobiles started on January 29th 1886 with patent number 37435, granted by the *Kaiserliche Patentamt des deutschen Reiches* [14]. The patent was issued for the invention of a *Fahrzeug mit Gasmotorenbetrieb*, i.e. a vehicle with gas engine. This date can be considered as the birthday of cars which have changed the world in a significant way.
As automobiles have changed people's life, automobiles themselves have evolved tremendously compared to the first one as depicted in Figure 1.1. Modern cars embed complex distributed real time systems in order to realize even more safe, comfortable, and economical vehicles. The value of the installed electronics per automobile was in the year 2000, on a world wide average, 155 USD. In 2020, it is expected to be 590 USD in an average car [39].

In 2002, software and electronics already led to 40% of the manufacturing costs of an automobile. The availability of increasingly powerful and cheaper hardware makes the implementation of increasingly elaborated functionality in mass production possible. The system's complexity increases accordingly.

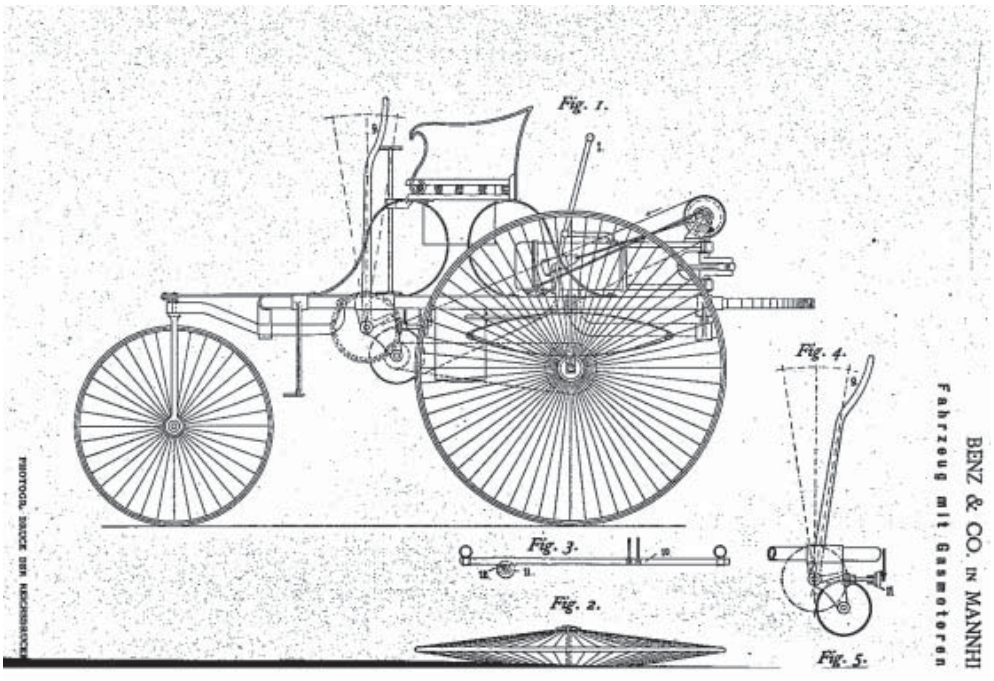

Figure 1.1: Detail of Patent No. 37435: Fahrzeug mit Gasmotorenbetrieb

Some figures about the current Audi A8: The current premium upper class automobile has 270 MB of embedded functional software. The software is distributed on 100 electronic control units (ECUs), connected with 7 bus systems. 2370 wires are installed to connect the ECUs and the associated devices.

The ratio of innovations realized with software in the automotive industry has raised from 20% to 85% over the last years [53] and is still growing [78, 115]. Ferdinand Dudenhöffer, Director of the Center of Automotive Research (CAR) of the FH Gelsenkirchen in 2002 [68], stated that electricity and electronics provide the base for 90% of all innovations in automobiles in the near future.

As a result, the importance of the software embedded in automobiles to customers and hence market shares will be vital. In [24] the conclusion is that this development establishes software as a key technology in the automotive domain. And only software enables the car manufacturers to integrate functionality for the customer to use virtually any electronic device like smart phones and laptops in combination with services provided by the car. Each device that can interact with the car adds another possible configuration to the possible ones of the car and increases the configuration space of the embedded software.

This thesis focuses on software aspects of embedded systems. Software is expected to have the greatest impact on customers, industry and market shares

in the automotive industry [95, 26]. The example in Figure 1.2 illustrates the development of modern automobiles to complex systems with integrated hardware and software to present the advances achieved compared with the first automobile in Figure 1.1.

Figure 1.2: Inner View of Components of Modern Automobile

The realization of innovations in modern vehicles is only possible by a shift from electronics to software [95].

Software development offers flexibility to build more variants and, moreover, potential to reduce development time and costs. The standards released by standardization committees like the Association for Standardization of Automation and Measuring Systems (ASAM) allow for the reuse of artifacts in different variants. However, as safety critical functionality is realized by software, it plays a critical role and a software failure can result in expensive damage to the hardware or traffic participants. As a consequence of this, the quality assurance of the entire development process, and of the resulting software is of vital interest.

However, 44% of the embedded system designs meet only 20% of the functionality and performance expectations [53]. In 2005, Dr. John J. Wargin from Hewlett-Packard stated that 60% to 70% of the warranty efforts trace back to software failures [141]. A lack of appropriate test approaches for functional validation and verification contributes to this figures. Thus, a method to improve

the quality of the development process starting from the first models which specify a functionality or system is required. The method should facilitate a systematic and structured test derivation in different development phases of the system. Automation of steps is additionally targeted as it can substantially reduce the costs of development [86].

The issues for a method that should cope with these requirements are extrapolated in Section 1.2.

1.2 Challenges to the Automotive Domain

The integration of increasingly complex and distributed functionality in automobiles is inevitable. The main driving forces are customer demand for new features, legal stipulations, and the integration of novel techniques. New power train techniques require, especially because of safety aspects, advanced and distributed systems for monitoring and control. The development of the embedded software to realize the functionalities is one of the greatest challenges in the automotive domain [80].

The verification and testing of the implemented functionality evolve into issues of high importance. The rising complexity, especially caused by the growing interconnectedness of the integrated functionality, makes extensive verification often impossible, because time and resources which are available are often not sufficient to test all possible scenarios [156]. The impracticality to test everything is intensified by the possible configuration space of modern vehicles [95]. Measuring the complexity in possible bit combinations during runtime of the embedded software the complexity does not grow linearly but exponentially.

The existing norms for functional safety were adapted in order to account for the increasing implementation of software for safety critical systems in vehicles. The upcoming international standard ISO-26262 [59] is developed based on the IEC 61508 for the "Functional safety of electrical/electronic/programmable electronic safety-related systems" [41, 31]. The ISO-26262, which is titled "Road vehicles – Functional safety" explicitly addresses the safety of systems embedded in road vehicles. It states requirements on the methods applied in the development as well as during the whole life cycle of products that integrate safety-related systems. Required activities are assigned according to the risk classes, i.e. *Automotive Safety Integrity Levels* (ASIL A - D). Level A is the lowest level, and level D is the hightest of safety criticality. The MISRA guidelines [83] substantiate this by requirements on the use of model based methods for safety critical software and hardware.

The AUTOSAR (AUTomotive Open System ARchitecture) [8] was developed to ease the development of embedded software for automobiles. AUTOSAR specifies an open and standardized automotive software architecture, which was jointly developed by automotive OEMs (Original Equipment Manufacturer), suppliers, and tool vendors. The objective is to create and establish open and common standards for automotive software architectures. The latest AUTOSAR releases integrate advanced functionalities like energy management, diagnosis, and comfort, whereby the first AUTOSAR releases accounted only for basic system functionalities.

The currently latest AUTOSAR 4.0 [8] specification explicitly accounts for safety as design aspect. It allows for the partitioning of hardware and software for ASIL coexistence. It accounts for end-to-end verification and supports the modeling of multi-core architectures. AUTOSAR 4.0 provides modeling capabilities to improve the specification process in order to be able to do early validation and partly automated code generation. The improved AUTOSAR architecture allows for modular design and reuse of well proven legacy components. This enhancement is necessary as embedded systems become increasingly complex and distributed [80]. Moreover, the number of system variants and configuration possibilities increases. Modular design and portability between different hardware platforms are key requirements for the design to come up to this development.

The aforementioned ISO 26262 [59] covers functional safety aspects during the development process of safety critical functionality. It also covers quality assurance activities during the whole life-cycle of a product with safety critical functionality. The ISO 26262 includes activities for the steps of specification, design, implementation, and verification. Functional safety is defined as a system property. This implies that the fulfillment of the requirements by a component or module on its own is not sufficient. Safety critical functionality is usually realized with systems consisting of many components such as sensors, processing units, and actuators. Hence the fulfillment of a safety goal requires different properties of all components that constitute the functionality. Properties of components may only be valid under assumptions that were made about the environment or other involved components. Taking the whole configuration space into consideration it is hard to answer which configurations should be used for testing according to ISO 26262.

This implies that components are, considering the ISO 26262, not as independent as the AUTOSAR standard suggests. OEMs tend, however, to increase the

number of variants and of possible configurations. The AUTOSAR standard itself neither provides suitable tools nor a method to validate and verify properties of components in changing system designs [84]. It provides only guidelines to check the conformance of model properties to the AUTOSAR standard.

No specific method for the development of embedded systems is commonly accepted as the way to do it. The methods, processes, and tools used for the development are usually hold proprietary by the companies. Some methods and practices are generally recognized as good, but there is no general set of best practices. Formal methods are used in some organizations, but due to their expenses, usually only for safety-critical applications. Rigorous methods are in general considered more cost-effective, and sequence based specification (SBS), which is the basis for the method presented in Section 4.2, is a rigorous method [104, 152, 73]. Nowadays, the processes used in industry combine formal, rigorous, and other methods. Good tool support exists and is widely used for Model Based Design (MBD) and automatic code generation from models. Automated testing on simulators (Hardware in the Loop [21], Software in the Loop [35], Model in the Loop [25, 67]) is also widely used in such organizations [116, 117].

The modern software engineering suggests to use test-driven development (TDD) to cope with an increasing software complexity [10]. The main principle behind TDD is to specify test cases before actually starting to take design decisions and to start the implementation. Two major issues of TDD are the maintenance of existing test cases under changing requirements and the handling of similar aspects for several test cases.
To ensure the properties of components in early design phases, it is necessary to be able to do early verification and validation of properties under changing environments and functional allocations. The increasing number of variants of automobiles causes a higher variability of the embedded software. This variability must be considered by the methods for specification and design [118]. Functionalities may be removed or changed at different levels of abstraction to realize a feature for a requirement of a variant.

The methods employed for specification, modeling, and verification must cope with quickly changing design decisions as they are common in early design stages. Moreover, it should be possible to take these models as basis for verification activities in later development phases. Testing as method for verification and validation is most useful for automotive OEMs, because it can still be applied

when parts of the system cannot be formally modeled, especially in the case of physical devices, are proprietary, or are hard to formalize by a reliable and valid model.

1.3 Scope of the thesis

A brief recapitulation of the current problem and context of the thesis is given. The issues handled within this thesis are presented and the contribution and research findings are motivated. Then the structure of the thesis is outlined, providing a road-map for the reader.

1.3.1 Problem Statement

The early specification of requirements in the development process helps to clarify the needs and to reduce misunderstandings. Yet, this is only rarely done using specifications written in formal syntax, but in natural language or sketches. Agile development approaches like TDD propose the early description of use cases by test cases in order to verify even the first implementation models against the requirements.

Yet, a complete and formal requirement specification as an artifact for the whole development process is still missing. The development and validation of increasingly context-aware sensor based functionalities is increasingly complex with the rising amount of possible data and timing constellations. Additionally, testing such systems requires even more complex environments to be set up in order to evaluate the system behavior in e.g. safety critical situations.
A method is needed that allows for the early description of the requirements in a formal and consistent manner. Test management indicators and quality metrics to support the definition of testing and acceptance criteria should be definable on the basis of the requirements description.

Nowadays, model-based design is the established way in the automotive domain for the prototyping and analyzing of embedded software systems. Widespread design tools such as MATLAB/Simulink [77], NI LabVIEW [87] or ETAS ASCET [37] are used for the graphical modeling and automatic code generation of platform independent or dependent code. These tools make it possible to do early prototyping and to decrease the entire development time. Rapid design methods should be used in addition and not as substitution to sound design. As the requirements specification evolves and first implementations are generated, the necessity increases that the complete behavior meets the intended one and the final result fulfills the intentions with sufficient quality.

One important criterion is dependability. Software dependability comprises *risk*, *safety*, and *reliability* [50]. System dependability is not only based upon software dependability. Hence during hardware software co-design the interference between physical and virtual dependability must be considered. The software design can explicitly account for dependability issues of the underlying physics. For sound dependability analysis, the results of test cases on the target hardware must be taken. Finally, the entire system with the real hardware must be used for dependability evaluation.

The method used for specification and verification must support the evaluation of dependability from the first implementation models of components to the entire distributed system. Agile methods such as TDD should be used complementary.

1.3.2 Contribution

The thesis deals with model based requirement specification and model driven testing in the development of software. The main focus is on the development of embedded and distributed software for automobiles.

A key challenge in the development of embedded software is the assurance of the quality. Testing is a key activity to control and assure the quality. Hence, the thesis's main contribution, though the word "quality" may not be explicitly stated in many sections, is about the quality of software.

In this thesis, a method is described for the modeling of requirements and test case derivation that has been successfully applied in the pre-development and integration testing of an automotive OEM.

The following issues are researched in this thesis:

1. How can requirements be reflected in the test model?
2. How can quality properties of the test model be achieved in industrial practice?
3. What should be the method to create the test model?
4. What is the role of the test model to control the quality of the system?
5. What aspects should be reflected by the test model and what elements should it include?

6. How can time and timing aspects be reflected by the test model?
7. How should discrete and continuous signals be handled by the models?
8. Should test models be combined with information from system models?
9. Where is it recommendable to combine reference models as oracles with test models?
10. Should the same language be used for the test model, system models, and oracle models?
11. What steps can be automated by the use of models?
12. What steps should not be automated?
13. What criteria can be used to assess the quality of a test case?
14. What aspects should be considered in the test strategy?
15. What algorithms should be used for test case derivation?
16. How can significant test cases be derived?
17. What are useful test metrics and indicators?
18. How can the quality of the system under test be assessed?
19. Which indicators serve as test ending criteria?

The main contributions of this thesis can be divided into the following main findings:

1. An enhanced modeling paradigm on the basis of an established modeling principle in order to be able to reflect time and timing issues by the test model. The subsequent activities like test planning and test case generation profit from the new information.
2. A method to create the test model. The method should be applicable in practice and facilitate the achievement of quality criteria to a high degree. The method for testing of functional and non-functional requirements as it allows for the consideration of dynamic dependencies of data and time. Design patterns were developed for the application in integration testing on HIL simulators.

3. Useful test management indicators in practice. New computations on the basis of semi-Markov process were developed that make it possible to integrate timing aspects for the test management and assessment of the test activities.

4. A combination of reference models with the test model to account for testing of continuous signals.

5. Test case generation strategies. The strategies were developed to address the testing aims in the automotive domain throughout the whole development process. The strategies facilitate the automated generation of significant test cases.

The test models can be regarded as specification of the functional test requirements and serve as basis for all subsequent test activities. Test management indicators, that integrate time, can be directly computed from the test model. Model driven test case generation is possible on the basis of these models to derive test cases.

The method was developed for the application throughout the whole development process. Several projects were conducted that proved the benefits of the developed method in the early development phase of Model in the Loop testing as well as in integration testing phase on Hardware in the Loop simulators.

1.3.3 Outline

This chapter gives a general idea of the contents of the thesis. It introduces the background, the issues tackled within the thesis and the objectives.
The remainder of the thesis is structured as follows.

In Chapter 2, fundamental terms and concepts for this thesis are introduced. The relation between testing objectives, quality, and confidence in a system under development (SUD) is set up. The main test phases in a typical development cycle and testing methods for software are described. Established approaches, regarding the automotive domain, for testing of embedded systems are outlined. Next, another approach for testing of embedded systems is presented. The last presented approach is testing with Markov chain usage models (MCUM) in Subsection 2.5.4, which is one foundation for the work presented in this thesis. The chapter concludes with a summary of the goals and the own contribution of the thesis.

The Time Usage Model (TUM) is introduced in Chapter 3. TUMs allow the explicit integration of timing in the test model. First a definition is given, followed by a guide to the modeling concepts of timing aspects such as durations and timing dependencies in the model. The role of the model in the development process, starting from the requirement specification to the test activities, is outlined.

In the subsequent Chapters 4 to 6 the appliance of the TUM is described in detail. Chapter 4 starts with method how to create the TUM on the basis of the requirement specification. Requirements analysis and specification are treated, followed by the principles behind the method to create the TUM. The extended method to create a TUM with explicited consideration of timing requirements is presented with the example of a seat belt reminder functionality. Design patterns were developed for the appliance in the industrial context. The design patterns are presented within the industrial case studies in Section 8.2.3.

The next step with the TUM in the development process is the assessment of the model with metrics and planning of test activities with management indicators. Metrics and management indicators can be automatically computed on the basis of the model. Chapter 5 gives an overview of classic metrics and indicators that can be computed on the basis of the underlying Markov chain. These indicators can be used before test case execution to assess the test effort to reach a level of confidence and to plan the testing. After test case execution they can be used to assess the quality of the system under test (SUT), serving as a stopping criterion for testing. Next, new metrics and indicators that integrate time information are presented. The mapping of the model elements to a semi-Markov process (SMP) is described and the computations for the new metrics and indicators.

Having analyzed the model with metrics and determined the test goals, test derivation strategies can be used to generate automatically test cases. Strategies control the generation to derive test cases according to criteria. Criteria can be determined by test purpose, test strategy, and test resource constraints. Chapter 6 presents algorithms for the automated generation of test cases from TUMs. Novel algorithms were developed for generation to explicitly integrate the timing information. These novel algorithms allow for the testing requirements of the automotive domain and safety critical systems.

The complexity of the systems under development makes the evaluation and assessment of the behavior of the system even more complex. The growth of information needed for this comes along with it. The integration of this informa-

tion into the TUM which is used for the generation of test cases is hardly feasible. In Chapter 7 it is presented how reference models can be used in combination with the test model for this task. The reference models are used like an executable specification, providing information for the evaluation of the system to be tested. The test model can therefore be kept generic in order to be able to derive virtually any possible usage scenario and therefore test case from the model.

The appliance of the methods described in Chapters 4 to 6 is presented in Chapter 8. The approach was applied at a large German car manufacturer in both the early development phase and the system integration phase of embedded systems.

The requirement specification of active safety functionalities were modeled in the pre-development phase in TUMs. Test cases were generated from the TUMs to assess the implementation models in MATLAB/Simulink. The test cases were executed in MATLAB/Simulink in Model in the Loop simulation. In the system integration phase an example project from the energy management is presented. Test cases were derived from the model and automattically executed with EXAM (EXtended Automation Method) on Hardware in the Loop simulators. In both applications benefits could be achieved by the presented approach.

Conclusions are summarized in Chapter 9. The achieved results and capabilities are summarized and the benefits are recalled. The thesis concludes with an outlook in Chapter 10. Perspectives for future research and development of the presented method are given.

1.3.4 Publications

The thesis bases on several publications in which intermediate development stages and different aspects of the presented work are presented. A list of these publications is given in the following.

- In [128], preliminary work for this thesis is summarized that has been conducted in the diploma thesis in [123].
- First findings about the requirement from the automotive domain to integrate time systematically into the test model are presented in [129]. The deficiency of classic MCUMs as test model is addressed by first considerations and a case study to integrate time in the test model.
- Modeling concepts to reflect time and timing dependencies in extended usage models are presented in [131]. Algorithms for test case generation based on the new models are introduced.

- Metrics and measures that can be computed on the new model are presented in [130]. The presented solutions are based on the concepts for system performance analysis of communication systems [42].

- In [133], the enhancement of EXAM [64], which is the test automation on Hardware in the Loop simulator across the Volkswagen group, with TUMs is presented. The extended architecture is presented and the layers are described. With this extension TUMs can be used for the generation of automated executable test case in EXAM. First projects are briefly described.

- The results from a project to specify and systematically test the decision making of the energy management and engine control with TUMs is presented in [125].

- Design patterns have been developed for the creation of TUMs in the industrial context. The design patterns take into account the idea of the rigid enumeration of all possible input sequences for the creation of the model. The requirements for the test model and appliance of the design patterns is presented in [132].

- A summarized presentation of the appliance of TUMs for system integration testing with EXAM is given in [134]. It is embedded in the automotive context and the relation to other established approaches is given. Preliminary results from the appliance of TUMs in Model in the Loop Testing are presented.

- The results from the appliance of TUMs in the pre-development stage of active safety functionalities are presented in [124]. The modeling approach for the early test of implementation models is described as well as the test automation ASTunit for Model in the Loop testing. Results from the application of TUMs for an active safety project are summarized.

- In [127], a direction for future work is presented. The idea is to combine internal component dependency diagrams with the TUM. In doing so, internal dependences can be used for the test case derivation from TUMs. The presented work was entitled with the *Best Paper Award*.