Part I.

Introduction

In the development of modern vehicles, the infotainment system [54] belongs to the innovative area. In comparison to the conventional areas such as the motor, body construction and drive train, the infotainment system gains ever increasing influence on the appearance of a vehicle in the eyes of both customers and competitors [119]. Infotainment systems of current premium vehicles are highly integrated and distributed systems and provide a variety of information, entertainment and communication functions with the help of numerous electronic control units (ECUs) which are connected to each other via bus systems [86].

Figure 0.1 shows the infotainment system for the car model A8.

Figure 0.1.: Automotive infotainment system
1.GUI and central display 2.control unit 3.touch pad
4.turn-press button 5.multi-function-wheel 6.speech button

A human machine interface, abbreviated as HMI, is the composition of interfaces through which the user communicates with the machine. In the context of infotainment systems, an HMI can include (as shown in Figure 0.1) a graphical user interface (GUI) on the central display, a touch pad, buttons, a turn-press button, a multi-function wheel and speech input and output facilities. In the A8, the touch pad, buttons and turn-press button are integrated into one unit, known as the control unit. In many premium vehicles, the HMI also contains a graphical interface on the instrument cluster; this is not shown in Figure 0.1.

In spite of this general definition, the term “HMI” has a specific meaning in the context of HMI development in the area of automotive infotainment systems. In this context, the HMI does not mean the hardware interfaces as introduced above, but the software of the graphical user interface. In most of the current research in this area, the term “HMI” is a synonym for the graphical user interface of the central display [112] [45] [54].

In this thesis, the term “HMI” is assumed to have that same meaning. This means that the goal of this thesis is to test the graphical user interface on the central display. Both the graphical representation and the dynamic behavior of the HMI are focal points of this thesis. The HMI to be tested is modality-independent. This means the events to which the HMI reacts can be triggered by different input sources. Furthermore, the GUI of the instrument cluster and consequently the synchronization between the instrument cluster and the central display are not included in the scope of this thesis.

The HMI of a current premium infotainment system can be a huge system. An Audi HMI currently in development contains up to 2200 screens, 100 pop-up menus, up to 10000 texts per language and a very complex dynamic behavior. For instance, the description of the dynamic behavior can require 200 diagrams, 50 hierarchies, 2400 view states and 14000 transitions. In [45], it was declared that up to 4000 UI elements such as screens, buttons, lists and messages are used for the BMW 7-Series.

Furthermore, the HMI of a premium vehicle usually provides a large set of variants that are a result of differing car models, markets, languages and system hardware [16]. Variants can also be caused by individual combinations of functions, e.g. in some variants the navigation function is integrated; in others it is not. Figure 0.2 presents an example of two different HMI variants. In practice, more than 100 variants are possible. This variability of HMIs results in very high testing complexity.

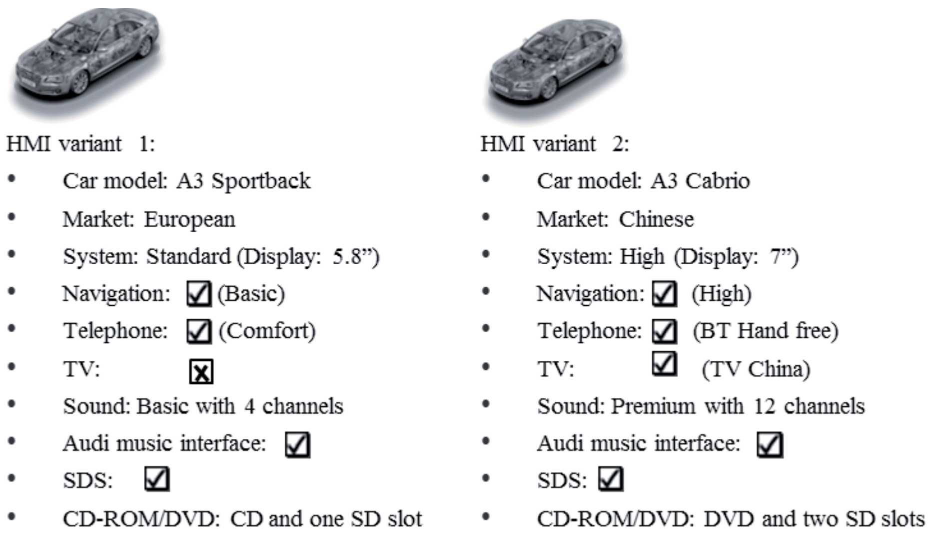

Figure 0.2.: An example of two HMI variants

Chapter 1.

Problem and Solution Summary

This thesis addresses the model-based testing of infotainment system HMIs with the particular aim of modeling and testing variability of the HMIs.

Current problems will be introduced from an industrial perspective as well as a research perspective. Afterwards, an overview of the solution will be presented.

1.1. Problem summary

As mentioned above, the HMI of a current infotainment system can contain up to 2200 screens, 100 pop-up menus, 10000 texts per language and a very complex dynamic behavior, the description of which requires up to 200 diagrams, 50 hierarchies, 2400 states and 14000 transitions. More than 100 variants can be caused by different car models, markets, languages, system hardware types and individual function configurations. HMI variability necessitates very high specification and especially testing efforts.

Current situation

Currently, HMI tests are performed almost entirely manually in the serial development [27] [112]. Once a test specification has been created for one project, it is usually manually adapted for later projects. In this process, defining new test cases and increasing test coverage are usually ignored due to limited time. Often, only the most important tests, which are usually a subset of the tests described in the test specification, can be executed. Manual testing is very time-consuming and expensive. First of all, it is quite difficult to achieve systematic and high test coverage via manual testing. Currently, increasing test coverage of infotainment system HMIs is a common goal of many manufacturers, especially when it comes to test coverage of foreign language systems [27]. An internal statistic of a current infotainment system project has shown that, although the HMI is only one small component out of many ECUs and bus systems in an infotainment system, it produces about 10% of the errors. This statistic is based on the internal evaluation of error tickets, which are randomly chosen from different developmental phases. This result should provide an indication of the profitability of automating HMI tests and increasing test coverage.

Problems in the industry

Model-based testing [42] [105] [62], or MBT in short, makes possible automatic test generation and systematic test coverage. In combination with automated test execution, which is usually called test automation [13], MBT provides a reduction in human resource requirements and costs. Model-based testing of application functions is well established in industry today. However, model-based testing of infotainment system HMIs is still very rare.

The most efficient method of MBT is to verify the System Under Test (SUT) against tests generated from the specification. However, most HMI specifications are still informal, even though formal specifications are more precise and can be reused for development. Efforts to establish formal specifications [37] have failed because the tools used are based on concepts designed for development and are not intended for specification. Development tools support the complexity of code generation, whereas a specification should describe the requirements in abstract form only. The complexity involved in using development tools thus leads to a failure to create formal specifications.

Tests can also be generated from the system model (Section 2.1). However, a system model is created for generating software code. It usually cannot be directly used for test generation due to a lack of test data. The necessary test data must be extended in advance if a system model is to be used for test generation. In this case, the code generator is validated against the test generator. Non-conformance between the implementation and the specification cannot be detected. Therefore, MBT is meaningful only if the test model and the system model are different.

Furthermore, an HMI testing concept that answers fundamental questions is still missing in the industry. For instance, the issues regarding which HMI errors should and can be dealt with, how to model the HMI for testing purposes, how to model variability and test different variants of the HMI, etc., have only recently been addressed [65].

Finally, variability is a new challenge for the development of infotainment system HMIs. The industry is still busy with variability development, e.g. in optimizing the development process and extending the development tools. The next step, i.e. testing under consideration of variability has not yet become a focus.

Problems in the research

The research topic of this thesis spans the domains of GUI, model-based testing and variability, in which the GUI plays a central role. Related work can be found in the 4 domains which are presented in Figure 1.1.

There are currently a number of research efforts that address the model-based testing of GUI applications [84] [79] [104] [96] [20] [89]. The GUI applications in question are mostly standard PC applications such as a calendar which contains all the necessary logic and data within the application. In comparison to standard GUI applications for PCs, an infotainment system has many special characteristics such as embedding and variability which do not exist in standard GUI applications and therefore are not considered for testing. The special characteristics of infotainment system HMI will be introduced in Section 2.2.5.

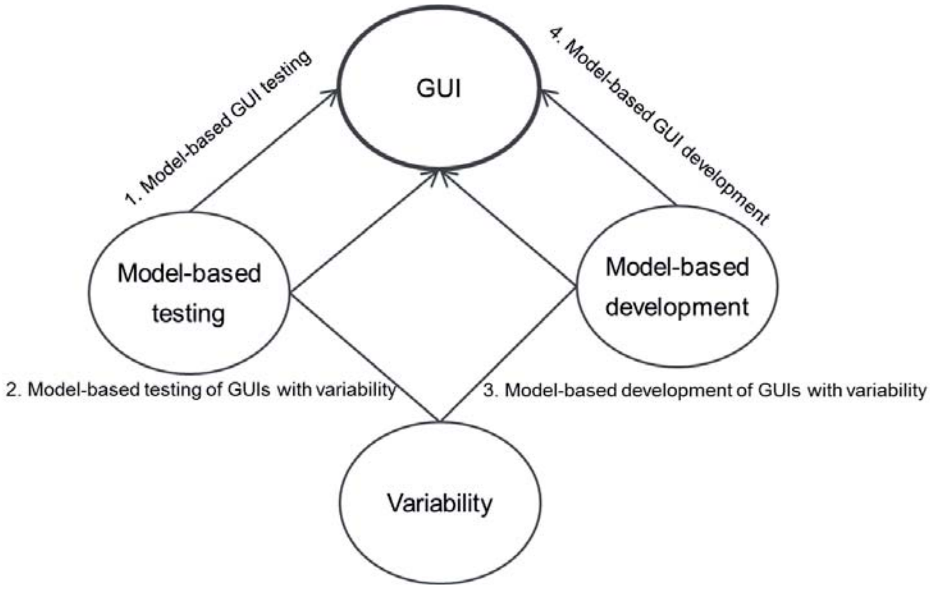

Figure 1.1.: Related research areas

A small amount of research efforts address the model-based testing of GUIs with variability. However, many of the problems we face in HMI testing under variability do not exist in these research areas and hence have not been resolved.

There is also a small amount of research work addressing model-based development of GUIs with variability e.g. [16]. Model-based development and model-based testing face common problems, but also different problems. For instance, model-based testing involves the additional task of identifying data required for the test generation (Chapter 5) and finding a suitable test generation method and adequate selection criteria (Chapter 6) etc. In particular, if the variability of HMIs is taken into account, new specification methods must be designed and testing must address the problem of redundancies. This issue will be introduced in Part III.

1.2. Solution summary

This thesis presents a model-based testing approach for infotainment system HMIs with the particular aim of considering variability. The approach can be divided into the following parts:

1.2.1. The framework for model-based HMI testing

In this section, we propose a model-based testing framework for infotainment system HMIs. Variability is not yet considered here.

First, a statistical analysis has been done in order to identify the possible HMI errors occurring in practice. Based on this result, HMI errors which can be automatically detected in the testing framework will be clarified.

We propose two basic components for the framework: the test-oriented HMI specification and the test generation as presented in Figure 1.2.

Figure 1.2.: Proposed model-based testing framework for infotainment system HMIs

The test-oriented HMI specification

A test-oriented HMI specification describes the expected behavior of the HMI with a layered structure and contains sufficient data in suitable forms to generate valid tests. The test-oriented HMI specification allows abstract specification of the HMI.

Test generation

It is not the focal point of this thesis to develop a new test generation method. A test generation algorithm is introduced in order to demonstrate how tests can be generated from the test-oriented HMI specification and what the tests look like. This test generation algorithm serves as a basis for the later generation method that takes variability into account.

1.2.2. Integrating variability into model-based testing

In this section, the variability of HMIs is taken into account. Two essential problems exist for testing under variability: how to model variability and how to generate tests for different variants without redundancies.

Modeling variability

Variability of HMIs can exist both in the dynamic behavior and the representation. Layers of the test-oriented HMI specification are extended to specify the commonalities and variabilities of different HMI variants based on the product line approach.

Test generation under variability

The test generation algorithm is extended to take account of the variabilities which are described in the test-oriented specification and to generate tests for different variants. It would be inefficient to perform a test generation for each variant to be tested. Therefore, a test generation algorithm that takes account of the variabilities is designed to avoid redundant test generation.

However, tests generated for different variants still include a large set of identical tests, which verify the commonality of different variants. A method is applied to automatically identify such tests and preserve them from redundant executions.

1.3. Organization of this thesis

This thesis comprises four parts as presented in Figure 1.3.

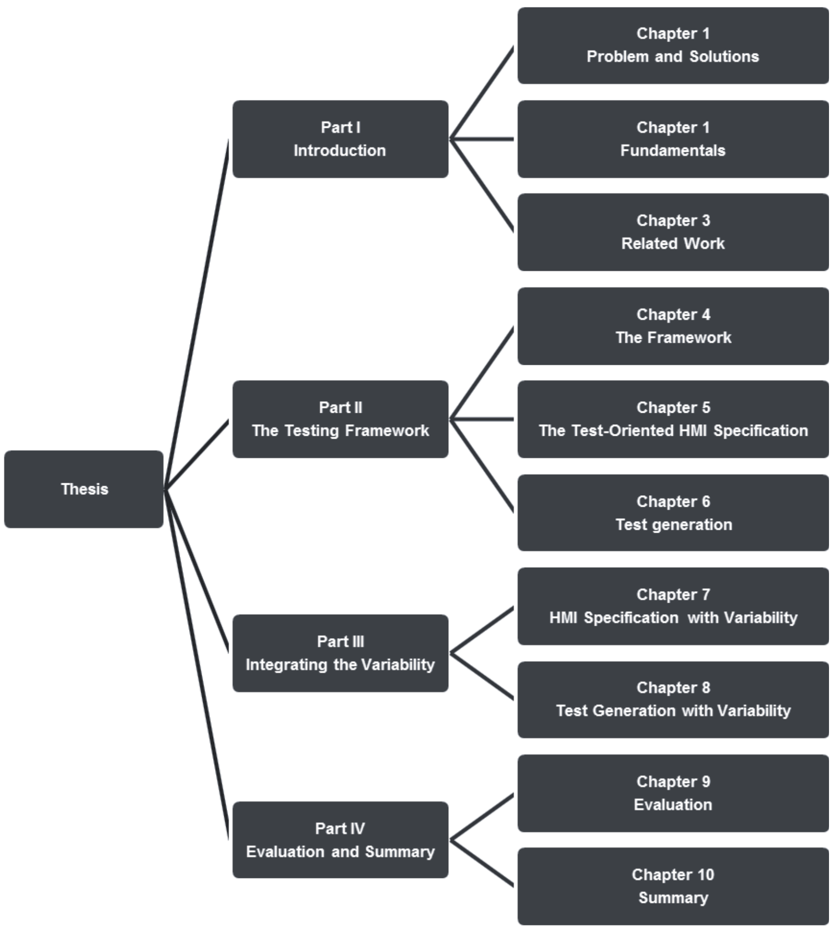

Figure 1.3.: Organization of this thesis

The first part includes Chapter 1 to 3. Chapter 1 has introduced the problems and solutions up to the present time. Chapter 2 gives an introduction to the fundamentals in the domains of model-based testing and automotive HMIs. Chapter 4 presents some of the related work, compares this with the proposed approach and explains why existing approaches are not suitable for testing infotainment system HMIs. References to other relevant work will be distributed throughout the later chapters due to the diversity of the sub-topics.

In the second part, we demonstrate the proposed testing framework. Here variability is not yet considered. In Chapter 4, components of the framework and the testing goals are presented. This thesis focuses on two types of tests: menu behavior tests verifying the dynamic behavior and screen tests verifying different classes of screen contents. Chapter 5 describes the requirements of the test-oriented HMI specification. Chapter 6 demonstrates a test generation algorithm and generated tests.

Part III focuses on variability. Chapter 7 introduces how to model variability. For modeling variability, each layer of the test-oriented HMI specification introduced in Chapter 7 will be extended. Chapter 8 demonstrates a test generation method that considers variability and generates tests for different variants without redundancies. Although redundant test generations could be avoided, generated tests for different variants still contain identical tests, which verify the commonality of these variants. At the end of chapter 8, a method of avoiding redundant executions of such tests will be introduced.

Part IV includes the evaluation and summary. Chapter 9 evaluates the modeling and testing concept for variability from the efficiency perspective. First, a mathematical plausibility analysis is demonstrated in order to show that the proposed modeling and testing concept is profitable based on a practical HMI project. Second, a general discussion is led. Factors are identified on which the efficiency improvement of the proposed approaches is dependent. Also, worst cases are identified in which it is not profitable to use these approaches. Finally, an demonstration of efficiency improvement is given using concrete data. Chapter 10 gives a summary and an overview of future work using the perspectives of the research and the industry.