Automated Consequence Analysis for Automotive Standards (AUTO-CAAS)

1. Project Plan

1.1. Problem definition

The Automotive Open System Architecture (AUTOSAR) standard is gaining momentum with several automotive manufacturers (such as Volvo) and there is a growing trend towards new vehicle platforms based on the latest versions of this standard. The standard enables manufacturers to allow Tier-1 suppliers to contract arbitrary Tier-2 software developer for ECUs, as long as the developed software conforms to the specified behavior according to AUTOSAR. This is in clear contrast to earlier situation, in which a preferred Tier-2 developer was appointed to develop software for all Tier-1 hardware suppliers. This paradigm shift brings about economical and financial benefits (both for suppliers and manufacturers). However, it also introduces certain risks and challenges.

The AUTOSAR standard is complex and does leave room for interpretation and optimizations. In order to be competitive, Tier-2 developers strive after implementing several optimizations and utilizing room for interpretation of the standard to make their product out-perform the competition.

The goal of this project is to exploit the technology of model-based testing in order to detect deviations from the AUTOSAR standard and furthermore trace the consequences of such deviations into visible deviating behavior (failures). To this end, we will use and enhance the model-based testing framework developed at QuviQ to detect deviations from the AUTOSAR standard. This framework is, for example, used by SP (Technical Research Institute of Sweden) to certify software delivered to Volvo Car Corporation. As noted before, one of the major obstacles in using the current model-based framework is the different interpretations of the standard. Unless the consequences of these interpretations are properly analyzed, such variations are justified by the developers. This poses a major challenge for the widespread application of the standard as a model for certifying components, modules, ECUs and vehicle functions.

We believe that we do have a strong position to make these consequences visible using the theory and the technology at hand: we have the different behaviours encoded in the different variants of models and hence, we can analyze these differences in order to demonstrate possible failures due to the mismatches between different combinations of variants. To this end, we will demonstrate the possible failure traces on concrete implementations, thanks to the open source implementation of AUTOSAR and its development environment provided by ArcCore.

1.2. Background Theories and methods

1.2.1. AUTOSAR

Standardizing the conformance process has been mentioned as a key goal of the AUTOSAR standard [Fennel+2006]. The recent releases of the AUTOSAR standard have partially achieved this goal by providing a standard for interfaces, behaviors and configurations for basic software [AUTOSAR 2011]. According to AUTOSAR, conformance and interoperability tests take place at various levels in the development of vehicle functions: starting from individual components, going on with (micro-)integration to modules, integration into ECUs and single vehicle functions after integration into the network. Figure 1 from [Gilberg+2010] provides a general overview of this process. A final step of testing is performed in the operational environment involving multiple vehicle functions. The scope of this project is from conformance testing of module micro-integration to testing single vehicle functions (the yellow- and orange-colored areas in Figure 1). In particular, we exploit the results of model-based conformance testing to predict and diagnose interoperability failures at the vehicle function level. Conformance test results are very helpful in identifying signature faults as well as behavioral faults. However, there are "gray areas" [Gilberg+2010] in the results of conformance testing, which are non-conclusive due the abstractions made in the AUTOSAR behavioral models and the ambiguities in specifications. Both issues leave some room for various, at times conflicting, design decisions by the suppliers and OEMs. Such conflicting design decisions can give rise to later failures when composing modules into ECUs and ECUs into vehicle functions.

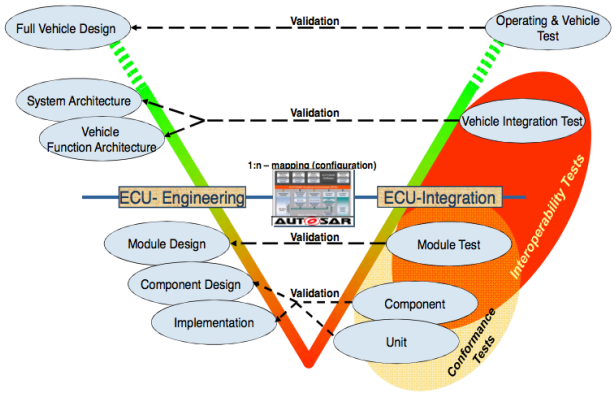

Figure 1. Conformance Testing in AUTOSAR [Gilberg+2010]

1.2.2. Model-based testing

Model-based testing (MBT) [Broy+2005] is a rigorous and structured technique to test computer systems. A schematic view of a typical MBT ecosystem is given in Figure 2 (the figure refers to the ecosystem used in model-based testing of an embedded system in the financial domain involving the project manager of this proposal [Asaadi et al 2001]). The process starts with making test models from the requirements and standards. Then a conformance test engine is in charge of generating test cases from the test models and executing these test cases in order to interact with the implementation under and to establish whether it conforms to the specification.

Examples of MBT test engines include Microsoft SpecExplorer [Veanes+2008], UPPAAL TRON [Hessel+2008], RT-Tester [Peleska and Huang 2013] and QuickCheck [Arts+ 2006, Hughes 2007], we refer to [Asaadi+2012, Vishal+2012] for our prior experience with industrial application of some of these tools). In the context of this project, we will used the property-based testing approach as implemented in the QuickCheck tool and in particular, use the existing rigorous specification of the AUTOSAR standard in QuickCheck. Our industrial partners provide examples of experimental implementation of components and module that demonstrate non-conforming behavior. When composing such components and modules, a summary of detected (non-conforming) behavior will be used to predict possible failures. Moreover, in case of actual failures, the behavioral summary model built through interaction with the implementations will be used to designate the root cause of failure.

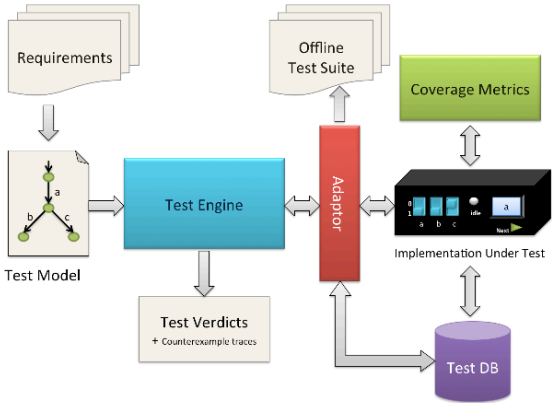

Figure 2. Schematic View of Model-Based Testing

1.2.3. Fault diagnosis and automated debugging

Fault diagnosis and model-based fault diagnosis havea long tradition in dynamical systems and supervisory control [Isermann 1997, Chen and Patton 1999, Isermann 2005]. Fault diagnosis ideas have subsequently been exploited in computer systems, e.g., in the form of spectrum-based diagnosis of software systems [Harrold 2000]. In spectrum-based diagnosis, passed and failed executions are scrutinized, and annotated with information about the execution of each line (block or module) of program code. Note that for diagnosis one does not differentiate whether in a particular run a block caused the failure or not; it is just checked whether the block is part of the whole execution.

An alternative approach to spectrum-based fault diagnosis, which uses more semantic information, is delta debugging [Zeller and Hildebrandt 2002], which uses a set of passing and failing conditions in order to efficiently uncover a small failing execution.

To exploit these approaches in our context, we exploit the extra information obtained in the process of conformance testing to narrow down the search process when applying fault diagnosis and debugging techniques. We envisage that exploitation of these extra pieces information, which are made available through conformance testing will result in faster and more accurate diagnosis. We have tried both spectrum-based diagnosis and delta debugging in our past research [Woehrle+2013] and hence, do have in-house knowledge about both.

1.2.4. Symbolic execution and concolic testing

Symbolic execution has been successfully applied to test and verify computer (particularly software) systems in the past ten years [Williams+ 2005,Cadar+ 2011]. To apply symbolic execution in software testing, one usually starts by

running the system under test (symbolically or concretely with random seed values) and following the execution trace until reaching decision points. Conditions at decision points are accumulated along the execution and by using constraint solvers (such as powerful satisfiability-modulo-theory-solvers), the obtained conditions are turned into concrete valuations for parameters. Hence, new concrete test cases are obtained, leading to maximum coverage of the code. This technique is often called "concolic (a combination of concrete and symbolic techniques in) testing". In our context, concolic testing can be particularly useful in producing summaries of models and implementation during the conformance testing process, similar to the approaches reported in [Godefroid 2007, Siddiqui+ 2013].

1.3. Expected results

We develop an automated diagnosis approach that exploits the information from the conformance testing process in order to predict whether the integration of concrete realizations will lead to any failure. Additionally, we use the information gathered during the conformance testing process in order to diagnose the later observed integration and vehicle-function level failures and find the root cause of failure.
This will resolve a contemporary problem in the application in the practice of automotive software and systems. Particularly, use of the popular AUTOSAR standard in our research forms the basis of our model-based approach and enables its wide-spread application in industrial practice.

1.4. Consortium

The consortiums features a unique composition of both a knowledge-provider (CERES), a tool and technique developer (QuviQ) and application platform developer (ArcCore). Below we briefly describe each of the involved partners and their role and contributions in the project:

- **Center for Research on Embedded Systems at Halmstad University.** Centre for Research on Embedded Systems (CERES) at Halmstad University focuses on research in Cooperating Embedded Systems. The research at CERES aims at exploiting the opportunities for cooperation that new enabling technologies provide for the benefit of industry. CERES has an international and vibrant research environment with about 40 researchers (including 6 full professors). Hence, CERES is home to wide range of expertise in the area of embedded and cyber-physical systems. In particular, CERES is represented in this project by the research group on Model-Based Testing and Verification, led by prof. Mohammadreza Mousavi, who will be the project manager. The research agenda for this group is focused on providing practical industrial strength MBT solutions that can deal with product lines of embedded systems.
  The role of CERES in this project is to provide the necessary knowledge and research skills in order to co-develop a novel, yet practical, approach for the research problem and co-develop it within the toolsets provided by QuviQ and ArcCore.

- **QuviQ AB.** Quviq is a spin-off from research performed at Chalmers. The company has been founded in 2006, by John Hughes and Thomas Arts. The company has commercialized the property based testing techniques by selling the QuickCheck tool as well as services around this tool. Large customers are among others, Ericsson, Motorola, GeminiMobile, Volvo and SP. Quviq has also a number of customers that develop applications using Erlang, such as Basho and Klarna.
  The role of QuviQ in this project is to provide both the tools and the models for the conformance testing according to the AUTOSAR standard. The rich research-oriented experience in QuviQ enhances the possibility of co-development with CERES. In addition, QuviQ has a history of successful partnership with ArCore.

- **ArcCore AB.** ArcCore is a leading provider of state-of-art products and services for the embedded systems market. Based on a solid expert knowledge in real-time platforms, it develops and markets products for the software industry enabling its customers to develop innovative solutions in a faster and more cost-efficient way.
  The role of ArcCore in this project is to provide the implementation and the implementation platforms for the AUTOSAR standard, respectively, based on their Arctic Core and Arctic Studio products, which provide an support for various levels of AUTOSAR development and integration.

1.5. Project structure

The project is structured into the following 6 work packages, which are in turn decomposed into several tasks with specified deliverables. Figure 3 provides an overview of the project structure.

WP1: Familiarizing with the literature and tools. (Length: 8 months, duration: months 1-8) The goal of this work package is to familiarize the involved researchers (particularly the postdoctoral researcher) with the to-be-exploited techniques and tools.

Task 1.1: Model-Based Testing Techniques (length: 2 months, duration: months 1-2). The goal of this task is to familiarize the junior researcher with the underlying theories of MBT to be used throughout the project. From the past and ongoing projects and courses of the project manager on the subject ample study material has been developed, which can be used for this task. **(Involved people: project manager, postdoctoral researcher. Deliverable: a technical report structuring the basic knowledge on the subject, authored by the junior researcher and edited by the project manager.)**

Task 1.2: Standard (AUTOSAR) (length: 2 months, duration: months 3-4). The goal of this task is to familiarize the junior researcher with the

Task 1.3. Tool (QuickCheck Tool and Models) (length: 2 months, duration: months 5-6). Quviq has already provided tutorial material on its tool and will provide licenses to both the QuickCheck tool and AUTOSAR models. We will interact with Quviq experts in learning the tools and understanding their models. **(Involved people: project manager, postdoctoral researcher and QuviQ experts. Deliverable: a tutorial on the tool and the models co-authored by the postdoctoral researcher and the project manager and edited by Quviq.)**

Task 1.4. Implementation and Development Environment (Arctic Core and Arctic Studio)) (length: 2 months, duration: months 7-8). ArcCore will provide tutorial and support for its implementation and development environment. We will interact with Quviq experts in learning the tools and their implementations. **(Involved people: project manager, junior researcher and ArcCore experts. Deliverable: a tutorial on the tool and a minimal set of components co-authored by the postdoctoral researcher and the project manager and edited by ArcCore.)**

WP2: From conformance testing to summaries. (Length: 9 months, duration: months 12-15, 17-18, 21-23) The goal of this task is to create model and implementation summaries that compactly describe the variation points between the AUTOSAR model and the implemented component / module. These symbolic models also include possible parameterizations of the implementation.

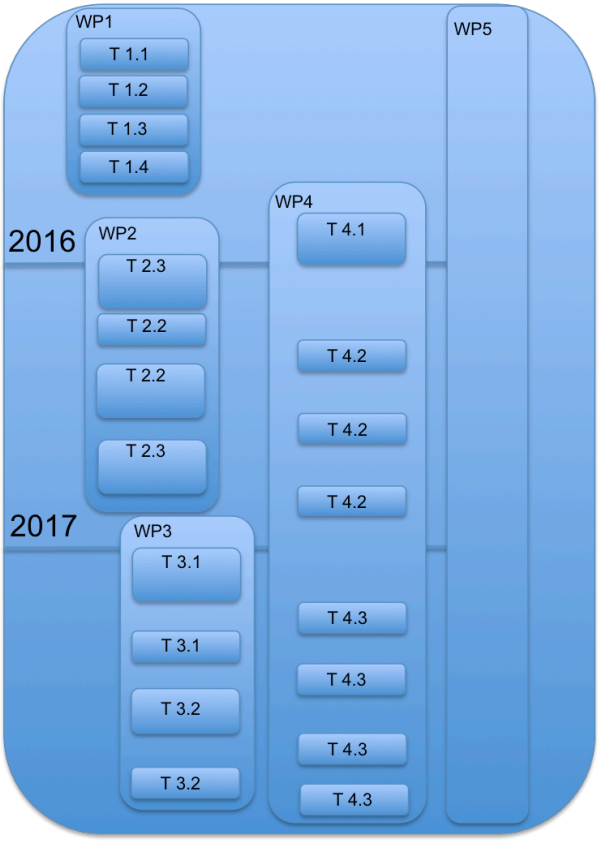

Figure 3. Schematic View of Project Structure

Task 2.1 Defining the domain of symbolic summaries for AUTOSAR components / modules. (Length: 3 months, duration: months 12-15) The goal of this task is to define a formal framework that is expressive enough to act as the common semantic domain both for AUTOSAR models and implementations. Moreover, we shall specify the semantic properties and define composition and reduction techniques for such models and implementations and prove them. **(Involved people: project manager, junior researcher and QuviQ experts. Deliverable: a theoretical paper on the semantic domain of AUTOSAR components, its properties and reductions co-authored by the postdoctoral researcher and the project manager.)**

Task 2.2. Integrating the extraction of summaries in the conformance testing process. (Length: 3 months, duration: months 16,18-19) The goal of this task is to summarize the deviating behavior in the conformance testing process in the semantic domain defined in Task 2.1 during the conformance testing in order. Note that this task is interleaved with Task 4.2 in order to re-visit the approach by trying it on our set of examples gathered in Task 4.1. In this task symbolic execution and concolic testing techniques and tools can be exploited in order to extract the summaries efficiently. **(Involved people: project manager, junior researcher and QuviQ experts. Deliverable (combined with Task 4.2): a technical report on the extraction of summaries during conformance testing co-authored by the postdoctoral researcher and the project manager and reviewed by the QuviQ experts.)**

Task 2.3. Implementation of the integrated approach in the conformance testing tool. (Length: 3 months, duration: months 20-23) The goal of this step is to integrate the developed approach in the QuviQ toolset. This will be performed in collaboration with the QuviQ expert. This task is again interleaved with application to the case studies developed in Task 4.1. **(Involved people: project manager, junior researcher and QuviQ experts. Deliverable (combined with Task 4.2): software and documentation of the developed tool co-authored by the postdoctoral researcher and the project manager and edited by the QuviQ experts.)**

WP3: Exploiting Conformance Data in Diagnosis (Length: 8 months, duration: months 25-27, 30-31, 33-35) The goal of this task is to exploit the information gathered in the conformance testing process in WP2 in order to predict possible failures and find the root causes of failure.

Task 3.1 Aligning failures with summaries. (Length: 4 months, duration: months 25-26, 28-29) The goal of this task is to use the summary of mismatching behaviors gathered in WP2 in order to both predict possible failures, as well as exploit an existing failure trace in order to find the root cause for failure. Both goals boil down to a guided (respectively, forward and backward) search in the composition of summaries in order to reach a failure (a particular failure in the second case). In this task, we will exploit and integrated existing bodies of knowledge in the (symbolic) model checking literature and the automated debugging and fault diagnosis approaches. **(Involved people: project manager, junior researcher and ArcCore and QuviQ experts. Deliverable: a conference paper on the two diagnosis approaches and their illustration on a subset of the case studies, co-authored by the postdoctoral researcher and the project manager and edited by the ArcCore and QuviQ experts.)**

Task 3.2 Implementing the diagnosis approach and integrating it in the toolset. (Length: 4 months, duration: months 31-32, 34-35) The goal of this task is to implement the developed techniques in the tool chains of ArcCore and QuviQ. We will use our case studies as benchmarks for our imeplementations **(Involved people: project manager, junior researcher and ArcCore and QuviQ experts. Deliverable (combined with Task 4.3): software and**

documentation of the developed tool co-authored by the postdoctoral researcher and the project manager and edited by the ArcCore and QuviQ experts. A journal paper summarizing the project, its techniques and the results of its applications on the case studies.)

WP4: Case Studies (Length: 11 months, duration: months 9-11, 16, 19-20, 24, 27, 30, 33, 36) The goal of this task is to initially come up with a benchmark for models and implementations to be used throughout the project. Subsequently, we apply each of the developed techniques this benchmark in order to evaluate the effectiveness of our developed approaches and improve them iteratively.

Task 4.1. Composing a benchmark set of canonical models and implementations. (Length: 3 months, duration: months 9-11) The goal of this task is to initially come up with a set of canonical examples both for the AUTOSAR specifications and the corresponding implementations; for the former we use the models provided by QuviQ and for the latter we use the implementations provided by ArcCore. We also use techniques such as mutation in order to produce faulty implementations. We will also provide a simplified set of examples that can be published without exposing the confidential details of QuviQ models. **(Involved people: project manager, junior researcher and ArcCore and QuviQ experts. Deliverable: a tool paper describing the developed benchmark co-authored by all participants.)**

Task 4.2. Evaluating the summary extraction approach. (Length: 3 months, duration: months 16,19-20,24) The goal of this task is to apply the summary extraction technique (also in the course of conformance testing) to the developed benchmark in Task 4.1 and use the obtained results to improve and evaluate the approach. **(Involved people: project manager, junior researcher and QuviQ experts. Deliverable: see Tasks 2.2 and 2.3.)**

Task 4.3. Evaluating the diagnosis approach (Length: 3 months, duration: months) The goal of this task is to apply the diagnosis techniques in Tasks 3.1.and 3.2 to the developed benchmark in Task 4.1. **(Involved people: project manager, junior researcher and ArcCore and QuviQ experts. Deliverable: see Tasks 3.1 and 3.2.)**

WP5: Dissemination (throughout the project). The industrial partners have an open publication policy and agreed to publish the results (after screening) at academic and industrial venues. Typical academic venues include ICST, ISSRE, ISSTA, ASE, ICSE, ETAPS (FASE,ESOP), and ACM SAC (SVT) conferences as well as MBT and A-MOST workshops. Typical industrial venues include the Scandinavian Embedded Conference, Test Automation Day and Vehicle ICT Arena events.
After disseminate the initial results in such venues, we plan to publish a complete account of the research in reputable archival journals such as SoSym, IEEE TSE, ACM TOSEM, and Software Testing, Verification and Reliability Journals.

References

[AUTOSAR 2011] AUTOSAR BSW & RE Conformance Test Specification, Release 4.0, Revision 2, 2011.

[Arts+ 2006] T. Arts, J. Hughes, J. Johansson, and U. Wiger. Testing telecoms software with quviq QuickCheck. In Proc. of ERLANG'06, ACM, 2006.

[Asaadi+ 2012] H.R. Asaadi, R. Khosravi, M.R. Mousavi, and N. Noroozi. Towards Model-Based Testing of Electronic Funds Transfer Systems. In Proc. of FSEN'11, vol. 7141 of LNCS, Springer, 2012.

[Broy+ 2005] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner. (Eds.) Model-Based Testing of Reactive Systems. volume 3472 of LNCS, Springer, 2005.

[Cadar+2011] Cristian Cadar, Patrice Godefroid, Sarfraz Khurshid, Corina S. Pasareanu, Koushik Sen, Nikolai Tillmann, Willem Visser: Symbolic execution for software testing in practice: preliminary assessment. In Proc. of ICSE 2011, 2011.

[Chen and Patton 1999] R.J. Chen and R.J. Patton. Robust Model-Based Fault Diagnosis for Dynamical Systems. Kluwer, 1999.

[Fennel+ 2006] H. Fennel et al., Achievements and Exploitation of the AUTOSAR Development Partnership, SAE Convergence Congress, Detroit, 2006.

[Gilberg+2010] A.A. Gilberg, B.B. Kunkel, C.A. Ribault, D.P. Robin, and E.N. Spinner. Conformance Testing for the AUTOSAR Standard, Embedded Real Time Software and Systems Conference, 2010.

[Godefroid 2007] Patrice Godefroid: Compositional dynamic test generation. In Proc. of POPL 2007, IEEE, 2007.

[Harrold+2000] M.J. Harrold, G. Rothermel, K. Sayre, R. Wu, and L. Yi. (2000). An empirical investigation of the relationship between spectra differences and regression faults. Software Testing Verification and Reliability, 10(3), 171-194.

[Hessel+ 2008] A. Hessel, K.G. Larsen, M. Mikucionis, B. Nielsen, P. Pettersson, and A. Skou. Testing real-time systems using UPPAAL. In Proc. of FMT'08, vol. 4949 of LNCS, Springer, 2008.

[Hughes 2007] John Hughes. QuickCheck testing for fun and profit. In Proc. of PADL'07, Springer, 2007.

[Isermann 1997] R. Isermann. Supervision, fault-detection and fault-diagnosis methods—an introduction. Control engineering practice, 5(5), 639-652, 1997.

[Isermann 2005] R. Isermann. Model-based fault-detection and diagnosis–status and applications. Annual Reviews in control, 29(1), 71-85, 2005.

[Peleska and Huang 2013]	J. Peleska and W.-l. Huang, Model-Based Testing With RT-Tester. Lecture slides of the HSST'13, Halmstad University, 2013.
	J.H. Siddiqui, S. Khurshid. Scaling symbolic execution using staged analysis. ISSE 9(2): 119-131, 2013.
[Veanes+ 2008]	M. Veanes et al. Model-based testing of object-oriented reactive systems with Spec Explorer. In Proc. of FMT'08, vol. 4949 of LNCS, Springer, 2008.
[Vishal+ 2012]	V. Vishal, M. Kovacioglu, R. Kherazi, and M.R. Mousavi. Integrating Model-Based and Constraint-Based Testing Using SpecExplorer. In Proc. of MoTiP'12, IEEE CS, 2012.
[Williams+ 2005]	N. Williams, B. Marre, P. Mouy, and R. Muriel. PathCrawler: Automatic Generation of Path Tests by Combining Static and Dynamic Analysis. In Proc. of the EDCC'05, pp. 281–292. Springer, 2005.
[Woehrle+ 2012]	M. Woehrle, R. Bakhshi, and M.R. Mousavi. Mechanized Extraction of Topology Anti-patterns in Wireless Networks. In Proc. of iFM 2012, vol. 7321 of LNCS, Springer, 2012.
[Zeller and Hildebrandt 2002]	A. Zeller A and R. Hildebrandt. Simplifying and isolating failure-inducing input. IEEE Transactions Software Eng. 2002; 28:183–200, 2002.