

An Effective Verification Strategy for Testing Distributed Automotive Embedded Software Functions: A Case Study

Annapurna Chunduri¹(✉), Robert Feldt¹, and Mikael Adenmark²

¹ Department of Software Engineering, Blekinge Tekniska Högskola,
SE-371 79 Karlskrona, Sweden

anuchunduri11@gmail.com, robert.feldt@bth.se

² Department of Systems and Integration Test, Scania AB,
SE-151 87 Södertälje, Sweden
mikael.adenmark@scania.com

Abstract. Integration testing of automotive embedded software functions that are distributed across several Electronic Control Unit (ECU) system software modules is a complex and challenging task in today's automotive industry. They neither have infinite resources, nor have the time to carry out exhaustive testing of these functions. On the other hand, the traditional approach of implementing an ad-hoc selection of test scenarios based on the testers' experience typically leads to both test gaps and test redundancies. Here, we address this challenge by proposing a verification strategy that enhances the process in order to identify and mitigate such gaps and redundancies in automotive system software testing. This helps increase test coverage by taking more data-driven decisions for integration testing of the functions. The strategy was developed in a case study at a Swedish automotive company that involved multiple data collection steps. After static validation of the proposed strategy it was evaluated on one distributed automotive software function, the Fuel Level Display, and found to be both feasible and effective.

Keywords: Verification strategy · Distributed automotive embedded software · Test coverage · Test process improvement · Case study

1 Introduction

The automotive industry has been significantly affected by the industrial software revolution over the past decade. The share and importance of software within a vehicle is growing steadily. Early on it was anticipated that 90% of all future automotive innovations will be driven by software [10] and since then it has become clear that the industry is increasingly software-centric.

An automotive vehicle consists of Electronic Control Unit (ECU) systems, which are essentially embedded microcontrollers with corresponding software components. These ECU systems interact in order to execute the desired functionality in the vehicle like controlling the engine and operating air bags.

In the past, each single ECU system had a single dedicated function. Hence, execution of a function required the software within only one of the ECU systems to execute independently. Nowadays, the functions are being designed to be realized through the interaction among different sub-functions and ECU systems via multiple Control Area Networks (CANs) [11]. Such cross-functionality i.e., a function distributed across multiple ECU systems and consequently across several system software modules is a common and complex phenomenon in today’s automobile vehicles.

In the automotive industry, the standard V-model is most widely used for the engineering processes of embedded system software development. A typical V-model implemented by Scania [1], a major manufacturer of commercial heavy vehicles in the European automotive industry, can be seen in Fig. 1. Similarly, across the automotive industry, testing of the embedded system software functions occurs at different test levels. Here, the term ‘test level’ is used to indicate the test focus. *“Each test level describes an area of test responsibility”* [1]. For instance, at the system test level, for each individual ECU system of the vehicle, the software modules are mounted on the corresponding ECU system hardware like the engine and the gearbox and tested independently. At the vehicle integration test level, the software modules corresponding to all the ECU systems which make up the vehicle are tested together in their actual operational vehicle environment. The testing across the test levels is performed either as Hardware in the Loop (HIL) testing, where individual ECU systems or the entire vehicle is executed under simulated environments [12], or using real ECU systems or vehicles. The exact number of test levels and terminology used to describe each test level may differ from one automotive company to another. But what remains the fundamental similarity in the concept of testing is that, at each test level, the test strategy adopted aims to address and test the system software behavior at a different level of abstraction and provides a different degree of coverage of the object under test [21].

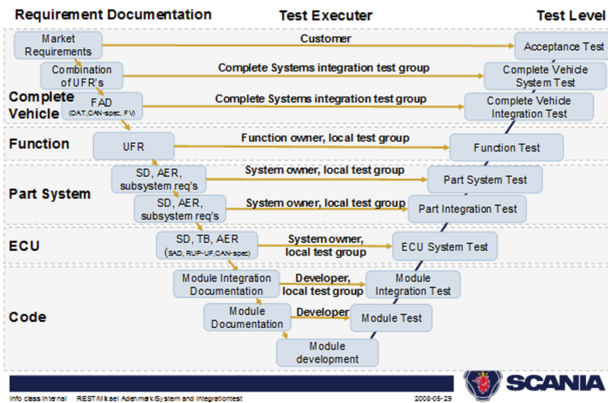


Fig. 1. The V-model implemented at Scania [1] to test their distributed embedded software functions

The distributed nature of functions across several embedded system software modules makes integration testing of the automotive system software a complex and challenging task [17]. Exhaustive integration testing of the distributed functions using the numerous variants across the software modules is not feasible since the automotive industry neither has infinite resources nor has the time to carry out such exhaustive testing [3]. On the other hand, going by the traditional approach of implementing an ad-hoc selection of test scenarios based on the testers experience typically leads to both test gaps and test redundancies across the test levels [8]. Hence, there is a pressing need for a feasible and effective verification strategy for integration testing of the distributed automotive embedded software functions that would help improve test coverage while reducing test redundancies and test gaps across the test levels. A review of research in test, verification, and validation in the automotive domain has revealed that research focusing on studying such challenges of high-level integration testing of the distributed automotive embedded software functions like in [11, 27], is limited. More over, research proposing and validating strategies and solutions to cope with such critical integration test challenges, like in [3], has been found to be sparse.

2 Research Design

The research was conducted in association with Scania, a major manufacturer of commercial heavy vehicles in the European automotive industry. This section provides an overview of the research design. It is based on the guidelines for conducting and reporting case study research in [20].

2.1 Research Questions

The goal of the research was to initially identify the challenges in automotive system software testing of the distributed functions. Thereafter, the research aimed to propose and evaluate a verification strategy which was based on addressing a suitable subset of the identified challenges. Here, the evaluation of the proposed strategy was in terms of its effectiveness and feasibility. Initially, the strategy was evaluated based on static comparative analysis using scientific literature knowledge and industry expert opinion. Thereafter, the strategy was evaluated based on its implementation on a legacy distributed automotive software function- Fuel Level Display. Hence, in addition to having a realistic real world setting where the proposed strategy was studied, the analysis has been extended towards investigating the perceived effectiveness and feasibility of the proposed strategy based on static data collected from industry experts and scientific literature.

Based on the above mentioned aims of the study, the research questions (RQ) formulated are as follows.

RQ1: What are the major challenges with the current approach to test distributed software functions across test levels at the automotive case company?

RQ2: What is the feasibility and effectiveness of implementation of the proposed

process enhancement based verification strategy according to (a) scientific literature knowledge, (b) industry expert opinion and (c) Historic data in a real world setting?

2.2 Case Selection

Scania was chosen to be a suitable case for conducting the research and answering the research questions. Scania is present in more than 100 countries and has been successfully running for 250 years with over 45,000 employees. It represents a real world setting of an automotive industry that is facing software-related challenges in several fronts including the area of integration testing which is the focus of the current research. Hence, this single-case study has been deemed appropriate due to its *representative* nature [31] of the automotive industry companies that are effected by the advent of software in the field of integration testing.

2.3 Data Collection Procedure

One of the data collection methods was through conducting interviews of different practitioners involved in the software testing activities at the case company. A non-probabilistic judgement based sampling technique [14] was employed for selection of interview subjects. Practitioners who were involved in, and had adequate knowledge and experience in testing were chosen. The identified practitioners included a total of 13 test engineers from different departments with varied experience belonging to 3 categories based on test levels, which are, System Test Engineers, Function Test Engineers, and Lab and Vehicle Integration Test Engineers. The interviews conducted were face-to-face, semi-structured interviews with open-ended and close-ended questions. Apart from interviews, data from relevant artifacts in databases and archived documents in the form of test reports, test results, test cases, function requirements specification and function architecture design documents among others have been used as alternative sources of data to achieve data triangulation [20].

2.4 Data Analysis Procedure

The data collection methods facilitated collection of both qualitative (predominant) and quantitative data through interviews, archived documents and databases. Strauss and Corbin's Grounded Theory (GT) approach [26], a commonly used method that facilitates the analysis of qualitative and quantitative data was employed to answer the research questions. The three steps of coding process - open coding, axial coding and selective coding- were performed. The theory so generated was confirmed through the hypothesis confirmation method of triangulation. In addition to data triangulation, user-group triangulation [30] based on selection of interview subjects from multiple categories and multiple subjects from each category, and evaluator triangulation [30] based on reviewing the results obtained with several researchers working within similar areas of research was employed. Hence, the final output of data analysis presents results that are both grounded and supported by a body of evidence.

2.5 Validity Procedure

The proposed strategy addresses process-based challenges. The validity of the strategy was evaluated based on its *effectiveness* of obtaining the desired results and its *feasibility* for implementation with optimum cost and effort. To assess these aspects of the strategy, a static comparative analysis of the process-enhancement approach proposed in this strategy with alternative process enhancement approaches was performed. This analysis was based on scientific literature knowledge and industry expert opinion. Here, scientific literature knowledge was captured by conducting a literature review as per the guidelines presented in [19]. To capture industry expert opinions, six practitioners from Scania with vast experience and high knowledge in the relevant areas were carefully chosen. The evaluation study was conducted based on an assessment of the *cost*, in terms of time and people effort in implementing the strategy, *certainty of Return on Investment (ROI)* in terms of the likelihood that the implementation of the strategy would bring the desired results, and *value of the ROI* in terms of the extent to which the results obtained on implementing the strategy solve the challenges at hand. There after, historic data pertaining to testing a distributed automotive software function - Fuel Level Display across the different test levels for three test rounds spread over three months was collected. This historic data was then used to implement the proposed strategy and analyse the results to assess its feasibility and effectiveness in a real world setting.

3 Challenges in Automotive System Software Testing

The challenges, in the form of issues, pertaining to the approach implemented to test distributed automotive software functions across the test levels at the case company have been summarized and presented in Table 1 to answer research question RQ1. It was identified that the challenges found through the case study were similar to the challenges presented in existing scientific literature [11, 27]. The study of the interdependent relationships that exist among the identified issues helped establish that there is a very intricate relationship that can be represented through a multi-way dependency among the identified set of people, process and technology issues. For instance, let us consider the complex relationship between one identified set of people, process and technology issues presented in Fig. 2. Hence, it was inferred from the results obtained on performing several steps of data analysis that, an intricate set of people, process and technology issues formed the source for the fundamental issues for integration testing of the distributed software functions at the case company.

A further study of the areas of people, process and technology and the relationships that exist between the issues within these areas in the current context was performed. It was then identified based on literature [18] and on case study results (refer to Fig. 2), that process issues lie at the core of the three identified issue areas. Technology is adopted to fit the process established, and people knowledge and effort is built around the process. Hence, addressing the process

Table 1. Identified challenges in the form of issues with current approach to test distributed embedded software functions at Scania

SNo	Issue category	Description of issues
1	People Issues	Ambiguity of test role at function test level and a lack of bigger picture of function test effort distribution across test levels
		Lack of appropriate means to share knowledge pertaining to test results across the test levels
2	Process Issues	Incomplete, vague and untestable higher level function requirements specification
		Variant-focused testing is highly dependent on each individual tester’s knowledge with lack of means to handle function variants through simplistic representation
		Lack of adequate traceability across requirements and test results of the different test levels
3	Technology Issues	Lack of tool support to assist in test management, requirements management and change management
		Complexity of having test artifacts (test results, test reports, test cases, etc.) and function and system requirements spread across multiple locations with limited interoperability
4	Fundamental Test Issues	No comprehensive function test coverage information across test levels
		Ambiguous and inaccurate test reports generated at vehicle integration test level
		Lack of test report generation at the function test level
		Lack of appropriate means to identify where or whether redundant testing and risky test gaps are present across the test levels

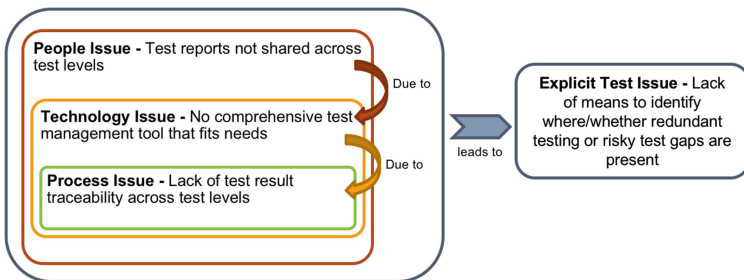


Fig. 2. An instance of the identified people, process and technology issue relationship

issues was deduced to be suitable to help pave the way for studying and addressing the people and technology issues that persist around the established process. Thus, the verification strategy proposed in this research aims to address the process issues in order to resolve the fundamental test issues identified in the approach for testing distributed embedded software functions at the automotive case company.

4 Process Enhancement Based Verification Strategy

The verification strategy proposes to enhance the function requirements specification by implementing a semi-formal scenario-based modelling that provides a means to capture complete and testable requirements with adequate function variant information. Thereafter, the strategy proposes to adopt a multi-level reuse concept of combining test results from the system, function and vehicle integration test levels by establishing appropriate traceability across the requirements and among the test results. Here, test levels with a focus on requirements-based test coverage have been considered to be within the scope. The obtained comprehensive test coverage information can then be used to identify test gaps and test redundancies and take more data-driven decisions to enhance integration testing and thereby improve test coverage of the distributed software functions.

4.1 Steps for Implementation of the Verification Strategy

The steps for implementation of the proposed verification strategy are presented in Fig. 3. A brief description of these steps is discussed below.

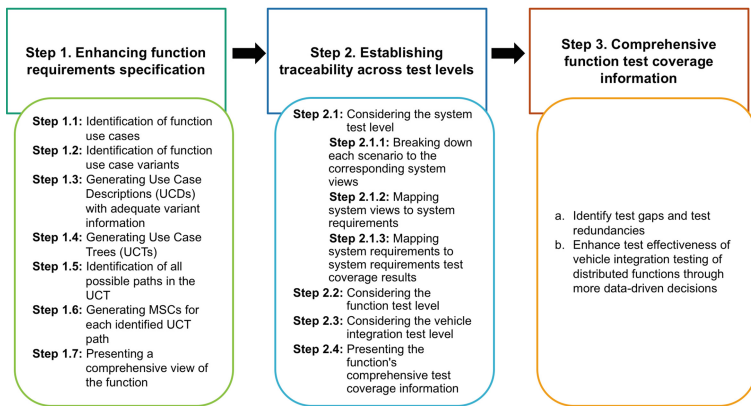


Fig. 3. Steps for implementation of proposed process enhancement based verification strategy

STEP 1. Enhancing Function Requirements Specification

The first step deals with adopting a semi-formal scenario-based modelling of the distributed software function requirements. It initially involves the identification of function use cases and use case variants. Here, each use case of a function describes one of its unique behaviour and use case variants are all possible contexts in which the function use case takes a different sequence of actions to execute the same behaviour described by the use case. For example, let us consider the function Fuel Level Display, for which two use cases are activation and deactivation of the feature of displaying low fuel level warning on the Instrument Cluster (display component) of the vehicle, when the activation and deactivation conditions are met. Here, for one of the use cases- activation of low fuel level warning- its variants can be identified to be all those contexts in which different steps are executed to reach the same outcome of displaying low fuel level warning when activation conditions are met. This here would mean that if activation of low fuel level warning is executed differently in vehicles with liquid engine and in vehicles with gas engine, then the use case variants include vehicles with liquid engine and vehicles with gas engine.

The next step is to generate a Use Case Description (UCD) for each use case of the function and a Use Case Tree (UCT) for each function. A UCD is a step-by-step description of the interaction between the participating systems required to execute the use case in its operational environment, written in structured natural language. There are several different templates proposed in literature for generating UCDs. One such template is presented by Somé [25] which is the inspiration for the current template proposed. The template is tailored and enhanced to be fit the automotive industry. A UCT is a graphical tree-structure representation of all possible paths containing unique combination of steps from the UCD for the execution of the use case.

Thereafter, all possible UCT paths, termed ‘scenarios’, are to be identified and corresponding Message Sequence Charts (MSCs) [25] are to be generated such that

- (a) Nodes common to all variants are covered at least once for each variant, and
- (b) Nodes specific to a particular variant are covered at least once for that variant.

Here, a scenario is one single sequence of interaction between the involved systems to execute a use case of the function. Adequate notes should be used within the MSCs to capture the important information of the UCD within the scenario models to enhance their understandability. This representation can hence be used to provide a comprehensive view of the function as a set of complete and testable function requirement scenarios with adequate variant information.

STEP 2. Establishing Traceability across Test Levels

This step deals with establishing traceability to obtain comprehensive test coverage information for the functions across the System, Function and Vehicle Integration test levels.

Initially, the system test level is taken into consideration. Here, there is a need to establish traces between the function scenarios involving multiple ECU

systems and system level requirements of each involved ECU system. There after, the system level requirements test coverage pertaining to the mapped requirements are to be captured against the function scenarios. This provides an understanding of how much of the function is tested at the system test level across all the relevant systems. To establish traces between the function scenario and system level requirements, each function scenario is taken and broken down to the system view. Thus, each scenario has a corresponding set of system views of all the systems that contribute to its realisation. Here, each system view represents a set of all relevant input and output combinations pertaining to that system's role in the scenario.

The next sub-step is to consider the function test level. The test focus at this level is on testing the overall function requirements. Therefore, the test results should be mapped to the corresponding scenarios. Here, the scenarios and their expected outcomes represent the function requirements that are to be tested.

The last sub-step is to consider the vehicle integration test level. At this test level the test focus is on interface communication across the systems involved in the execution of the function scenarios. Therefore, at this test level, similar to the function test level, the test results are to be mapped against the function scenarios. Finally, as a result of implementation of this step, the obtained comprehensive function test coverage information across test levels is to be presented.

STEP 3. Comprehensive Function Test Coverage Information

The comprehensive function test coverage information so obtained can then be used within the last step of the strategy to identify risky test gaps, avoidable test redundancies and make a more data-driven decision during integration testing of the distributed embedded software functions to enhance their test coverage.

5 Validation Results and Discussion

The results of the study to validate the *feasibility* and *effectiveness* of the proposed verification strategy to answer research question RQ2 are presented and discussed in this section.

5.1 Scientific Literature and Industry Expert Opinion

As presented in Sect. 4, the proposed strategy is a semi-formal process enhancement approach (Approach 2) to solving the fundamental issues in integration testing of the distributed software functions. Hence, a static comparative analysis to its alternatives - informal and formal process enhancement approaches (Approaches 1 and 3) was conducted to validate its relative feasibility and effectiveness. For the three alternative process enhancement approaches, the cost, the certainty of ROI and the value of the ROI was assessed using a relative ordinal scale of Highest(H), Medium(M) and Least(L). The results so obtained are presented in Table 2.

Table 2. Comprehensive results of comparative analysis of alternative process enhancement approaches

Approach	Estimation based on Scientific Literature			Estimation based on Industry Expert Opinion		
	Cost	Certainty of ROI	Value of ROI	Cost	Certainty of ROI	Value of ROI
(1)	L	M	L	L	H	L
(2)	M	H	M	M	M	H
(3)	H	L	H	H	L	M

It has been identified based on scientific literature knowledge that, informal approaches using natural language for the concerned process enhancement areas (refer to Table 1) present ease of adoption and lack of need of any special skills which implies low cost of implementation [9]. Yet, such natural language textual requirements and requirements based testing, irrespective of the level of abstraction of the system or software, have been found in ample of cases to still be prone to misinterpretation, ambiguity, inconsistencies and inaccuracy [5]. Moreover, it has been found through experience and reported in literature that in the automotive industry, textual natural language requirements and requirements based testing are “*only part of the game*” [29]. The industry identifies the need for the textual requirements at each level to be supported with adequate ways to capture, model and present various other complex attributes of the industry and thereby facilitate enhanced testing. A similar opinion has been found to prevail among the industry experts interviewed. It is evident that majority of them believe the ROI on such an investment will relatively add least value to the industry in terms of solving the test process issues at hand.

While formal methods have been found to be one of the best options for model based requirements specification, testing and aligning these two disciplines for safety critical systems like the systems within automotive industry [4], it has also been found to be difficult for practitioners to implement. It requires high experience in representing requirements in mathematical and logical notations [5, 22]. This makes it is a highly cost-intensive solution. Such is also the opinion of the experts interviewed, who believe that formal approaches might lead to relatively highest cost of implementation with least certainty of ROI.

On the other hand, focusing on semi-formal methods through use cases and scenarios as a means to model and test the functional behaviour of a software or a system has been found to have their own challenges like the challenge of generating executable test cases for enhancing efficiency of testing [2]. But exploring semi-formal Model Based Testing (MBT) methods and addressing the challenges within it has been found to generate solutions that are more cost effective and scalable as compared to the formal MBT methods. This makes it an interesting area of research with high focus from the industry for whom cost effectiveness and feasibility of solutions are critical [4]. A similar estimation of relatively high value of implementing this approach has been found to be the

opinion of industry experts. While it might cost the industry considerable time and effort for its implementation, majority believe it is a relatively most feasible approach that will likely produce relatively most effective results.

5.2 Historic Data

The strategy was implemented for Fuel Level Display function which is distributed across six systems, with two function variants and six function scenarios. The comprehensive function test coverage information across the test levels was captured for three test rounds (refer to Fig. 4). This helped identify the following:

- It was evident that most of the test effort across the test levels was focused on liquid engine trucks. This exposes the risky test gap in testing the function adequately across test levels for its implementation in gas engine variant.
- There is redundant testing of the main function scenarios (SCN), SCN1 and SCN5, at the function and vehicle integration test levels. On obtaining comprehensive function test coverage information across test levels, such redundancies can be avoided by ensuring the function and vehicle integration test levels cover different scenarios of the function.
- On analysing the system level function test coverage information, it was identified that most of the test gaps at the system level lie in three of the six systems involved.
- With this data, any change can be traced from the system it is pertaining to, to the appropriate function scenarios that it effects. This reduces the effort in analyzing the change impact at the vehicle integration test level.

Besides successfully obtaining the desired results, two critical gaps which have the capability to potentially hamper the effectiveness of the strategy were also identified. These include missing system-level requirements and missing test reports at the function test level. Another critical result obtained was the identification of a potential error in the function design for one of its variant which was unnoticed previously. This opens up the interesting possibility to more extensively study the

Use Case Variant	Function Scenario (SCN)	TW1602			TW1606			TW1610		
		System Test Level	Function Test Level	Integration Test Level	System Test Level	Function Test Level	Integration Test Level	System Test Level	Function Test Level	Integration Test Level
Liquid	SCN1	61.11%	100%	100%	77.78%	100%	100%	72.22%	100%	100%
	SCN2	57.89%	Not Tested	Not Tested	73.68%	Not Tested	Not Tested	73.68%	Not Tested	Not Tested
	SCN3	50%	Not Tested	Not Tested	71.42%	Not Tested	Not Tested	64.28%	Not Tested	Not Tested
	SCN5	52.94%	100%	100%	76.47%	100%	100%	52.94%	100%	100%
	SCN6	40%	Not Tested	Not Tested	50%	Not Tested	Not Tested	50%	Not Tested	Not Tested
	SCN4	26.67%	Not Tested	Not Tested	46.67%	Not Tested	Not Tested	46.67%	Not Tested	Not Tested
Gas	SCN1	53.33%	Not Tested	100%	46.67%	Not Tested	100%	40%	Not Tested	100%
	SCN5	43.75%	Not Tested	Not Tested	75%	Not Tested	Not Tested	50%	Not Tested	Not Tested
	SCN6	40%	Not Tested	Not Tested	50%	Not Tested	Not Tested	50%	Not Tested	Not Tested

Fig. 4. Comprehensive Fuel Level Display function scenario test coverage across the test levels for three test rounds

effectiveness of the proposed strategy for enhancement of distributed embedded software function design through scenario-based requirements specification.

While the implementation of the strategy in a real world setting helped study its practical feasibility and effectiveness, it also helped identify that its efficiency will likely reach its highest potential with the incorporation of suitable tool support and right people knowledge at steps of the strategy where maximum effort in terms of time and cost was invested. These steps include manual generation of the function scenario models and gathering information spread across multiple data sources to establish the desired traces across the test levels. Hence, these aspects that account to maximum effort in implementation present areas where the efficiency of the strategy can most likely be further enhanced. The current study therefore sets the process focused platform for moving towards a comprehensive solution for the issues with the current approach to test distributed embedded software functions in the automotive industry.

5.3 Threats to Validity

Referring to Runeson et al. [20], a description of the validity threats, along with the counter measures adopted to reduce them to the best of the researchers' ability are presented in this section.

Construct Validity. Construct validity deals with whether the operational measures studied accurately represent what is being investigated. In the conducted research, one construct validity threat was a possibility of designing interview questions for the case study which may be misinterpreted by the interviewees and lead to collecting irrelevant data. This risk was controlled by reviewing and revising the interview questionnaire and by performing simultaneous data collection and data analysis so the questionnaire can be readjusted if deemed necessary. Moreover, triangulation methods were employed to ensure the validity of what was being studied.

Repeatability and Reliability. Repeatability deals with whether the research conducted is repeatable, hence making it reliable. There was a risk that the case study conducted will not be adequately documented to make it repeatable. This risk was controlled by using the case study protocol to design and well document the research. Moreover, the actions undertaken throughout the research were reviewed and discussed among the researchers. Such auditing has been found to help mitigate this risk [31].

Internal Validity. Internal validity deals with how causal relationships are examined and relevant conclusions are drawn. In the conducted research, there is a possible risk of poor data analysis leading to incorrect conclusions. This risk was countered by discussing the results in length among the researchers and other senior practitioners at the case company. Moreover, the results of data analysis were backed with adequate literature support to validate the conclusions drawn. However, there still exists a possible internal validity threat that exists in this research due to the consideration of limited test data for drawing conclusions

from the implementation of the proposed strategy. This threat can be reduced further by focusing future work on studying the results of implementation of the verification strategy on more extensive test data spread across several test rounds.

External Validity/Generalisability. External validity deals with the extent to which the findings can be generalized. Since the research was conducted within a single automotive company, the findings may not hold in other cases. This risk was reduced to a great extent by explaining in sufficient detail the context within which the study was conducted in Sect. 2.2. Another major external validity threat lies in the implementation of the proposed strategy for only one distributed embedded software function. These threats exist and can be mitigated by focusing future work on implementing the strategy to a larger number of more complex distributed functions and studying its feasibility and effectiveness within other automotive companies.

6 Related Work

Modelling requirements for verification is not a new concept. It has been adopted for various verification activities, most popularly for testing and is termed MBT. MBT has been found to help significantly enhance the testing process [4]. A semi-formal approach, based on the use of scenarios for modelling requirements within MBT is being increasingly studied in literature [2, 15, 28]. A large part of the research that focuses on using such semi-formal model-based approach to aligning requirements specification and testing through traceability has been found to involve proposing solutions but little research focus is on evaluating and validating these proposed solutions [4].

Tsai et al. in [28] explore scenario-based traceability in order to select test cases for impact analysis and regression testing of changes. The authors propose the use of scenarios in regression testing of system software function changes for reduced test effort through improved impact analysis. While fundamentally it is similar to the work in the current research in terms of exploring scenarios for improved testing, it differs in aspects where the current research aims to capture scenarios and propose their suitability in reducing test gaps and test redundancies and improving test effort distribution across multiple test levels of the automotive industry V-model.

Nebut et al. in [15] present work on deriving function tests from use cases in the form of test scenarios through test objectives. The major focus of the presented research, similar to [28], is on the automation of the process of using scenario-based requirements specification and testing. Where as, in the current research, the focus is on studying and validating the effectiveness of using scenarios for testing and traceability in the automotive industry.

Within the context of embedded system software, specifically in the automotive industry, most previous related work in MBT and subsequent study on traceability and alignment of the disciplines of requirements specification and testing is based on formal methods [13, 24]. While some studies focus on the

concept of test scenarios as a means of managing the requirements of the increasingly complex automotive embedded software, they take a formalized approach to the generation and use of test scenarios like in [7].

Moreover, most of the testing focus through such formal modelling methods is concentrated at the system integration and lower test levels of the automotive V-model [23]. While some apply their approaches to distributed functions at the vehicle integration test level like in [24], focus on this area in general is limited. MBT at the vehicle integration test level of the automotive test process and more specifically a semi-formal scenario-based MBT with established traces to lower-level system requirements and testing is unknown today but with great potential for improvement [6]. Such a multi-level reuse of test effort has been recognized for test effort reduction during test case generation [16]. Within the current thesis research, this concept of multi-level reuse is dealt with a broader perspective of reducing test effort by providing a means to take more data-driven decisions to improve test coverage by identifying and mitigating test redundancies and test gaps. Such comprehensive test strategies to cope with the complex aspects of the automotive software testing process like [3], are limited for integration testing of the distributed software functions.

7 Summary and Conclusion

With the advent of software for the realization of several functions of a vehicle, there has been and continues to be several challenges that the automotive industry faces to which it was unfamiliar a few decades ago [10, 12, 29]. One such identified facet of the software-related challenges is in the area of system software testing [11, 27]. The automotive embedded software functions are increasingly distributed in nature, implying that the software modules for the realization of the functions are distributed across several ECU systems of the vehicle. This adds to the complexity of integration testing of the distributed software functions [27]. Research focusing on studying the challenges in integration testing of the distributed software functions [11, 27] and possible effective and feasible strategies to tackle these challenges [3] within the automotive industry is limited. Hence, the current research contributes to the area of integration testing of distributed automotive embedded software functions.

The scenario-based function requirements specification proposed as part of the process enhancement based verification strategy in this research has been found to provide an effective and feasible means to capture test-driven requirements in the form of function scenarios. This was found to make them suitable for establishing the desired traceability. Thereafter, the strategy proposed in this research presents a manner in which the requirements and testing data across test levels can be comprehensively tied together to aid in effectively performing the integration testing of the distributed embedded software functions within the automotive industry. Implementation of the proposed strategy in a real world setting has given promising results. The broad concept of multi-level reuse of test results across the test levels of the automotive V-model was found to provide an effective and feasible means of capturing comprehensive test coverage

information pertaining to the distributed software functions. Consequently, it has been identified that there is a critical link between requirements and testing that needs to be established across all test levels for implementing the multi-level reuse concept. Such traceability links provide a means to assess the effectiveness of the current approach to test the distributed software functions by identifying and mitigating test redundancies and test gaps across the test levels. It hence helps take more data-driven decisions for integration testing of the distributed software functions and thereby enhance function test coverage.

References

1. Adenmark, M.: Scania Test Levels, Scania Internal Document (REST08012) (2008)
2. Arnold, D., Corriveau, J.P., Shi, W.: Modeling and validating requirements using executable contracts and scenarios. In: Proceedings of the 8th ACIS International Conference on Software Engineering Research, Management and Applications (SERA), pp. 311–320. IEEE (2010)
3. Barhate, S.S.: Effective test strategy for testing automotive software. In: International Conference on Industrial Instrumentation and Control (ICIC), pp. 645–649. IEEE (2015)
4. Barmi, Z.A., Ebrahimi, A.H., Feldt, R.: Alignment of requirements specification and testing: a systematic mapping study. In: Proceedings of the 4th IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 476–485 (2011)
5. Von der Beeck, M., Margaria, T., Steffen, B.: A formal requirements engineering method for specification, synthesis, and verification. In: Proceedings of the 8th Conference on Software Engineering Environments, pp. 131–144. IEEE (1997)
6. Bringmann, E., Kramer, A.: Model-based testing of automotive systems. In: 1st IEEE International Conference on Software Testing, Verification, and Validation, pp. 485–493 (2008)
7. Conrad, M., Fey, I., Sadeghipour, S.: Systematic model-based testing of embedded automotive software. *Electronic Notes Theor. Comput. Sci.* **111**, 13–26 (2005)
8. Dhadyalla, G., Kumari, N., Snell, T.: Combinatorial testing for an automotive hybrid electric vehicle control system: a case study. In: IEEE 7th International Conference on Software Testing, Verification and Validation Workshops (ICSTW), pp. 51–57. IEEE (2014)
9. Ferrari, A., dell’Orletta, F., Spagnolo, G.O., Gnesi, S.: Measuring and improving the completeness of natural language requirements. In: Salinesi, C., Weerd, I. (eds.) REFSQ 2014. LNCS, vol. 8396, pp. 23–38. Springer, Heidelberg (2014). doi:[10.1007/978-3-319-05843-6_3](https://doi.org/10.1007/978-3-319-05843-6_3)
10. Grimm, K.: Software technology in an automotive company: major challenges. In: Proceedings of the 25th International Conference on Software Engineering, pp. 498–503. IEEE Computer Society (2003)
11. Kasoju, A., Petersen, K., Mäntylä, M.V.: Analyzing an automotive testing process with evidence-based software engineering. *Inf. Softw. Technol.* **55**(7), 1237–1259 (2013)
12. Broy, M.: Challenges in automotive software engineering. In: Proceedings of the 28th International Conference on Software Engineering, pp. 33–42. ACM (2006)

13. Marinescu, R., Saadatmand, M., Bucaioni, A., Seceleanu, C., Petterson, P.: A model-based testing framework for automotive embedded systems. In: Proceedings of the 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 38–47. IEEE (2014)
14. Marshall, M.N.: Sampling for qualitative research. *Family Pract.* **13**(6), 522–526 (1996)
15. Nebut, C., Fleurey, F., Le Traon, Y., Jezequel, J.M.: Automatic test generation: a use case driven approach. *IEEE Trans. Softw. Eng.* **32**(3), 140–155 (2006)
16. Perez, A.M., Kaiser, S.: Integrating test levels for embedded systems. In: Proceedings of the Testing: Academic and Industrial Conference - Practice and Research Techniques (TAIC PART), pp. 184–193. IEEE, September 2009
17. Praprotnik, O., Gartner, M., Zauner, M., Horauer, M.: A test suite for system tests of distributed automotive electronics. In: 2nd International Conference on Advances in Circuits, Electronics and Micro-electronics (CENICS), pp. 67–70. IEEE (2009)
18. Radeka, K.: The toyota product development system: integrating people, process and technology by James M. Morgan and Jeffrey K. Liker. *J. Prod. Innov. Manage* **24**(3), 276–278 (2007)
19. Rowley, J., Slack, F.: Conducting a literature review. *Manage. Res. News* **27**(6), 31–39 (2004)
20. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **14**(2), 131–164 (2008)
21. Saglietti, F.: Testing for dependable embedded software. In: 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 409–416. IEEE (2010)
22. Shah, U.S., Jinwala, D.C.: Resolving ambiguities in natural language software requirements: a comprehensive survey. *ACM SIGSOFT Softw. Eng. Notes* **40**(5), 1–7 (2015)
23. Shokry, H., Hinchey, M.: Model-based verification of embedded software. *IEEE Comput.* **42**(4), 53–59 (2009)
24. Siegl, S., Hielscher, K.S., German, R., Berger, C.: Formal specification and systematic model-driven testing of embedded automotive systems. In: Proceedings of the Europe Conference & Exhibition on Design, Automation & Test (DATE), pp. 1–6. IEEE (2011)
25. Somé, S.S.: Supporting use case based requirements engineering. *Inf. Softw. Technol.* **48**(1), 43–58 (2006)
26. Strauss, A., Corbin, J.: *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage Publications, Beverly Hills (1990)
27. Sundmark, D., Petersen, K., Larsson, S.: An exploratory case study of testing in an automotive electrical system release process. In: 6th IEEE International Symposium on Industrial Embedded Systems (SIES), pp. 166–175. IEEE (2011)
28. Tsai, W.T., Bai, X., Paul, R., Yu, L.: Scenario-based functional regression testing. In: Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC), pp. 496–501. IEEE (2001)
29. Weber, M., Weisbrod, J.: Requirements engineering in automotive development - Experiences and challenges. In: Proceedings of the IEEE Joint International Conference on Requirements Engineering, pp. 331–340 (2002)
30. Wilson, C.E.: Triangulation: the explicit use of multiple methods, measures, and approaches for determining core issues in product development. *Interactions* **13**(6), 46–47 (2006)
31. Yin, R.K.: *Case Study Research: Design and Methods*. Sage Publications (2013)