The Science of Testing: An Automotive Perspective

Author, co-author (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)
Affiliation (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)

Abstract

Increasing automation in the automotive systems has re-focused the industry's attention on verification and validation methods and especially on the development of test scenarios. The complex nature of Advanced Driver Assistance Systems (ADASs) and Automated Driving (AD) systems warrant the adoption of new and innovative means of evaluating and establishing the safety of such systems. In this paper, the authors discuss the results from a semi-structured interview study, which involved interviewing ADAS and AD experts across the industry supply chain.

Eighteen experts (each with over 10 years' of experience in testing and development of automotive systems) from different countries were interviewed on two themes: test methods and test scenarios. Each of the themes had three guiding questions which had some follow-up questions. The interviews were transcribed and a thematic analysis via coding was conducted on the transcripts. A two-stage coding analysis process was done to first identify codes from the transcripts and subsequently, the codes were grouped into categories.

The analysis of transcripts for the question about the biggest challenge in the area of test methods revealed two specific themes. Firstly, the definition of pass/fail criteria and secondly the quality of requirements (completeness and consistency). The analysis of the questions on test scenarios revealed that "good" scenario is one that is able to test a safety goal and ways in which a system may fail. Based on the analysis of the transcripts, the authors propose two types of testing for ADAS and AD systems: Requirements-Based Testing (traditional method) and Hazard Based Testing. The proposed approach not only generates test scenarios for testing how the system works, but also how the system may fail.

Introduction

The introduction of Advanced Driving Assistance Systems (ADASs) and Automated Driving (AD) systems in cars have many benefits ranging from increased safety [1,2], lower emissions, reduced traffic congestion [3,4] and more useful time for the driver [5]. The potential benefits of automated systems have led the push towards their commercialisation. Interestingly, the public opinion about *"completely self-driving (fully automated) vehicles"* has been shown to be in line with the proposed safety benefits [6]. Automated systems offer many benefits in other industries too where they have been introduced, e.g. aviation, nuclear, chemical process, railways etc. Unfortunately, the introduction of automation in these industries was coupled with many accidents, some of which have continued to repeat themselves [7]. Even within the automotive industry, many relatively advanced features (at the time), have caused vehicle re-calls due to faulty software, costing millions of dollars to the manufacturers; e.g. to fix the ignition switch issue, General Motors spent approximately $400 million for the 2.6 million affected vehicles [8]. Fixing a bug during the development process costs an average of $25, while after release it increases to $16000 on an average [9]. A bug in a released product could be caused due to: 1) incorrect requirements 2) missing requirements 3) release of untested code, 4) testing sequence differs from use sequence 5) user applied untested input values 6) untested operating conditions [10]. The latter was illustrated in the Ariane 5 disaster [11], where software was reused from Ariane 4 software in the Ariane 5 system without enough testing [12]. This importance of operating environment and potential consequence of untested inputs was also seen in the recent Tesla "Auto-pilot" system crash [13]. It has been suggested that majority of the software related accidents are a result of the operation of the software rather than its lack of operation [14].

Therefore, in order to realize the benefits of automation or any other system, we need to ensure that the systems have a safe and a robust functionality. This may be achieved by testing and certification of the systems. However, lack of standardized test methods and test scenarios; and the lack of international standards to define safety requirements for automated systems, have led to a subjective interpretation of "safety", particularly for ADAS and AD systems in vehicles. While the ISO 26262 standard [15], provides some guidance for testing methods and approaches for a product development cycle, it too falls short to deal with the complexities of ADAS and AD systems. Furthermore, even with ISO 26262 – 2011 been increasingly adopted in the industry, there is still a lack of a *"quantified and rigorous process for automotive certification"* [16]. This is caused due to the lack of objective quantification of severity, exposure and controllability ratings which comprise the ASIL rating, causing inter- and intra-rater variations [16,17].

Current luxury cars are a complex system with over 100 millions lines of code as compared to 7 million in a Boeing 747 airplane [18]. The introduction of ADASs and AD systems is going to further increase the complexity many fold with multiple interactions between subsystems. Additionally, ADASs and AD systems offer a new challenge for testing and safety analysis [19]. While a variety of ADASs and AD systems exist or are in development, each of them offers a different kind of a challenge for testing. The move towards higher levels of automation is coupled with the challenge of testing and safety analysis as it needs complex solutions to include interactions between a larger number of variables and the environment. It is suggested that in order to prove that automated vehicles are safer than human drivers, they will need to be driven for more than 11 billion miles [20]. Even after 11 billion miles, such testing will *"only assure safety but not always ensure it"* [21], thus suggesting vehicle level testing or real world testing before start of production (SOP) wouldn't be enough to prove safety of the automated driving systems [22,23]. While software testing has been said to be the *"least understood part of the (system) development process"* [10], the authors believe that a scientific approach needs to be adopted to solve the challenge of identifying scenarios that capture the complex interactions within systems and system-environment in an efficient manner.

Understanding Scenarios

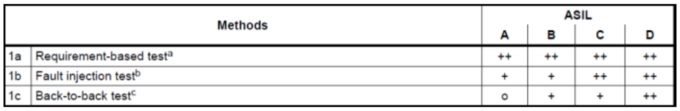
These complex interactions can be captured as use cases which *"describe the system behaviour as a sequence of actions linking the result to a particular actor"* (e.g. driver). Subsequently, scenarios (a specific sequence of a use case) present possible ways in which a system may be used to accomplish a desired function. However, writing scenarios require detailed domain knowledge, which is only found with experts. Moreover, the term "use cases" and "scenarios" have been used with a fuzzy meaning [24,25]. A use case is a collection of scenarios bound together by a common goal [25] and implies *"the way in which a user uses a system"* [26]. Scenarios have been suggested to have at least four different meanings: 1) scenarios to illustrate the system 2) scenarios for evaluation 3) scenarios for design 4) scenarios to test theories [24]. It is worth elucidating that a scenario that is good for illustrating a system demo (i.e. demonstration) may not be good for evaluating the basic functions (i.e. requirement based testing), as the former only uses a limited number of examples. Similarly, scenarios to test theories establish the strengths and more importantly the weaknesses of a design. Therefore, they go beyond the traditional requirement based testing.

Existing Requirements Based Testing (RBT) approach widely used in the industry, only ensures that the system meets its requirements while failing to identify the exceptions explicitly. Some exceptions may be covered sporadically due to the experience of historic failures rather than a scientific approach. Additionally, RBT is not able to ensure completeness of requirements. Requirements reflect the expert's view of system's functionality and possible usage. The identification of the requirements has a degree of subjectivity associated with it [27]. Different experts with different background knowledge analyse and classify systems differently, leading to an inter-rater variation in understanding requirements [17,28].

This is evident in the variation in the classification and identification of scenarios like the *"Black Swan"* scenarios or the *"unknown unknowns"* (scenarios that we don't know that we don't know) associated with the functionality of the system [29]. While requirements based testing captures the *"known knowns"* efficiently, the inability to ensure its completeness leads to the occurrence of *"unknown knowns"*, *"known unknowns"* and the Black Swan scenarios. In addition, to avoid the variation in understanding of the terms use-cases, scenarios and test cases, the authors adopt the definition as described in [30].

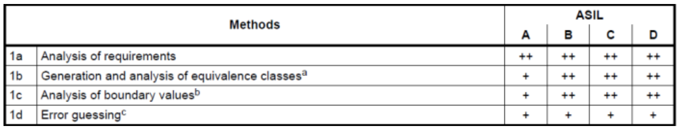
Types of testing

The international standard ISO 26262 [15] is the automotive industry best practice standard for functional safety. ISO 26262 Part 4 and Part 6 provide guidance on different methods for testing and for deriving test cases for software integration testing and software unit testing respectively. ISO 26262 – 2011 Part 4 and Part 6 recommend the use of test methods like requirement based test, fault-injection test and back-to-back comparison test (Figure 1). For each of the test methods, the standard recommends methods like analysis of requirements, analysis of equivalence classes, analysis of boundary values and error guessing as methods for deriving test cases (Figure 2).



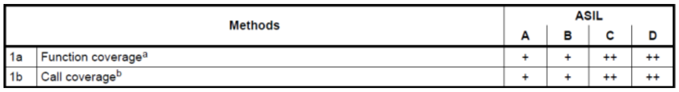
	Methods	ASIL			
		A	B	C	D
1a	Requirement-based test[a]	++	++	++	++
1b	Fault injection test[b]	+	+	++	++
1c	Back-to-back test[c]	o	+	+	++

Figure 1. Methods for testing functional safety and technical requirements as per ISO 26262 – 2011 Part 4



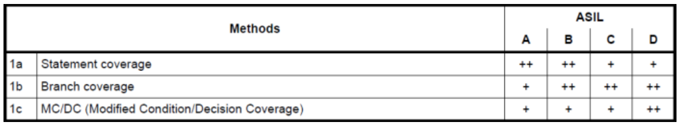
	Methods	ASIL			
		A	B	C	D
1a	Analysis of requirements	++	++	++	++
1b	Generation and analysis of equivalence classes[a]	+	++	++	++
1c	Analysis of boundary values[b]	+	++	++	++
1d	Error guessing[c]	+	+	+	+

Figure 2. Methods for deriving test cases for software unit testing as per ISO 26262 – 2011 Part 6



	Methods	ASIL			
		A	B	C	D
1a	Function coverage[a]	+	+	++	++
1b	Call coverage[b]	+	+	++	++

Figure 3. Software architecture level structural coverage metrics per ISO 26262 – 2011 Part 6



	Methods	ASIL			
		A	B	C	D
1a	Statement coverage	++	++	+	+
1b	Branch coverage	+	++	++	++
1c	MC/DC (Modified Condition/Decision Coverage)	+	+	+	++

Figure 4. Software unit structural coverage metrics as per ISO 26262 – 2011 Part 6

The ISO 26262 standard also recommends some metrics to measure the completeness of the testing process. These include coverage metrics like function coverage and call coverage at the integration level (Figure 3) and branch coverage, statement coverage and MC/DC (Modified Condition/Decision Coverage) at unit level (Figure 4).

As suggested by ISO 26262 – 2011, testers tend to go all-out for coverage metrics. While this is "a metric", it needs to be highlighted that it should be treated as the minimum metric. If achieving high coverage was the golden bullet for testing, then products in use would have very few bugs [10]. Since there are infinite possibilities for input suite, testers tend to use the *"best"* sample test to *"adequately"* test the system, where "best" and "adequately" is based on the subjective judgement of the tester [10].

While the advent of automated systems in automobiles has led to an increasing focus on incorporating functional safety in the design process, the current version of the standard fails to provide guidance on systems with high automation. The industry, acknowledging this gap has attempted to address it with the upcoming SOTIF (Safety Of The Intended Functionality) publically accepted specification [31]. This paper captures the essence of the gap in knowledge of testing for ADAS and AD systems and proposes a means to fill this gap.

Methodology

In order to understand the testing approach being undertaken by the automotive industry towards ADAS and AD systems to uncover the *"unknown unknown"*, *"unknown known"* and *"known unknown"* scenarios, the authors conducted a semi-structured interview study involving verification and validation experts in the automotive domain. Semi-structured interviews were conducted to understand the existing knowledge base for test scenario generation process in the automotive industry and their understanding and expectations from a good/ideal test scenario. Semi-structured interviews were adopted as they provide the flexibility to the interviewee to provide wider information and thus richer data, by enabling the formation of a understanding between the interviewer and the interviewee due to face to face contact [32]. Additionally, they allow the flexibility to examine topics in different degrees of depth (as per interviewees' interest and background) [33]. The interviews were transcribed and the text was sanitized to remove any proprietary mentions. A coding analysis was performed on the sanitized text and themes and categories were identified from the various interview answers. Coding analysis groups participant responses which are similar to give a broader understanding of responses.

In order to prevent any bias, the interviewees were allowed to talk freely while answering the questions and were not prompted for any answers. Participant interviews were transcribed into text and were later coded to perform thematic analysis. Key themes were identified in both parts of the interview.

Ethical approval for the study was secured from the University of Warwick's Biomedical & Scientific Research Ethics Committee (BSREC). All interview transcripts were anonymized and stored in a secure location and University of Warwick's data handling procedures were followed.

Participants

Eighteen industry experts, each participant having over 10 years' of experience in the field of testing and development of systems in the automotive industry were recruited for this study. Participants were selected from a diverse demography cutting across the automotive supply chain. Nine participants represented OEMs (Original Equipment Manufacturers), eight participants represented Tier 1/2 suppliers and the remaining participant represented academic /research organizations /start-ups working in the area of automated driving. To ensure independence of the interviewees, participants were recruited from different countries including the UK, Germany, India, Sweden, Japan and USA. The interviews lasted between 28.63 minutes and 103.15 minutes (average interview length: 48.25 minutes). Interviewees were also assured that any of the responses will not be identifiable to them as the transcripts would be anonymized before they were analysed.

Interview questions design

The interview was structured with six guiding questions, which were divided into two themes: 1) test methods (three questions) 2) test scenarios (three questions). Each guiding question had a set of follow-up questions, which were asked depending on the content of the answers. The set of follow-up (prompting) questions are described in Table 1. The follow-up questions were used to aid participants thought process and were designed to be minimally prescriptive to avoid biasing the answers. The guiding questions were formulated by the existing gaps in the literature. The six guiding questions were the following:

Test Methods

1. What test methods do you use for testing of automotive systems?
2. What are the challenges for each test method that you have faced?
3. What metric do you use to measure sign-off criteria for testing automated systems?

Test Scenarios

4. How do you ensure robust testing of automated automotive systems in various driving conditions?
5. How do you develop test scenarios for testing automated systems?
6. What criteria do you think make a good quality test scenario?

Table 1. Interview question design (follow-up questions)

Guiding Ques. #	**Follow-up Question(s)**
1	Reason for selecting a test method? What tools do you use as a part of your test setup?
2	What is your biggest challenge?
3	How was the metric developed? Is it a standard metric? (Company internal or industry standard)
4	What test scenarios do you use while doing real world / virtual testing?
5	What aspects are critical while developing a test scenario for autonomous system?
6	How did you develop those (for good quality test scenario) criteria?

Data Analysis

As this study employed a semi-structured interview format, the analysis of the data was mostly qualitative. In order to structure the data analysis and identify trends in the collected data, a coding strategy was used. A code *"is a word or short phrase that symbolically assigns a summative, salient, essence-capturing , and/or evocative attribute to a portion of language-based or visual data"* [34]. By reading through the transcribed interview text, codes were assigned to the text which enabled conversion of the interview text into an easy to understand tabulated format. An example of a code and corresponding text is discussed here. One of the responses to the question on the biggest challenge in testing faced by the interviewees was, *"it is difficult to create the specification to verify against and because of the lack of specification, it is difficult to put a criteria for completeness of testing"*. The corresponding code assigned to the text was *"how to ensure completeness of requirements"* and *"how to judge test completeness"*. However, it is evident that such a coding process is subjective due to the understanding and biases of the coder. In order to overcome this, a two stage coding process was followed which was reviewed by an independent expert.

First cycle coding

The first coding cycle involved reading through the interview transcripts to assign codes. As the data in a semi-structured interview transcripts can be varied, different methods of coding such as structural coding, descriptive coding, process and in-vivo coding, were used [34].

Second cycle coding and category identification

Since different coding methods were used in the first coding cycle, some of the codes were similar or split. In order to synthesize the first cycle codes to develop a more cohesive understanding, axial coding was used in the second phase which led to the creation of categories for the first cycle codes. Table 2 illustrates the coding process for the answers received to question 5.

Table 2. Development of codes for question 5

Participant answers	**1st cycle codes**	**2nd cycle codes**
“what kind of environmental influences could lead to an ill function”	Environmental factors, failures	- Identify failures, system limits, and hazards. - Using **systematic method** to identify failures, hazards
“try to define a test to see the degradation of the performance”	Degraded performance, faults	
“understand situation in which our system will reach any kind of limit ”	System limits, degraded performance	
“fidelity of test scenario comes down to the FMEA. Because out of the FMEA there is possibility of a failure you need a control method for”	Identify failures, systematic way, FMEA	
“we would engineer faults into the system…. Blocking the radar. Put radar absorbent material (RAM) for the radar.”	Create faults, block sensors	
“it is currently done via FMEA, System FMEA and Hazard Analysis”	Systematic way, FMEA, HARA	- Using **systematic method** to create test scenario library
“have a catalogue of tests”	Test library	
“systematic way (of) what kind of influencer I have into the behaviour of the functionality…”	Factors influencing functionality, systematic method	
“you can think about you have a matrix…then we look at what kind of combinations are possible”	Test library	

Results

While it was found that tools (software platforms) for test execution were not an issue for most organizations, the infrastructure requirement for test platforms (hardware-in-the-loop setup and instrumented test vehicles for real-world testing) had exponentially increased with ADAS and AD systems as compared to traditional automotive systems. In addition, the large amount of data handling required for sensors used in the ADAS and AD systems was another challenge.

In response to the first question on test methods used for testing, the participant responses could be grouped in two themes. One group of participants commented that they follow the software development V cycle and implemented model-based design tools using simulation in a major part of their development process. On the contrary the other group was of the opinion that simulation is of limited use for ADASs and AD systems as it is *“almost impossible”* to model sensors, especially RADAR and LiDAR sensors and they mostly depended on real world testing.

More importantly, the input to the test execution platform (test case vectors) was a common concern acknowledged by all participants. When asked about the biggest challenge faced by the participants while performing testing, two specific themes emerged. While the OEMs credited *“test case generation and definition of pass/fail criteria”* as their biggest challenge; tier 1/2 suppliers credited *“quality of requirements (including completeness and consistency)”* as their biggest challenge. This difference can be credited to the culture in the automotive supply chain where the suppliers develop individual systems and the responsibility for integration of these systems lies with the OEMs. However, both the groups failed to mention any solutions to the challenges faced by them during the testing phase; the ability to identify and define the *“known unknown”* and the *“unknown unknown”* scenario space.

When asked about the parameters and criteria for good test scenarios, there seem to be an agreement on the ability to test *“known unknown”* and *“unknown unknown”* situations, as a key feature of a good test scenario. However, a deeper analysis of the responses revealed two distinct themes on ways to achieve “good” test scenarios. Firstly, creating “good” scenarios from requirements is dependent on the skill and experience of the test specifiers. Secondly, “good” scenarios should be able to test safety goals and ways in which the system may fail or reach system limits. This is generally not covered by system requirements. Moreover, the need for a systematic method of identifying the system limits or failure scenarios was highlighted by the participants. Most experts mentioned that Requirements Based Testing (RBT) is insufficient as there is a challenge in ensuring completeness of requirements. RBT captures the typical scenarios as suggested by the requirements and represents the most common real world scenarios. Such testing ensures that the most common bugs are identified [10].

While approaches to improve requirement based testing have been discussed in literature [35,36], discussion on the ability to increase the *“known known”* by identifying the unknown space is limited. One of the reasons mentioned by experts about RBT was that it is impractical to have a requirements document capturing the multitude of scenarios an automated driving system might encounter, rendering the classical V-cycle for software development obsolete.

In the testing process, it is important to establish when to stop testing and sign-off the system-under-test. When the participants were asked about a metric used to measure the sign-off criteria, to our surprise, the answers demonstrated the lack of any standard metric in place. Unfortunately, the sign-off point was dependent on the budget allocated and SOP time. However, all participants acknowledged that this wasn’t the ideal situation and needs to change for ADASs and ADS systems. However, some participants did provide some insight into an ideal situation and using false positive and false negative rates as metric for sign-off.

When asked about how participants ensured that the ADASs and AD systems were tested robustly, they mentioned using a test catalogue which was developed from experience. However, all participants agreed that for ADAS and AD systems, more real world testing is needed due to challenges in simulation environment. On the time split between real-world and virtual testing, one of the participants commented: *"95% is real world testing and 5% is simulation. But for me it should 50-50. For the moment the robust model of the simulation is stopping (this to happen)"*.

Discussion

One of the challenges of identifying *"black swan"* scenarios is their lack of correlation with time [23]. Based on the analysis of the interviews, to increase the area covered by the *"known knowns"* in the test scenario space, the authors propose a two-pronged approach to testing of ADASs and AD systems to create test scenarios and test cases (Figure 5). The first branch concerns using traditional RBT approach, while the second branch uses a Hazard Based Testing (HBT) approach for creating test scenarios. Traditional RBT method covers only a fraction of the possible test scenario space for the systems (Figure 5). The addition of the second testing branch (HBT) improves the coverage of the test scenario space by increasing the *"known known"* scenario space. However, it does not guarantee full coverage of the test space (Figure 5). While RBT checks the working of the system as per expectations (defined requirements), HBT explores how the system may fail by identifying possible failure scenarios.

HBT draws its inspiration from the world of security analysis. In security analysis, the use of misuse cases has been suggested as a way of testing for security concerns [37]. Misuse cases can *"help document negative scenarios"* [37]. The key to the success of HBT is to have a structured, robust and well-documented method of identifying hazards. This was also highlighted in the themes obtained from the analysis of the interview transcripts. The two themes were: *"failure or hazard scenarios"* and *"systematic method (objective) to obtain them"*. On being asked about how to develop test scenarios, one of the participants commented: *"try to define a test to see the degradation of the performance",* while another participant mentioned: *"what kind of environmental influences could lead to an ill function"*.

In order to identify hazards, various methods like HAZOP , FMEA [38], Event Tree Analysis, JANUS [39], Accimaps [40], HFACS [41–43], Fault-tree analysis [44,45], bow-tie analysis [46], System Theoretic Process Analysis (STPA) / Systems Theoretic Accident Model & Processes (STAMP) [47–49] etc. have been used in the industry and research community. Some of these methods were developed for simple systems and fall short in analysing modern ADAS and AD systems which have multiple interactions between system, human operator and the software [50]. In case of an Adaptive Cruise Control (ACC) system, rather than testing the functional requirements, more emphasis needs to be laid on identifying the hazards associated with the usage of an ACC system. An analysis of the ACC system using any of the earlier said methods to identify hazards would lead to one of the potential hazards as *"unintended braking"*.

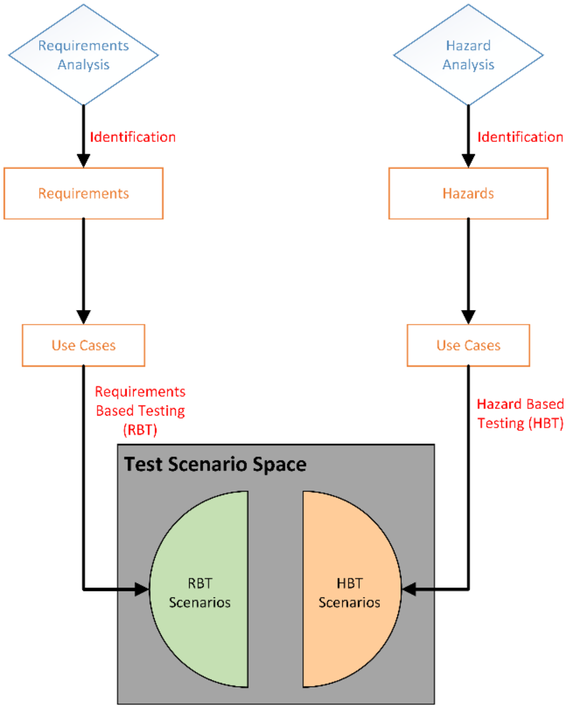


Figure 5. Proposed testing approach for test-scenario generation

Some of the hazard identification methods developed specifically for ADAS and AD system (e.g. HFACS, JANUS, STPA) further analyse the system interactions to identify that one of the potential causes of an *"unintended braking (hazard)"* could be the "vehicle maneuvering through a steep bend" causing the radar system to believe that there is an obstacle in front. Therefore an HBT approach would identify such situations which would have been missed in a traditional RBT approach.

In order to identify the safety goals and the hazards, a Hazard Analysis and Risk Assessment (HARA) process needs to be conducted. The automotive HARA has its own issues like subjective variation due to skill and experience of the testers and completeness of the HARA, some of these issues have been answered in the literature [17]. Once the systems have been tested, their capability and safe performance can be correctly established and can form part of the knowledge to be imparted to the drivers, in real time or before they start their usage, establishing their "informed safety" level to improve trust in ADASs and AD systems [51]. However, in order to create the "informed safety" level, a more systematic and structured process needs to be adopted to testing. As one of the interviewees mentioned, *"Testing is a science"*.

In this study, the authors had a limited sample size for the interview pool due to resource constraints. While a large number of practitioners are involved in the field of verification and validation, it would be a major challenge to interview a representative sample size. However, due to the expertise of the interviewees, the authors believe that the current findings provide an important insight in the future direction for testing of automated automotive systems, which will be an essential component of the system development process.

Conclusion

The lack of a scientific approach to testing has led to the inability of the industry to tackle the challenges offered by ADAS and AD systems in terms of testing. The authors interviewed 18 automotive experts, each with over 10 years' of experience in testing and development of automotive systems. "Creating test scenarios" and "ensuring completeness of requirements" were highlighted as the main challenges for testing of ADAS and AD systems. However, none of the experts could provide a solution to any of the two challenges.

Moreover, the experts suggested that a "good" test scenario is one that tests how the system fails or reaches its limits, in addition to having a structured approach to define the test scenario. Requirement based testing tends to elude capturing this test space, and thus according to the experts is not enough when it comes to testing ADASs and AD systems.

Based on a detailed analysis of the interview transcripts, the authors have proposed a new approach for testing to increase the coverage of the test scenario space. This has been achieved by reducing the occurrence of *"Black Swan"* scenarios (i.e., unknown unknowns) and *"known unknowns"*, by increasing the *"known knowns"* of the system. The proposed method comprises of a two-pronged approach to identifying test scenarios. The first branch comprises of the traditional requirement based testing (RBT) method, while the second branch comprises of a hazard based testing (HBT) approach. The latter requires the identification of hazards for a system. The *"known unknowns"*, *"unknown knowns"* and *"unknown unknowns"* get uncovered in the Hazard Based Testing branch. Safety goals, which give rise to hazards, are identified by conducting a Hazard Analysis and Risk Assessment (HARA) process. The proposed approach not only generates test scenarios for testing how the system works, but also how the system may fail, thus increasing the test scenario space.

References

1. Carbaugh, J., Godbole, D.N., and Sengupta, R., "Safety and capacity analysis of automated and manual highway systems," Transp. Res. Part C Emerg. Technol. 6(1–2):69–99, 1998, doi:10.1016/S0968-090X(98)00009-6.

2. Fagnant, D.J. and Kockelman, K., "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," 3rd ed., SAGE, ISBN 0965-8564, 2015, doi:10.1016/j.tra.2015.04.003.

3. Arem, B. van, Cornelie, J.G.V.D., and Visser, R., "The impact of Co-operative Adaptive Cruise Control on traffic flow characteristics," IEEE Trans. Intell. Transp. Syst. 7(4):429–436, 2005.

4. Shladover, S.E., "Cooperative (rather than autonomous) vehicle-highway automation systems," IEEE Intell. Transp. Syst. Mag. 1(1):10–19, 2009, doi:10.1109/MITS.2009.932716.

5. Vine, S. Le, Zolfaghari, A., and Polak, J., "Autonomous cars: The tension between occupant experience and intersection capacity," Transp. Res. Part C Emerg. Technol. 52:1–14, 2015, doi:10.1016/j.trc.2015.01.002.

6. Schoettle, B. and Sivak, M., "Public Opinion About Self-Driving Vehicles in China, India, Japan, The U.S., The U.K. and Australia," ISBN UMTRI-2014-21, 2014, doi:UMTRI-2014-30.

7. Coze, J.C. Le, "New models for new times. An anti-dualist move," Saf. Sci. 59:200–218, 2013, doi:10.1016/j.ssci.2013.05.010.

8. Malek, S., "What Software Developers Can Learn From the Latest Car Recalls," http://it-cisq.org/what-software-developers-can-learn-from-the-latest-car-recalls/, 2017.

9. Altinger, H., Wotawa, F., and Schurius, M., "Testing methods used in the automotive industry: results from a survey," Proc. of the 2014 Workshop on Joining AcadeMiA and Industry Contributions to Test Automation and Model-Based Testing - JAMAICA 2014, ISBN 9781450329330, 2014, doi:10.1145/2631890.2631891.

10. Whittaker, J.A., "What is software testing? And why is it so hard?," IEEE Softw. 17(1):70–79, 2000, doi:10.1109/52.819971.

11. Lions, J.L., "Ariane 5 Flight 501 Failure: Report by the Inquiry Board," 1996.

12. Weyuker, E.J., "Testing component-based software: A cautionary tale," IEEE Softw. 15(5):54–59, 1998, doi:10.1109/52.714817.

13. NHTSA, "Investigation Report: PE 16-007 (MY2014-2016 Tesla Model S and Model X)," 2017.

14. Leveson, N.G., "New Safety Technologies for the Automotive Industry," 3rd ed., SAGE, Detroit, MI, USA, 2006.

15. ISO, "Road vehicles — Functional safety (ISO 26262)," SAGE, 2011.

16. Yu, H., Lin, C.-W., and Kim, B., "Automotive Software Certification: Current Status and Challenges," SAE Int. J. Passeng. Cars - Electron. Electr. Syst. 9(1):2016-01–0050, 2016, doi:10.4271/2016-01-0050.

17. Khastgir, S., Birrell, S., Dhadyalla, G., Sivencrona, H., and Jennings, P., "Towards increased reliability by objectification of Hazard Analysis and Risk Assessment (HARA) of automated automotive systems," Saf. Sci. 99:166–177, 2017, doi:10.1016/j.ssci.2017.03.024.

18. Charette, R.N., "This car runs on code," IEEE Spectr. 46(3), 2009.

19. Khastgir, S., Birrell, S., Dhadyalla, G., and Jennings, P., "Identifying a gap in existing validation methodologies for intelligent automotive systems: Introducing the 3xD simulator," 3rd ed., SAGE, ISBN 9781467372664, 2015, doi:10.1109/IVS.2015.7225758.

20. Kalra, N. and Paddock, S.M., "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," Transp. Res. Part A Policy Pract.

94(December):182–193, 2016, doi:10.1016/j.tra.2016.09.010.

21. Transport Systems Catapult, “Taxonomy of Scenarios for Automated Driving,” 2017.

22. Koopman, P. and Wagner, M., “Challenges in Autonomous Vehicle Testing and Validation,” SAE Int. J. Transp. Saf. 4(1):2016-01–0128, 2016, doi:10.4271/2016-01-0128.

23. Wachenfeld, W. and Winner, H., “The New Role of Road Testing for the Safety Validation of Automated Vehicles,” Automated Driving, ISBN 978-3-319-31893-6: 419–435, 2017, doi:10.1007/978-3-319-31895-0_17.

24. Campbell, R.L., “Will the real scenario please stand up?,” ACM SIGCHI Bull. 24(2):6–8, 1992, doi:10.1145/142386.1054872.

25. Cockburn, A. and Fowler, M., “Question time! about use cases,” ACM SIGPLAN Not. 33(10):226–229, 1998, doi:10.1145/286942.286960.

26. Cockburn, A., “Structuring use cases with goals,” J. Object Oriented Program. 1997(5):1–16, 1997.

27. Flage, R. and Aven, T., “Emerging risk – Conceptual definition and a relation to black swan type of events,” Reliab. Eng. Syst. Saf. 144:61–67, 2015, doi:10.1016/j.ress.2015.07.008.

28. Ergai, A., Cohen, T., Sharp, J., Wiegmann, D., Gramopadhye, A., and Shappell, S., “Assessment of the Human Factors Analysis and Classification System (HFACS): Intra-rater and inter-rater reliability,” Saf. Sci. 82:393–398, 2016, doi:10.1016/j.ssci.2015.09.028.

29. Aven, T., “On the meaning of a black swan in a risk context,” Saf. Sci. 57:44–51, 2013, doi:10.1016/j.ssci.2013.01.016.

30. Khastgir, S., Dhadyalla, G., Birrell, S., Redmond, S., Addinall, R., and Jennings, P., “Test Scenario Generation for Driving Simulators Using Constrained Randomization Technique,” 2017, doi:10.4271/2017-01-1672.Copyright.

31. Griessnig, G. and Schnellbach, A., “Development of the 2nd Edition of the ISO 26262,” in: Stolfa, J., Stolfa, S., O’Connor, R., and Messnarz, R., eds., Systems, Software and Services Process Improvement. EuroSPI 2017. Communications in Computer and Information Science, Springer, Cham, ISBN 978-3-319-64217-8: 535–546, 2017, doi:10.1007/978-3-319-64218-5.

32. Louise Barriball, K. and While, A., “Collecting data using a semi-structured interview: a discussion paper,” J. Adv. Nurs. 19(2):328–335, 1994, doi:10.1111/j.1365-2648.1994.tb01088.x.

33. Robson, C. and McCartan, K., “Real world research : a resource for users of social research methods in applied settings,” 4th ed., Wiley, ISBN 9781405182416, 2016.

34. Saldaña, J., “The coding manual for qualitative researchers,” Third, SAGE, 2016.

35. Robinson-Mallett, C.L., “An approach on integrating models and textual specifications,” 2012 2nd IEEE Int. Work. Model. Requir. Eng. MoDRE 2012 - Proc. 92–96, 2012, doi:10.1109/MoDRE.2012.6360079.

36. Robinson-Mallett, C., Grochtmann, M., Köhnlein, J., Wegener, J., and Kühn, S., “Modelling requirements to support testing of product lines,” ICSTW 2010 - 3rd Int. Conf. Softw. Testing, Verif. Valid. Work. 11–18, 2010, doi:10.1109/ICSTW.2010.65.

37. Alexander, I., “Misuse cases: Use cases with hostile intent,” IEEE Softw. 20(1):58–66, 2003, doi:10.1109/MS.2003.1159030.

38. Stamatis, D.H., “Failure mode and effect analysis : FMEA from theory to execution,” 2nd ed., Milwaukee, Wisc. : ASQ Quality Press, 2003, ISBN 9780521190817, 2003.

39. Hoffman, R.R., Lintern, G., and Eitelman, S., “The Janus Principle,” IEEE Intell. Syst. 19(2):78–80, 2004, doi:10.1109/MIS.2004.1274915.

40. Salmon, P.M., Cornelissen, M., and Trotter, M.J., “Systems-based accident analysis methods: A comparison of Accimap, HFACS, and STAMP,” Saf. Sci. 50(4):1158–1170, 2012, doi:10.1016/j.ssci.2011.11.009.

41. Chen, S.T., Wall, A., Davies, P., Yang, Z., Wang, J., and Chou, Y.H., “A Human and Organisational Factors (HOFs) analysis method for marine casualties using HFACS-Maritime Accidents (HFACS-MA),” Saf. Sci. 60:105–114, 2013, doi:10.1016/j.ssci.2013.06.009.

42. Baysari, M.T., Caponecchia, C., McIntosh, A.S., and Wilson, J.R., “Classification of errors contributing to rail incidents and accidents: A comparison of two human error identification techniques,” Saf. Sci. 47(7):948–957, 2009, doi:10.1016/j.ssci.2008.09.012.

43. Wiegmann, D. and Shappell, S., “Applying the human factors analysis and classification system (HFACS) to the analysis of commercial aviation accident data,” Proc. of the 11th International Symposium on Aviation Psychology, Columbus, Ohio, 2001.

44. Lee, W.S., Grosh, D.L., Tillman, F.A., and Lie, C.H., “Fault Tree Analysis, Methods, and Applications - A Review,” IEEE Trans. Reliab. R-34(3):194–203, 1985, doi:10.1109/TR.1985.5222114.

45. Reay, K. a. and Andrews, J.D., “A fault tree analysis strategy using binary decision diagrams,” Reliab. Eng. Syst. Saf. 78(1):45–56, 2002, doi:10.1016/S0951-8320(02)00107-2.

46. Abimbola, M., Khan, F., and Khakzad, N., “Risk-based safety analysis of well integrity operations,” Saf. Sci. 84:149–160, 2016, doi:10.1016/j.ssci.2015.12.009.

47. Leveson, N.G., “Applying systems thinking to analyze and learn from events,” Saf. Sci. 49(1):55–64, 2011, doi:10.1016/j.ssci.2009.12.021.

48. Leveson, N., “A new accident model for engineering safer

systems," Saf. Sci. 42(4):237–270, 2004, doi:10.1016/S0925-7535(03)00047-X.

49. Leveson, N.G., "Engineering a Safer World," The MIT Press, ISBN 9780262016629, 2011.

50. Fleming, C.H., Spencer, M., Thomas, J., Leveson, N., and Wilkinson, C., "Safety assurance in NextGen and complex transportation systems," Saf. Sci. 55:173–187, 2013, doi:10.1016/j.ssci.2012.12.005.

51. Khastgir, S., Birrell, S., Dhadyalla, G., and Jennings, P., "Calibrating Trust on Automation in Vehicles through Knowledge : Introducing the concept of informed safety," Transp. Res. Part C Emerg. Technol. (Under Review).

Contact Information

Siddartha Khastgir
WMG, University of Warwick, UK.
Address: International Digital Laboratory, WMG, University of Warwick, UK, CV4 7AL
S.Khastgir@warwick.ac.uk

Stewart Birrell
WMG, University of Warwick, UK.
Address: International Digital Laboratory, WMG, University of Warwick, UK, CV4 7AL
S.Birrell@warwick.ac.uk

Gunwant Dhadyalla
WMG, University of Warwick, UK.
Address: International Digital Laboratory, WMG, University of Warwick, UK, CV4 7AL
G.Dhadyalla@warwick.ac.uk

Paul Jennings
WMG, University of Warwick, UK.
Address: International Digital Laboratory, WMG, University of Warwick, UK, CV4 7AL
Paul.Jennings@warwick.ac.uk

Acknowledgments

This work has been carried out under the VVIL project (Validation of Vehicle-in-the-Loop), funded by Innovate UK under the grant ref. 132302 and is a collaboration between Vertizan Limited and WMG, University of Warwick, UK. The authors would like to thank EPSRC (Grant EP/K011618/1) and WMG centre HVM Catapult, for providing the necessary infrastructure for carrying out this study. WMG is one of the seven centres that together comprise the High Value Manufacturing Catapult in the UK.