A System-based Safety Assurance Framework for Human-Vehicle Interactions

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

With the introduction of vehicular digitization and automation, there has been significant growth in the number of Electronic Control Units (ECUs) inside vehicles, followed by the broader use of Advanced Driver Assistance Systems (ADAS) and Automated Driving Systems (ADSs). The growth of the number of ECUs has also significantly increased the number of user interfaces. To conduct safe driving, in addition to those related to the real-time control of the vehicle, a driver also needs to be able to digest information effectively and efficiently from various ECUs via the Human-Machine Interface (HMI). To evaluate the safety of ADS, including its interactions with system users, some work has suggested that they will need to be driven for over 11 billion miles. However, the number of test miles driven is not a meaningful metric for judging safety. Instead, the types of scenarios encountered by the driver-vehicle interactions during testing are critically important. With a hazard-based testing approach, this paper proposes that the extent to which testing miles are 'smart miles' that reflect hazard-based scenarios relevant to potential unsafe driver-vehicle interactions is fundamental. The authors proposed an extension based on STPA's Human Mental Model to create hazard-based test scenarios related to human-machine interactions. The proposed approach has been applied to a real-world project associated with the testing of an SAE-Level 4 Autonomous Vehicle (AV) during its prototyping phase, which involves the interactions between the safety driver and the AV's ADS and X-by-Wire system. The authors also proposed an extension to the Scenario Description Language (SDL) that can be used to define hazard-based test scenarios. The test scenarios generated from the extended SDL have been used for scenario-based testing in real-world and simulation environments.

Introduction

In the past few decades, there has been an increase in the amount of digitization and automation in safety-critical systems in various industries. This includes aviation, railway, marine, health care, automotive, etc. In the automotive domain, there can be as many as 150 Electronic Control Units (ECUs) with over 100 million lines of code in a modern highly engineered vehicle as opposed to only a few ECUs equipped in the vehicle that was manufactured back in the 90s [1][2]. The growth in the number of ECUs has also significantly increased the number of user interfaces [3], which includes but are not limited to in-car touch screens and buttons, push rotary controllers, swipe and gesture functions, and even speech recognition technology. To conduct safe driving, in addition to those related to the real-time control of the vehicle, a driver also needs to be able to digest and respond to the information from various ECUs via the Human-Machine Interface (HMI) effectively and efficiently [4]. The role of human drivers has therefore gradually shifted from controlling the vehicle to monitoring the in-vehicle controllers [5].

As the complexity of in-vehicle human-automation interaction has increased, so has the number of operational modes and the number of ways of triggering those operational modes [6]. As highlighted in [7] that one of the challenges in evaluating the risk and safety of complex systems with safety-critical applications is that the knowledge of overall (system-level) activity is poor. Such a challenge is not only applicable to the system users who have very limited information about the system, but also to the system developers who were unable to identify the unknown hazards of the system. Traditional methods, such as Event Tree Analysis (ETA), Fault Tree Analysis (FTA), and Bow-tie Method, view a complex system as a collection of independent components with linear relationships. However, one of the key features of complex systems is non-linearity and the dependent nature of causal links caused by feedback loops [8]. With complex systems, especially those involving human-automation interactions, accidents emerge due to the diverse interactions with a wide and open system – i.e., emergent behaviors [9]. Such emergent behaviors could trace back to the driver's ability to form an adequate mental model of the vehicle, and the behavior of the in-vehicle controllers responding to the control commands from the driver [10]. The challenges associated with identifying hazards of complex systems due to emergent behaviors suggest the need for new approaches [11][12].

In the automotive domain, to prove that ADSs are safe, it is suggested in [13] that there will be a need to drive ADSs for more than 11 billion miles. One of the reasons behind the suggestion is that it would potentially capture all possible 'black swan' scenarios and known unknown scenarios [14]. However, the identification of unknown scenarios of complex systems remains a challenge for test engineers and safety analysts [15]. This is because of the nature of the black swan scenarios which has an extremely low probability of occurrence, but with a significant consequence if occurred [16].

Whilst a safety claim could involve the ADS vehicle running on a sunny, clear, and straight road for 11 billion miles, such a claim may not be sufficient or representative to justify that the ADS is safe within its Operational Design Domain (ODD) where dynamic traffic and weather conditions occur [17]. Therefore, instead of ensuring the quantity of the tested miles, it is more critical and necessary for test engineers to focus on their quality as a way of identifying the 'unknown unknowns (i.e., black swan) and 'known unknown' scenarios.

Requirement-Based Testing and Hazard-based Testing

There has been Requirement-Based Testing (RBT) widely used to evaluate the performance and reliability of autonomous vehicles and their built-in ECUs [18]. RBT enables the test engineers to evaluate whether the system under test satisfies the pre-defined requirements – i.e., it captures the 'known knowns' efficiently. However, the coverage of the 'known knowns' cannot be guaranteed by RBT, which could lead to the 'unknown knowns', 'known unknowns, and the 'unknown unknowns / black swan' scenarios. Hazard-Based Testing (HBT) was therefore introduced to complement the traditional RBT used in the automotive domain [8]. Compared to RBT which evaluates how a system works, HBT aims to understand how a system fails or misbehaves. The aim of using HBT is to increase the 'known knowns' area and decrease the 'unknown unknowns' area and it has the following three steps [8]:

- Identifying hazards
- Creating test scenarios for the identified hazards
- Identifying pass criteria for the created test scenarios

The first step of the process involved the selection of suitable hazard identification methods. There were various methods explored during this step, including FMEA [19], FTA [20], HAZOP [21], and STPA [22]. In the system development context, STPA was selected as the hazard identification method as it can be applied at the early stage of the system development, and it also considers the human operator as part of the analysis. The advantages of STPA over other hazard identification methods, and its application to identify human mental model flaws, were presented in our previous paper [6]. Therefore, the focus of this paper was on the last two steps.

Formatting the Test Scenarios

An important aspect of developing and storing test scenarios is the need to appreciate the diversity of its end users. This includes test engineers, simulation engineers, regulators, the public, etc. The Scenario Description Language (SDL) was developed to format the test scenarios into a universal format that can be easily interpreted and used by different end-users [23]. However, formatting STPA-based test scenarios using SDL is still an open question. This paper also proposes a process that maps the elements of STPA Test Scenarios with those in SDL.

Research Questions

Having identified the research gaps for the safety of driver-vehicle interactions, this paper aims to answer the following two research questions:

- How to create test scenarios for driver-vehicle interactions for the identified hazards using STPA?
- How to format the test scenarios that are derived from STPA in an executable format?

Therefore, the rest of the paper is organized as follows: section II proposes the methodology to create STPA-inspired test scenarios and to format the test scenarios using SDL. Section III illustrates the results from a real-world case study of the application of the proposed method. Section IV discusses the results. Section V concludes the paper and proposes possible future works.

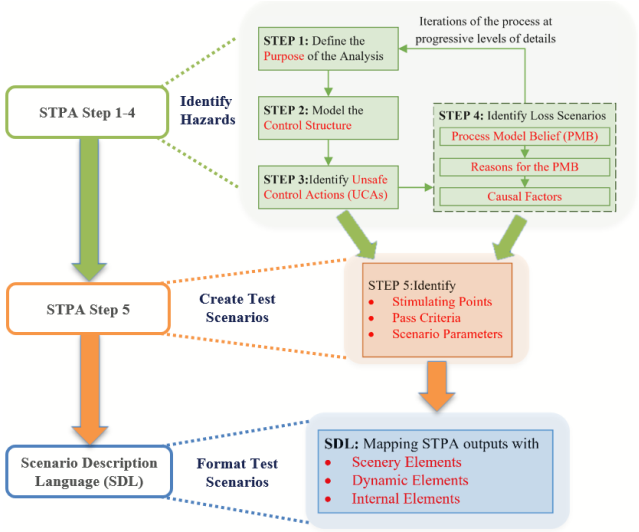

Figure 1. Flowchart of Creating and Formatting Hazard-based Test Scenarios

Methodology

In this section, the methods to create and format the test scenarios are introduced as illustrated in the second and the third blocks in Figure 1. The process of identifying hazards using STPA (as in the first block) can be found in our previous work [6]. The test scenarios are created by reusing the outputs from STPA.

Extending STPA to create Test Scenarios

Before starting the test scenario generation, it is important to understand how a test scenario should be constructed. According to [24], a test scenario should consist of a world, actors, and their behavior. However, to make the test scenario usable for testing, it was suggested in [8] that 'pass criteria' need to be included as a way of understanding what criteria must be met within a test scenario for a vehicle to receive a 'passing' score for its performance or safety. Furthermore, for a hazard to occur, there must be a certain condition that causes the system to be in a hazardous status. Such conditions could be caused by inadequate decisions made by the controllers, inappropriate implementation of the decisions, or miscommunications between the controllers. For example, when a driver is driving in the evening, there is a roundabout that was recently built in the middle of the road. The driver is relying on the map displayed on the sat-nav to understand the forward road layouts. However, the in-vehicle infotainment system does not support automatic update, and the map is showing a straight road instead of a roundabout. As a result, the driver might cross the roundabout without any deceleration. The condition that could trigger the driver to cross the roundabout is the straight road displayed on the map. To understand how the driver could behave under such conditions in hazard-based testing, it is necessary to ensure that the road layout shown on the map is not aligned with the ground truth, or in other words, the condition needs to be stimulated to form hazard-based testing. Therefore, a complete hazard-based test scenario will have six components, including scenery, environment, dynamic element, pass criteria, additional context, and stimulating point.

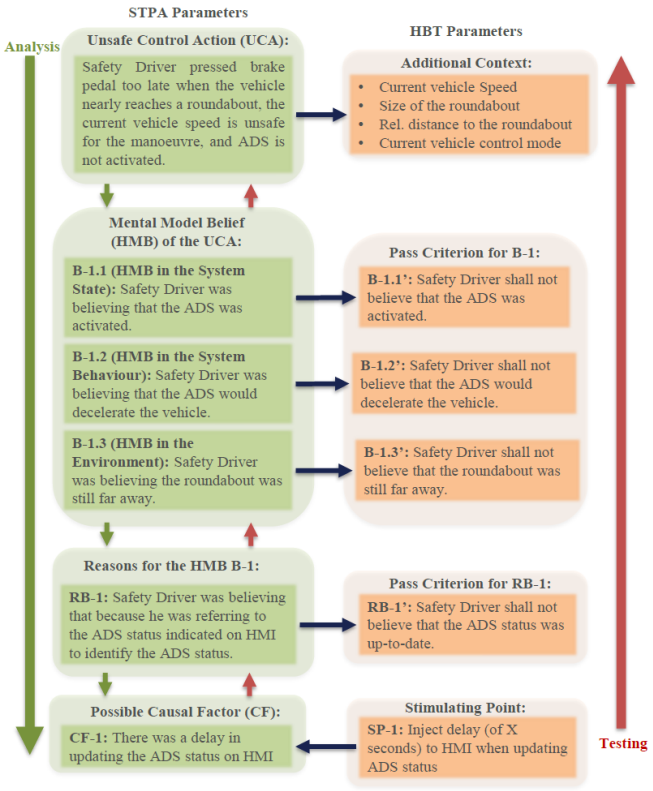

Figure 2. Flowchart of Identifying STPA and its Testing Parameters for the example UCA

Scenery includes all geo-spatially stationary objects in the Operational Domain (OD) of the vehicle [24], with attributes such as road signs, traffic lights, road furniture, etc. Environment describes the physical conditions outside the vehicle such as weather, lighting, connectivity, etc. Dynamic elements include all moving objects and actors in the OD of the vehicle. For example, pedestrians, animals, and road users are all categorized as dynamic elements. A pass criterion defines the set of conditions for which the test scenario will be considered a pass. The combinations of the scenery, environment, and dynamic elements form the base parameters of the test scenario.

Additional context refers to the context element of the Unsafe Control Action (UCA) – i.e., it describes the circumstance under which the behavior of a controller is unsafe either due to not providing a control action or providing a control action [25]. Consider a UCA – 'Safety Driver pressed brake pedal too late when the vehicle nearly reaches a roundabout, the current vehicle speed is unsafe for the maneuver, and ADS is not activated'. The context of the UCA is 'when the vehicle nearly reaches a roundabout, the current vehicle speed is unsafe for the maneuver, and ADS is not activated'. Several parameters can be extracted from the context, including 'current vehicle speed', 'size of the roundabout', 'distance to the roundabout', and 'current vehicle mode (i.e., manual control mode or auto-control mode)' etc. When selecting the context parameters and base parameters, the context parameters have a higher priority over base parameters. For example, in a hazard-based test scenario, the base parameters include a 'sharp bend in front of the vehicle' as part of the scenery element. From the same UCA that was discussed earlier, a 'roundabout' was extracted that was also in front of the vehicle. The context parameter 'roundabout' is selected here for the test scenario instead of the 'sharp bend' because the context parameter extracted from STPA is more of interest to us in terms of understanding how the safety driver behaves when approaching the roundabout. The difference between the context parameters and base parameters is that in the execution of test cases for the test scenario, the context parameters will have a higher resolution when parameter values are chosen.

Both pass criteria and stimulating points for the test scenario are derived from STPA Step 4. The goal of STPA Step 4 is to identify possible causal factors (CF) of the UCA and safety requirements to prevent the UCA from happening. For a UCA to occur, the process model of the controller could represent a belief that makes the control action it is directing appear to be safe (when it is unsafe, i.e., a UCA) [8]. When the controller is a human operator such as a driver, safety supervisor, or remote operator, the human mental model is used rather than the process model [26]. To understand how the human mental model was operating that could trigger the UCA, it is important to understand what the human operator was believing about the system state, the system behavior, and the environment at the time the UCA was happening [6]. Considering the same UCA – 'Safety Driver pressed brake pedal too late when the vehicle nearly reaches a roundabout, the current vehicle speed is unsafe for the maneuver, and ADS is not activated', when the UCA occurs, the safety driver's mental model could be equipped with several beliefs. Firstly, the safety driver could believe that ADS was activated (i.e., the belief in the system state); secondly, the safety driver could believe that the ADS would decelerate the vehicle (i.e., the belief in the system behavior); and lastly, the safety driver could believe that the roundabout is still far away (i.e., the belief in the environment). Let us call these three beliefs B-1.1, B-1.2, and B-1.3, and group them as B-1. If one or multiple elements of B-1 were not true, the safety driver would not direct the original control action (i.e., the original UCA). Therefore, one of the pass criteria for the test scenario would be defined as the negation of the human mental model belief B-1 – i.e., B-1'. Secondly, there must be some reasons behind the human mental model belief. Let us call these reasons causing the human mental model belief RB-1. Once again, if RB-1 were not valid, then B-1 would not be true, and the UCA would not be triggered. Therefore, the second pass criterion for the test scenario is the negation of the reasons for the human mental model belief – i.e., RB-1'. Both B-1' and RB-1' form the pass criteria of one of the test scenarios for the UCA. Figure 2 illustrates the process of deriving the loss scenarios and pass criteria from the analysis outputs.

A stimulating point indicates how the test engineers or simulation engineers could trigger the CF of the UCA that is identified in STPA Step 4. The intention of including a stimulating point is to trigger the CF in each Hazard-Based Test Scenario, while each scenario consists of the selected context parameters and base parameters. The goal is to understand if the system could turn into an unsafe state or if the driver's behavior could be affected due to the CF. For example, considering the aforementioned human mental model belief B-1 – 'The safety driver was believing that ADS was activated, ADS would decelerate the vehicle, and the roundabout is still far away, one of the reasons for B-1 (i.e., RB-1) could be that 'the safety driver was referring to the ADS status indicated on HMI to identify the ADS status'. One of the CFs could therefore be 'there was a delay in updating the ADS status on HMI'. Let us call this causal factor CF-1. To stimulate CF-1, a delay will be injected into the HMI so that there is a delay in updating or displaying the ADS status when the vehicle is approaching the roundabout (i.e., SP-1). The process of deriving the CF-1 and its corresponding SP-1 is also illustrated in Figure 2.

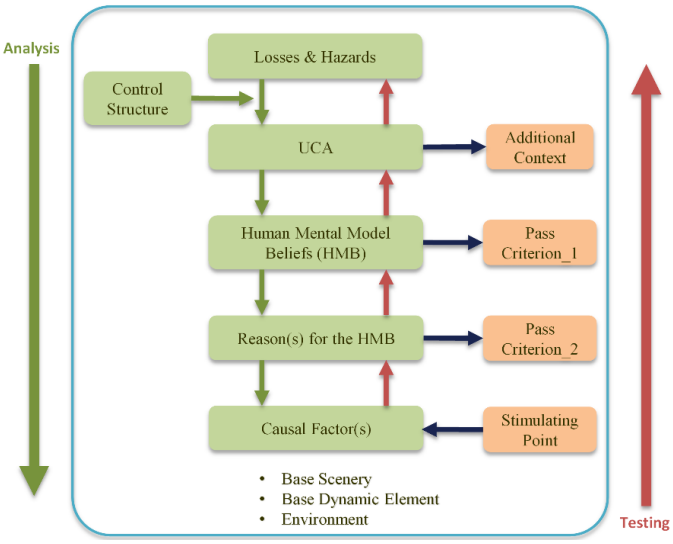

Figure 3. Flowchart of STPA-based Analysis and Testing

Based on the abovementioned examples, a proposed framework of creating STPA-based Hazard Based Test scenario elements is illustrated in Figure 3. During the analysis phase, the flow of the process follows the green arrows (i.e., from top to bottom). To create STPA test scenarios, the stimulating point, pass criteria, and additional context are derived from the outputs of STPA at different stages. During the testing phase, the flow of the process follows the red arrows (i.e., from bottom to top). To start with, the test scenario is set up to align with the base scenery, dynamic element, environment, and additional context. The 'Stimulating Point' is then injected to trigger the CF. It will then be of interest to the test or simulation engineer to understand if the CF could trigger the 'Reasons for the HMB' and consequently the 'Mental Model Belief' by observing and comparing them with the pre-determined pass criteria.

Formatting STPA Test Scenarios using Scenario Description Language (SDL)

There are two levels of abstraction in SDL – i.e., Level-1 and Level-2 [23]. SDL Level-1 describes a scenario at the functional level, which is more readable to the end-users. SDL Level-2 describes a scenario at the logical and concrete level, which is more readable to the machine. Both SDL Level-1 and Level-2 consist of two groups of elements: 1) SDL Scenery Element, and 2) SDL Dynamic Element. Similar to those in the base parameters, 'SDL Scenery Elements' consist of all the objects that are not capable of changing geographic positions even though some might be capable of changing states. 'SDL Scenery Elements' also includes the environment such as weather conditions, lighting, and connectivity. 'SDL Dynamic Elements' describe the behaviors of all the actors that can change geographic positions. Therefore, the base scenery parameters and environment are mapped with 'SDL Scenery Element', and the base dynamic parameters are directly mapped with 'SDL Dynamic Element'.

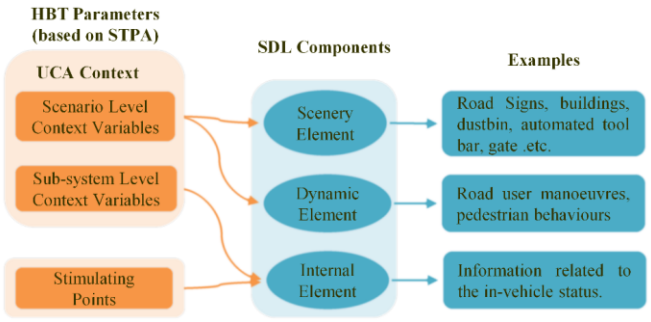

Figure 4. Mapping STPA Parameters with SDL Elements

Before mapping the STPA-derived elements (i.e., context parameters and stimulating points) into SDL, it is important to understand what group of elements can be included in a UCA context and the Stimulating Point. Considering the same UCA – 'Safety driver pressed brake pedal too late when the vehicle nearly reaches a roundabout, the current vehicle speed is unsafe for the maneuver, and ADS is not activated.' In the context 'when vehicle nearly reaches a roundabout, the current vehicle speed is unsafe for the maneuver, and ADS is not activated', both 'SDL Scenery Element' (i.e., roundabout) and 'SDL Dynamic Element' (i.e., the vehicle) are captured, which can be further parameterized as 'roundabout type', 'rel. distance to the roundabout', and 'vehicle motion' etc. Because these elements can be visualized from outside the vehicle, let us categorize them as 'Scenario-level Context Variable' (Scenario-level CV). Another important part of the context describes the internal configurations of the vehicle system at the time the UCA happens. For example, without the configuration that 'ADS is deactivated', the delay of brake request from the safety driver would not cause hazards (i.e., the UCA is invalid). The 'ADS is deactivated' can be parameterized as 'ADS Status', and since it cannot be visualized from outside the vehicle, let us categorize it as 'Subsystem-level Context Variable' (Subsystem-level CV). It should be noted that any context of the UCA describing the status of the driver or passenger (if available) is also categorized as 'Subsystem-level CV' because both driver and passengers are inside the vehicle. To enable SDL to describe the internal status of the vehicle under test, we extend the existing SDL structure by adding another group of elements called 'SDL Internal Element'. This covers all the elements that describe the parameters inside the vehicle under test.

When the UCA occurs, the process model or the human mental model of the controller has a belief that it is directing a safe control action, which could be unsafe, leading to a UCA. The CFs behind these beliefs could be due to the miscommunications between the controllers, flawed control algorithm or thinking process that processes the inputs from other controllers or the environment, or the inadequate execution process of the control actions. CFs describe the status of elements inside the vehicle that could trigger the UCA. Therefore, the stimulating points of the CFs are also mapped with 'SDL Internal Element' in SDL. Figure 4 illustrates the mapping between STPA elements and SDL.

Case Study

This section presents our case study of generating and formatting the test scenarios based on the proposed framework. The results (i.e., the UCA and its loss scenarios) of our previous paper submitted in [6] were reused to derive the test scenario parameters.

Creating the Test Scenarios

Consider the UCA – 'Safety Driver does not press the brake pedal when the vehicle under test (VUT) is entering a lower nominal speed area and ADS is disabled'. One of the loss scenarios identified in the paper is presented as shown below.

Firstly, one of the Safety Driver's human mental model beliefs in the vehicle system (i.e., B-1) was identified:

- B-1.1: Safety Driver was believing that ADS was activated. (i.e., the belief in the Process State)
- B-1.2: Safety Driver was believing that ADS would slow down the VUT automatically. (i.e., the belief in the Process Behavior)
- B-1.3: Safety Driver was believing that the speed limit would change. (i.e., the belief in the Environment)

Secondly, one of the reasons for the Safety Driver's human mental model belief B-1 was identified:

- RB-1: Safety Driver was believing that because he was referring to the ADS Status displayed on the HMI to identify the activation status of ADS.

And lastly, one of the possible causal factors for RB-1 was identified:

- CF-1: The HMI could have incorrectly displayed the ADS Status.

Based on the UCA and loss scenario, the test scenario parameters were then derived.

Pass Criteria

The pass criteria were derived as the negations of the human mental model belief B-1 and its reason RB-1:

Pass Criterion for B-1:

- Safety Driver shall not believe that ADS was activated. (i.e., the negation of B-1.1)
- Safety Driver shall not believe that ADS would decelerate the VUT. (i.e., the negation of B-1.2)

Pass Criterion for RB-1:

- Safety Driver shall not believe that the received ADS Status from HMI was correct. (i.e., the negation of RB-1)

It is important to note that the pass criterion for B-1.3 is not needed in this test scenario as it aligns with the statement in the UCA - i.e., the speed limit did change.

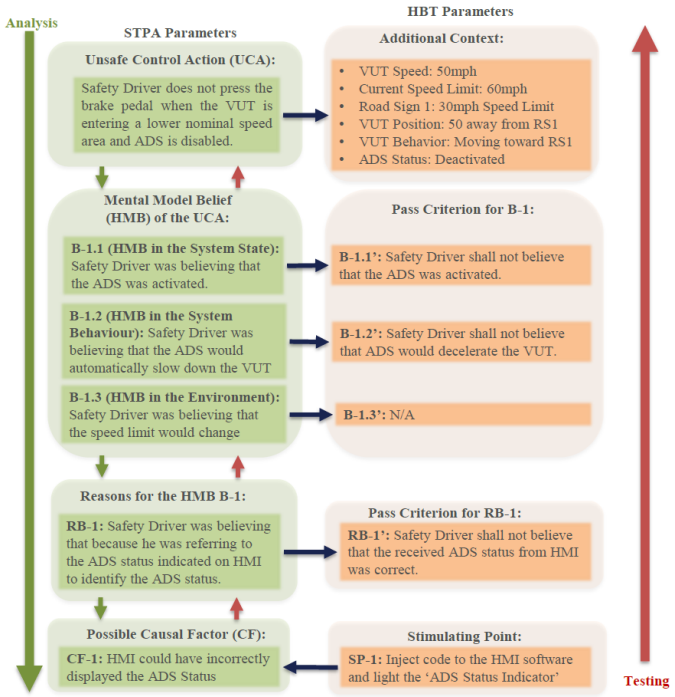

Figure 5. Flow chart of deriving HBT parameters for the case study

Stimulating Point

The stimulating point was derived from the CF as a way of understanding how to trigger the CF. Therefore, the stimulating point for CF-1 was identified as:

- SP-1: Inject code to the HMI software and light the 'ADS Status Indicator', indicating that ADS is activated (when it is deactivated).

Additional Context

The context of the UCA is: 'when the vehicle is entering a lower nominal speed area and ADS is disabled'. The context parameters that can be extracted from the context include:

- VUT Speed: 50mph
- Current Speed Limit: 60mph
- Road Sign 1: 30mph Speed Limit
- VUT Position: 50m away from Road Sign 1
- VUT Behavior: Moving toward Road Sign 1
- ADS Status: Deactivated

The process of deriving STPA and HBT parameters were summarized as illustrated in Figure 5.

Base Scenery, Dynamic Element, and Environment

The base parameters for scenery, dynamic element, and environment of this test scenario were selected from the ODD taxonomy in [27]. Whilst there can be a lot of different combinations of the base parameters selected for this test scenario, it is important to note that the selected base parameters shall not affect the decision-making of the Safety Driver. For example, there shall not be any other vehicles

(as part of the dynamic element) following closely behind the driver's vehicle as the driver could be concerned about a tailgating accident due to decelerating the vehicle.

Formatting the Test Scenarios

Once all the STPA test scenarios and test scenario parameters had been created, they were then formatted using SDL. Table 1 illustrates the mappings between STPA parameters and SDL Categories. The context variable 'ADS Status: Deactivated' was categorized as a Subsystem-level CV as it represents the scenario parameter inside the vehicle. Both Stimulating Point 'ADS Status Indicator: ON' and Subsystem-level CV 'ADS Status: Deactivated' were categorized as 'SDL Internal Element' in SDL. The rest of the context variables (i.e., 'VUT Speed', 'Current Speed Limit', 'Road Sign 1', 'VUT Position', and 'VUT Behavior') were categorized as scenario-level as they reflect the scenario parameters outside the vehicle. The Scenario-level CVs 'VUT Speed', 'VUT Position', and 'VUT Behavior' were categorized as 'SDL Dynamic Elements' as they describe the status of the VUT. The rest of the Scenario-level CVs were categorized as 'SDL Scenery Elements' as they describe the surrounding traffic rules.

A picture of the test scenario and its description in SDL was created as shown in Figure 6 and Figure 7 respectively. In the 'SDL Scenery Element' Category, all the STPA parameters identified earlier were captured including the 'Current Speed Limit' and the 'Road Sign 1'. The rest of the Scenery Elements describes the base scenery parameters for the test scenario. This includes the road network (defined as Network 1) that consists of two lanes L1 and L2 as well as the 'Road traffic direction', 'Lane markings', Road surface', 'Road edge features', and 'Road Structure'.

Table 1. Mapping between STPA Parameters and SDL Categories

STPA Parameters	STPA Category	SDL Category
ADS Status Indicator: ON	Stimulating Point	Internal
VUT Speed: 50 mph	Scenario-level CV	Dynamic
Current Speed Limit: 60 mph	Scenario-level CV	Scenery
Road Sign 1: 30mph Speed Limit	Scenario-level CV	Scenery
VUT Position: 50m away from Road Sign 1	Scenario-level CV	Dynamic
VUT Behavior: Moving toward Road Sign 1	Scenario-level CV	Dynamic
ADS Status: Deactivated	Subsystem-level CV	Internal

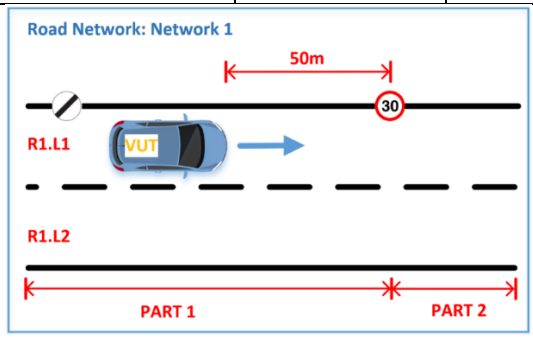

Figure 6. Diagram of the Test Scenario in Case Study

In the 'SDL Dynamic Element' Category, the STPA parameters 'VUT Position', 'VUT Behavior', and 'VUT Speed' were also captured. The behavior of the VUT was described in three phases. In Phase 1, the VUT was approaching 'Road Sign 1' with a speed of 50mph and rel. direction as rear side right (RSR) to 'Road Sign 1'. In Phase 2, the VUT came past the 'Road Sign 1' with the same speed and rel. direction as right (R) to 'Road Sign 1'. And lastly, in Phase 3, the VUT is moving away from 'Road Sign 1' with the same speed and rel. the direction of the front side right (FSR) to 'Road Sign 1'.

In the 'SDL Internal Element' Category, the subsystem-level CV 'ADS Status' was defined as 'Deactivated' throughout the test scenario. The timing of 'Internal Phase 1' is synchronized with the 'Phase 1' in the 'SDL Dynamic Element' Category. This means that when the VUT is approaching 'Road Sign 1', the stimulating point is activated, and therefore the 'ADS Status Indicator' on HMI is turned ON. The aim is to see if the safety driver could incorrectly believe that ADS is activated and then decide not to press the brake pedal (as per the pass criteria for B-1 and RB-1).

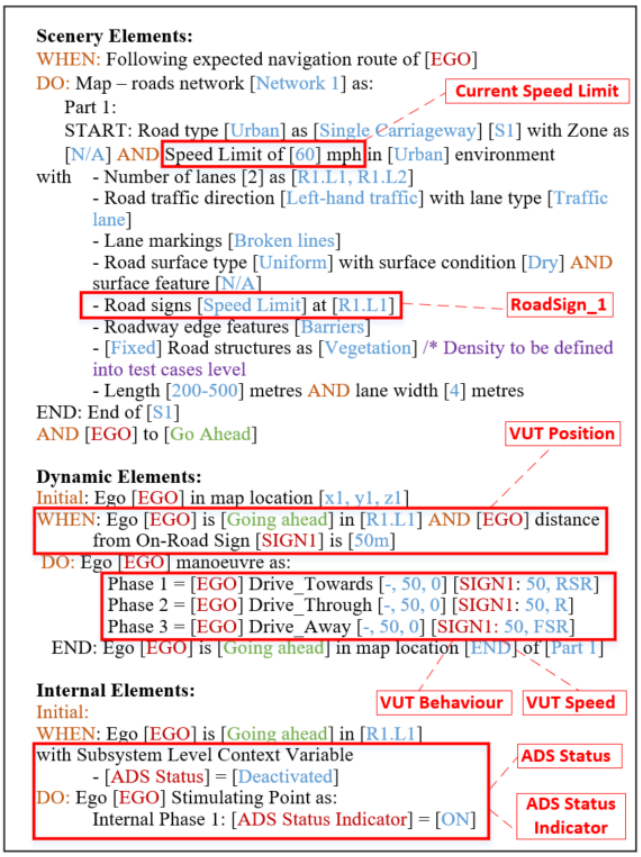

Figure 7. SDL Script of the Test Scenario in Case Study

Discussion

The increase in the amount of digitization and automation in automobile systems has expanded the complexities in the interactions between human drivers and the available user interfaces. Proving that human-vehicle interactions are safe has become a lot more challenging. This is due to the uncertainty of capturing all possible 'black swan' scenarios and known unknown scenarios. The concept

of hazard-based testing for human-vehicle interactions was therefore proposed in this paper.

The first step of hazard-based testing was to identify all possible hazards. STPA was selected as the hazard identification method in this work. The reasons for choosing STPA, the process, and its findings can be found in our previous publication [6]. The second and third steps were to create test scenarios for the identified hazards associated with the human-vehicle interactions and the pass criteria of the corresponding test scenarios. The extension to the STPA's human mental model to generate test scenarios for human-vehicle interactions was therefore presented in this paper. The generation of test scenarios involved two stages: firstly, the hazard-based test scenario parameters were derived from the outputs of the STPA analysis (i.e., UCAs and Loss Scenarios); and secondly, the base parameters for scenery, dynamic elements, and environment were selected from the ODD taxonomy in [27].

Once the hazard-based test scenarios were created, it is also important to expand the usability of these test scenarios. An extension to the existing Scenario Description Language (SDL) was then presented in this paper, by adding the 'SDL Internal Element' category to the existing framework. The SDL-formatted test scenarios can be used both in real-world testing and simulation environment. Therefore, for safety-critical testing involving driver-vehicle interactions, the 3XD simulator can be used [28]. In either way of testing, it is important to understand how the safety driver's mental model could be affected. It is therefore necessary to conduct regular psychological counseling with the relevant safety drivers to ensure that their daily driving behaviors are not affected by their experiences involved in the test scenarios.

As a way of evaluating the applicability of the proposed approach, we applied it to a real-world project of an SAE Level-4 Autonomous Vehicle during its prototyping phase. The application of the proposed method led to the creation of over 600 test scenarios involving the safety driver. While this may be considered a large set of test scenarios to execute for the safe driver, all the test scenarios were created to examine how well they align with the actual causes of losses. In other words, they represent the 'smart miles' of the required 11 billion test miles. There have been studies undertaken that compare various safety analysis methods, which have shown that STPA has more coverage in identifying system flaws as compared to other approaches [22][29]. To improve the efficiency of converting STPA test scenario parameters in SDL script, we have also developed an automation toolchain to automatically import and convert the parameters. The detail of the toolchain is outside the scope, and therefore it is not elaborated on in this paper.

Conclusion

A framework for hazard-based testing of human-vehicle interactions was presented in this paper. This includes the creation of test scenario parameters based on STPA's human mental model and the formatting of STPA-derived test scenarios. The proposed framework was applied to a real-world project related to the development and testing of an SAE Level-4 Autonomous Vehicle, which involves the interactions between the safety driver and the ADS and the X-by-Wire system of the vehicle. In total, we generated 635 test scenarios to test the driver-vehicle interactions in various environments and traffic conditions. Of the 635 test scenarios, 276 were related to interactions between the driver and the brake pedal, 77 were related to the interactions between the driver and the electric parking brake, 73 were related to enabling/disabling the ADS, 85 were related to the interactions between the driver and the acceleration pedal, and 124 were related to the interactions between the driver and steering wheel.

Although the 635 hazard-based test scenarios capture the 'smart miles' of the suggested 11 billion test miles, leading to a massive reduction of test scenarios, it is still important to reduce the amount of time of setting up the test scenarios both in real-world testing and in a simulation environment. Future research work will focus on how to link and combine multiple test scenarios into one, by extracting and analyzing keywords from their corresponding UCAs.

References

[1] J. Deichmann, B. Klein, G. Scherf, and R. Stuetzle, "The race for cybersecurity: Protecting the connected car in the era of new regulation," McKinsey Insights, no. October, p. N.PAG-N.PAG, 2019, [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&AuthType=ip&db=bsu&AN=139073472&site=ehost-live

[2] R. N. Charette, "This car runs on code," IEEE Spectr., vol. 46, no. 3, p. 3, 2009.

[3] X. Han and P. Patterson, "The effect of information availability in a user interface (UI) on in-vehicle task performance: A pilot study," Int. J. Ind. Ergon., vol. 61, pp. 131–141, 2017, doi: 10.1016/j.ergon.2017.05.015.

[4] F. Biondi, I. Alvarez, and K. A. Jeong, "Human–Vehicle Cooperation in Automated Driving: A Multidisciplinary Review and Appraisal," Int. J. Hum. Comput. Interact., vol. 35, no. 11, pp. 932–946, 2019, doi: 10.1080/10447318.2018.1561792.

[5] E. J. Rossetter and J. C. Gerdes, "Role of handling characteristics in driver assistance systems with environmental interaction," Proc. Am. Control Conf., vol. 4, no. June, pp. 2528–2532, 2000, doi: 10.1109/acc.2000.878648.

[6] S. Chen, S. Khastgir, I. Babaev, and P. Jennings, "Identifying Accident Causes of Driver-Vehicle Interactions Using System Theoretic Process Analysis (STPA)," Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., vol. 2020-October, pp. 3247–3253, 2020, doi: 10.1109/SMC42975.2020.9282848.

[7] A. Jensen and T. Aven, "A new definition of complexity in a risk analysis setting," Reliab. Eng. Syst. Saf., vol. 171, no. April 2017, pp. 169–173, 2018, doi: 10.1016/j.ress.2017.11.018.

[8] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings, "Systems Approach to Creating Test Scenarios for Automated Driving Systems," Reliab. Eng. Syst. Saf., vol. 215, no. December 2019, p. 107610, 2021, doi: 10.1016/j.ress.2021.107610.

[9] T. Melman, N. Beckers, and D. Abbink, "Mitigating undesirable emergent behavior arising between driver and semi-automated vehicle," 2020, [Online]. Available: http://arxiv.org/abs/2006.16572

[10] L. Morra, F. Lamberti, F. Gabriele Prattico, S. La Rosa, and P. Montuschi, "Building Trust in Autonomous Vehicles: Role of Virtual Reality Driving Simulators in HMI Design," IEEE Trans. Veh. Technol., vol. 68, no. 10, pp. 9438–9450, 2019, doi: 10.1109/TVT.2019.2933601.

[11] C. Chen, G. Reniers, and N. Khakzad, "Integrating safety and security resources to protect chemical industrial parks from man-made domino effects: A dynamic graph approach," Reliab. Eng. Syst. Saf., vol. 191, no. April, p. 106470, 2019, doi: 10.1016/j.ress.2019.04.023.

[12] E. Denney, G. Pai, and I. Whiteside, "The role of safety architectures in aviation safety cases," Reliab. Eng. Syst. Saf., vol. 191, no. June, p. 106502, 2019, doi: 10.1016/j.ress.2019.106502.

[13] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?,” Transp. Res. Part A Policy Pract., vol. 94, pp. 182–193, 2016, doi: 10.1016/j.tra.2016.09.010.
[14] R. Pawson, G. Wong, and L. Owen, “Known knowns, known unknowns, unknown unknowns: The predicament of evidence-based policy,” Am. J. Eval., vol. 32, no. 4, pp. 518–546, 2011, doi: 10.1177/1098214011403831.
[15] T. Bjerga, T. Aven, and E. Zio, “Uncertainty treatment in risk analysis of complex systems: The cases of STAMP and FRAM,” Reliab. Eng. Syst. Saf., vol. 156, pp. 203–209, 2016, doi: 10.1016/j.ress.2016.08.004.
[16] N. Bellomo, M. A. Herrero, and A. Tosin, “On the dynamics of social conflicts: Looking for the Black Swan,” Kinet. Relat. Model., vol. 6, no. 3, pp. 459–479, 2013, doi: 10.3934/krm.2013.6.459.
[17] P. Irvine, X. Zhang, S. Khastgir, E. Schwalb, and P. Jennings, “A Two-Level Abstraction ODD Definition Language: Part I*,” Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., pp. 2614–2621, 2021, doi: 10.1109/SMC52423.2021.9658751.
[18] B. Vaysburg, “Requirement-based automated black-box test generation,” 2001.
[19] H. Arabian-Hoseynabadi, H. Oraee, and P. J. Tavner, “Failure Modes and Effects Analysis (FMEA) for wind turbines,” Int. J. Electr. Power Energy Syst., vol. 32, no. 7, pp. 817–824, 2010, doi: 10.1016/j.ijepes.2010.01.019.
[20] W. S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie, “Fault Tree Analysis, Methods, and Applications - A Review,” IEEE Trans. Reliab., vol. R-34, no. 3, pp. 194–203, 1985, doi: 10.1109/TR.1985.5222114.
[21] J. Dunjó, V. Fthenakis, J. A. Vílchez, and J. Arnaldos, “Hazard and operability (HAZOP) analysis. A literature review,” J. Hazard. Mater., vol. 173, no. 1–3, pp. 19–32, 2010, doi: 10.1016/j.jhazmat.2009.08.076.
[22] M. S. Matter, “Modeling and Hazard Analysis using STPA,” 2012.
[23] X. Zhang, S. Khastgir, and P. Jennings, “Scenario Description Language for Automated Driving Systems: A Two Level Abstraction Approach,” Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., vol. 2020-October, pp. 973–980, 2020, doi: 10.1109/SMC42975.2020.9283417.
[24] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, “Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving,” IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC, vol. 2015-October, pp. 982–988, 2015, doi: 10.1109/ITSC.2015.164.
[25] N. G. Leveson, Engineering A Safer World, vol. 33, no. 3. 2017. doi: 10.1080/13623699.2017.1382166.
[26] M. France and J. Thomas, “Engineering for humans: a new extension to systems theoretic process analysis,” 19th Int. Symp. Aviat. Psychol., 2017.
[27] S. Khastgir, “PAS 1883:2020 Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification,” Bsi, 2020.
[28] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, “Identifying a gap in existing validation methodologies for intelligent automotive systems: Introducing the 3xD simulator,” IEEE Intell. Veh. Symp. Proc., vol. 2015-August, no. Iv, pp. 648–653, 2015, doi: 10.1109/IVS.2015.7225758.
[29] R. S. Martínez, “System Theoretic Process Analysis of Electric Power Steering for Automotive Applications,” pp. 1–197, 2015, [Online]. Available: http://sunnyday.mit.edu/STAMP/Sotomayor-Thesis.pdf

Contact Information

Mailing address: WMG, University of Warwick, Coventry, United Kingdom. Email address: Shufeng.chen@warwick.ac.uk. Telephone: +447719694974

Acknowledgements

The work presented in this paper has been carried out under the Innovate UK and Centre for Connected and Autonomous Vehicles (CCAV) funded OmniCAV project (Grant No. 104529). This work is also supported by UKRI Future Leaders Fellowship (Grant MR/S035176/1). The authors would like to thank the WMG centre of HVM Catapult and WMG, University of Warwick, UK, for providing the necessary infrastructure for conducting this study. WMG hosts one of the seven centres that together comprise the High-Value Manufacturing Catapult in the UK.

Definitions/Abbreviations

ADAS	Advanced Driving Assistance System
ADS	Automated Driving System
CF	Causal Factor
ECU	Electronic Control Unit
ETA	Event Tree Analysis
FMEA	Failure Mode and Effect Analysis
FTA	Fault Tree Analysis
HAZOP	Hazard and Operability Study
HBT	Hazard-based Testing
HMI	Human-machine Interface
OD	Operational Domain
ODD	Operational Design Domain
RBT	Requirement-based Testing
SDL	Scenario Description Language
STPA	System Theoretic Process Analysis
UCA	Unsafe Control Action
VUT	Vehicle under Test