2023-01-0574

An ODD-Based Scalable Assurance Framework for Automated Driving Systems

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

Due to the increasing complexities, the safety assurances for Automated Driving Systems (ADSs) and Advanced Driver Assistance Systems (ADASs) pose challenges. Recent development within the industry and academia suggests a scenario-based approach underpinned by the system's Operational Design Domain (ODD) for its safety assurance. In such framework, the ODD defines the safe operating boundary, whereas the scenarios set out individual test conditions. To assess the behavior of the system, a critical element for road safety is the ability to respect the rules of the road. This paper joins together ODDs, scenarios, and rules of the road to form a scalable ODD-based safety assurance framework. The backbone of the framework contains a coherent and common taxonomy to describe the ODDs and behavior library, the scenario tagging structure from the ASAM OpenLABEL standard has been used in the example use case.

The workflow utilizes the system's ODD and behavior library as input to perform filtering and matching activities over a set of testing scenarios and the rules of the road library. Firstly, the ODD and behavior input are used to filter the applicable scenarios within the initial scenario set. At the same time the ODD and behavior input can also be used to filter the applicable rules within the rules of the road library. By further utilizing the ODD and behavior tags covered by the applicable scenarios, and applying them to the rules of the road library, the applicable scenarios related rules can be identified; similarly using the ODD and behavior tags covered by the applicable rules, and applying them to the initial scenario set, the applicable rules related scenarios can be obtained. Such process allows the most relevant rules and scenarios to be used when testing a target system. Furthermore, by comparing the applicable scenarios related rules with the applicable rules, and comparing the applicable rules related scenarios with the applicable scenarios, one can gain an understanding on the effectiveness and the efficiency of the test. When combined with the wider scenario-based evaluation criteria, the framework illustrated within this paper provides a novel and effective way to conduct and evaluate tests.

Introduction

The recent advancement in ADS and ADAS technologies is driven by the many benefits [1][2][3][4]. However, due to the complexity of the system, their safety assurance poses challenges. Traditional approach for testing the systems utilizes a distance-based approach, however for ADSs they will need to be driven 11 billion miles to prove that they are 20% better than human drivers, making it not a feasible option [5]. The limitation of distance-based evaluation approach has shifted the industry and academia towards scenario-based testing approach. Scenario-based testing focuses on the quality of the miles driven rather than the quantity, purposely defined scenarios are used to test the interested aspects of a system [6][7]. Along the scenario-based testing workflow, the information carried by the scenario artefact is created, formatted, stored, executed, and analyzed.

Furthermore, the lack of consideration of the system's ODD is another drawback of the distance-based approach, ODD defines the operating conditions within which the system is designed to operate, being able to operate safely within a system's ODD is crucial. At a high-level, ODD contains the scenery (such as road, junctions, road structures), the environmental conditions (such as rainfall, snowfall, wind), and the dynamic conditions (such as types of agents, the maximum designated speed of the system) [8]. If a system designed for urban usage is safely driven for 11 billion miles in a rural environment, this is still not sufficient to prove they are safe for the urban use case. Along the V-model for system development, the ODD requirement for the system is usually introduced at the initial design stage (at the top left of the V shape), such requirements will be carried through the whole development and testing cycle. On the other hand, the role of scenarios is also embedded throughout the V cycle, different scenario abstraction level (from functional scenarios to concrete scenarios [9][10]) covers different phases. Based on the domain element composition, it is recognized that a scenario contains the ODD elements (road topology, traffic infrastructure, weather conditions etc.), as well as the behavioral elements (maneuvers or actions).

From a safety assurance point of view, one of the key criteria for the system testing is the compliance with the rules of the road. Rules of the road are developed and maintained by the regulatory authorities of a relevant region and are applicable to all road users operating within that region, including ADSs and ADASs. The 'digitization' of the rules of the road is currently gaining strong interest across the industry, academia, and government authorities. The goal for such 'digitization' is to make them directly applicable and interpretable to ADSs and ADASs. At a high-level, the rules of the road can also be broken down into ODD aspect (e.g., 'when approaching a pedestrian crossing') and behavioral aspect (e.g., 'stop if a pedestrian is crossing').

This paper details a framework which links the rules of the road with scenario-based testing by using the common denominators between the two, which are ODD and behavior. Such linkage can then be used to effectively search for the most relevant rules and scenarios for testing, and to evaluate the effectiveness of the underlying scenario set.

Related Work

A high-level scenario-based testing workflow along with its key components has been previously published, as illustrated in Figure 1 [7]. At the bottom row it contains three major building blocks - '*Scenario*', '*Environment*', and '*Certification/Safety Evidence & Arguments*'. Each of the building blocks is then further mapped to individual components on the workflow.

'*Scenario*' covers the creation of the scenarios (*'Create'*), the format for their description (*'Format'*), and the database storage of the generated scenarios (*'Store'*). '*Environment*' covers the distribution of scenarios across various execution environments (*'Plan'*), and the execution of them (*'Execute'*). '*Certification/ Safety Evidence & Argument*' covers the analysis of the execution results (*'Analyze'*), and consequently the safety evaluation results (*'Decide'*).

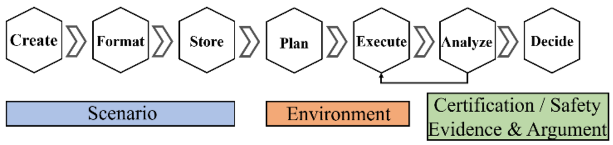

Figure 1. Key components along a high-level scenario-based testing workflow [7]

The scenario creation can be both data-based [11][12] or knowledge-based [13][14][7], upon the creation, the scenario content needs to be represented in a common and adequate format, such as [6][15][16][17]. A prerequisite of the scenario creation and formatting is a common understanding of the term *'scenario'* itself. Across the industry and academia, several efforts have contributed directly or indirectly towards the definition of scenario. Ulbrich et al. defines scenarios as the temporal development between scenes. Furthermore, a scene contains the environment including the scenery and dynamic elements, as well as self-representations of actors and observers [18]. In the two-level scenario description language from Zhang et al. [6], the scenario domain model contains the corresponding environmental conditions, dynamic behaviors, and the scenery [19]. In addition, the temporal nature is also contained within the language. The Pegasus project [20] also defines a mental model for describing elements within a scenario, which consists of road description, traffic infrastructure, temporal infrastructure changes, dynamic elements, environmental conditions, and connectivity. Such mental model can be directly mapped to the scenery, the environmental, and the behavioral aspects. The ASAM OpenSCENARIO 1.x [21] and OpenDRIVE [22] are two xml-based formats which have been widely used for scenario execution when combined [23]. OpenSCENARIO 1.x describes the dynamic entities, their behaviors and the environmental conditions, whereas OpenDRIVE describes the scenery elements such as road layout and additional objects. It can be seen that synergy and convergence around the key elements of scenarios are forming.

On the other hand, there are increasing emphasis and recognition of the role of ODD within the development and testing processes of ADSs. A widely used definition of ODD was introduced in SAE J3016 as follow [24]:

"Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics"

The BSI PAS 1883 [8] introduced a taxonomy which contains a standardized set of ODD attributes covering scenery, environmental conditions, and dynamic elements. In addition, the common units and categorizations for the attributes, where applicable, are also stated. The SAE AVSC00002202004 [25] also presented a conceptual framework and lexicon for defining ODDs, the lexicon includes key attributes covering environmental conditions, road surface conditions, roadway infrastructure, operational constraints, road users, roadside objects and connectivity. On-going effort is also seen in the ASAM OpenODD [26] project and the ISO 34503 [27] standard on ODD format and taxonomy. The ASAM OpenLABEL standard [28] also utilizes an ODD and behavior based model for scenario tagging and organization. It can be seen that, based on the review of ODD-related standards, synergy across the industry is also forming to address the question on what an ODD is and what attributes should an ODD contains.

From the rules of the road perspective, being able to comply with the rules of the road is a mandatory requirement for all road users, therefore it also applies to ADS and ADAS applications. Take the UK highway code [29] as an example, it is a set of information, advice, guidance and mandatory rules for road users in the UK with the objective of promoting safety. The currently applicable road users including drivers, motorcyclists, cyclists, animal riders and pedestrians. Initiatives across the industry, academia and regulatory bodies have been seen which focus on adapting and codifying the existing highway code rules into a set of unambiguous, machine interpretable rules which can be directly applicable for ADS and ADAS applications. This set of digitized rules can then form part of the testing oracle for the safety evaluation process of system.

Although fundamentally differ by their nature and definitions, there exist strong commonalities in terms of domain model composition between scenarios, ODDs, and rules of the road, as illustrated in Figure 2. Furthermore, each of them has distinctive roles within the evaluation framework for ADSs and ADASs, the main section of this paper will explore such commonalities and incorporate them into an assurance framework.

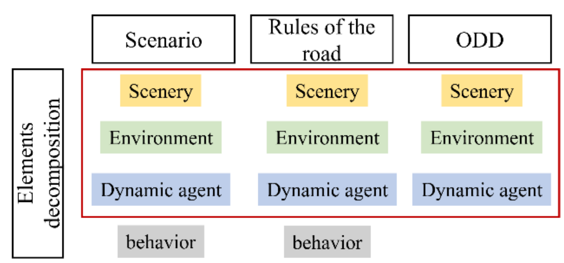

Figure 2. Commonalities of elements decomposition between scenarios, ODDs, and DHC

Behavior and Behavior Library

For ADSs and ADASs, behaviors include both moving behaviors, as well as communicating behaviors. Several standards and publications contain similar categorization of the behavioral elements. The ASAM OpenXOntology[30] presented four different ways to categorize behaviors, one of the categorizations divides the behaviors into communicating activities and moving activities. Communicating activity is defined as the traffic participants giving visual, or acoustic signals in order to relay its intentions to other traffic participants, whereas moving activity will result into a change in location of the subject traffic participant. Similar conclusion is also seen in ASAM OpenLABEL [28] within the scenario tagging chapter, within which the behaviors are categorized into communication and motion. The OpenLABEL 1.0 standard is further adopted as the underlying structure for the Safety Pool™ scenario database[31], which as of now is the world's largest scenario database funded by the UK government. Furthermore, the two-level scenario description language (SDL) [6]also utilizes similar set of behaviors for describing scenarios. As a result, the moving behaviors broadly contains *cut-in, cut-out, drive, lane change right, lane change left, reverse, turn left, turn right, cross, move away, move towards, stop, run, slide, walk.* The last three are unique to pedestrians, whereas the first eight are unique for vehicles, the four behaviors in the middle are common between pedestrians and vehicles. Moving behaviors can also be categorized by their relations to other entities [30][6], and split into relative behaviors (activities that require more than one participant to define) and absolute behaviors (single-participant activities). Please note that the behavior set cannot be exhaustive and will be evolving, in addition, organizations might want to add further specialized details of behaviors. The extensible nature of the underlying domain model allows such extension, both the OpenLABEL and OpenXOntology document the details on adding custom ontology extensions. Table 1 illustrates the categorization of the moving behaviors based on relative behaviors and absolute behaviors.

Table 1. Categorizing behaviors into relative behaviors and absolute behaviors

Absolute Behaviors	Relative Behaviors
Drive	Cut-in
Lane change right	Cut-out
Lane change left	Move away
Turn left	Move towards
Turn right	Cross
Stop	Overtake
Reverse	
Run	
Slide	
Walk	

While the moving behaviors have gained strong emphasize across the industry, there has been less focus on communicating behaviors. In addition to the visual and acoustic communicating behaviors as stated in the definition of communicating activity in the ASAM OpenXOntology, there is also increasing focus on the V2X (vehicle-to-everything) related communicating behaviors [32]. The current identified communicating behaviors, based on the ASAM OpenXOntology [30], the ASAM OpenLABEL [28], and the two-level SDL [6], include *flash headlight, signal emergency, signal hazard, signal left, signal right, signal slowing, sound horn, wave, unicast, broadcast, multicast*. The first eight are conventional road users communicating/signaling methods, the last three belong to connectivity-based (vehicle-to-vehicle, vehicle-to-infrastructure etc.) communicating method.

This list of behaviors outlines the generic behavioral categories a road user could perform, however for individual systems there could be additional behaviors that are specific for certain use cases, for example *rotate* for forklift. In such case, an extension to the domain model can be made to tailor towards such unique use case. The collection of the all the available behaviors within the domain model, baseline behaviors plus any extensions, forms the **behavior library**.

ODD Attributes

As mentioned earlier, ODD defines the operating conditions under which a system is designed to function. To go into the next detail level, Figure 3 illustrates a class hierarchy of ODD attributes. It can be seen that scenery further contains zones, drivable area, junctions, special structure, fixed road structures, and temporary road structure. Zones include geographical areas with special road layout, regulations, or ambient conditions, for example school zone, interference zone etc. Drivable area contains different types of roads and shared spaces, together with their geometries, lane specification, surface conditions. Junctions contains different types of roundabout and intersections. Special structures, fixed road structures, and temporary structures include additional relevant features that can be placed on top of the drivable area to form part of the scenery, such as pedestrian crossing, buildings, vegetation, automated barriers etc. Within environmental conditions, weather further contains different types of rainfall, snowfall, and wind. particulates induced conditions such as fog and mist, sand and dust are included within Particulates. Illumination contains the lighting conditions, it includes both natural and artificial illuminations. Connectivity contains the types of positioning and communicating methods. Within the dynamic elements, the macroscopic traffic characteristics are included, together with the subject vehicle designed speed. For more details, please refer to the ODD taxonomy for ADSs [8].

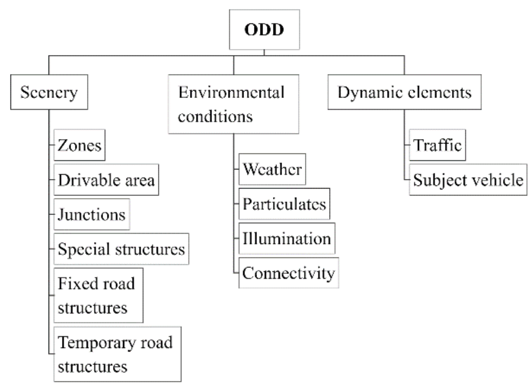

Figure 3. High-level class hierarchy of ODD attributes

Rules of the Road

An example of the rules of the road is the current UK highway code [29]. While certain rules might not be directly relevant to ADSs and ADASs, such as rules related to *fitness to drive* (rules 90 to 94), *alcohol and drugs* (rules 95 to 96); it has been identified that large portion of the rules across the whole rule sets are directly applicable to ADSs and ADASs, such as rules related to *using the road* (rules 159 to 203), *driving in adverse weather conditions* (rules 226 to 237).

Furthermore, this set of directly applicable rules can be broken down into ODD elements and behavioral elements. Take the rule 191 (pedestrian crossing) from the UK highway code as an example, it states that:

*"You **MUST NOT** park on a crossing or in the area covered by the zig-zag lines. You **MUST NOT** overtake the moving vehicle nearest the crossing or the vehicle nearest the crossing which has stopped to give way to pedestrians"*

The diagram representing this pedestrian crossing rule is displayed in Figure 4 [29]. It can be seen that the associated ODD elements are *road, lane, pedestrian crossing, zig-zag lines, warning light, pavement*, whereas the behavior elements are *park, overtake, give way, cross (walk)*. Through a mapping process, all the ODD and behavior elements extracted from the highway code rules can be represented within the behavior library and the ODD taxonomy introduced in the previous two sections.

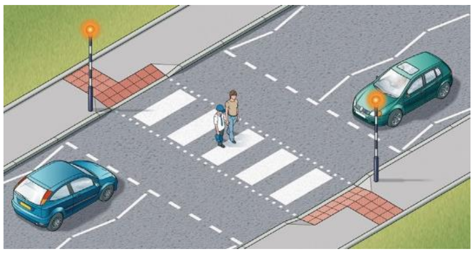

Figure 4. Diagram illustrating the UK highway code rule 191 on pedestrian crossing [29]

ODD Based Assurance Framework

In this section, a filtering mechanism will be introduced which allows for the identification of the relevant traffic rules based on any ODD and behavior input. Subsequently, a similar mechanism which allows for the identification of the relevant scenarios based on the same ODD and behavior input is also introduced. Lastly, it will unify different sets of results obtained from the previous two steps, and cross-check and evaluate the relevance between the filtered scenarios and rules.

Rules of the Road & Scenario Tagging Based on ODD and Behavior

Each of the traffic rules that is directly relevant to ADSs and ADASs can be decomposed into the corresponding ODD elements and behavior elements. On-going research activities (which WMG, the University of Warwick is part of) have been carried out to create a codified behavioral ruleset for highly automated vehicles based on the UK highway code. It has been found that around 50% -70% of the current highway code rules are fully or partially applicable to ADSs, which amounts to over 200 rules. By iterating through the whole rule set and assigning ODD & behavioral tags to each of the rules, a rule library which contains all the tags can be constructed. Upon the completion of such rule library, user could input any desired ODD and behavior elements that belong to the common underlying domain model, if there are satisfying conditions met based on such input, the corresponding rules will be return as the results. Figure 5 illustrates the end-to-end process of creating a rules library with ODD and behavior tags, as well as how it can be used for searching and querying. As can be seen that once the relevant ODD and behavior tags are identified for each of the rules, the tagging and querying processes are similar to the ASAM OpenLABEL scenario tagging process at the conceptual level. The ASAM OpenLABEL scenario tagging process has been implemented within the Safety Pool™ scenario database [31], and documented within the standard. However, using similar mechanism for tagging the rules of the road has not been done prior to this paper. The main difference is that rules of the road instead of scenarios are being tagged in this case. Similarly, following the ASAM OpenLABEL scenario tagging process, individual scenarios can also be associated with their ODD and behavior components.

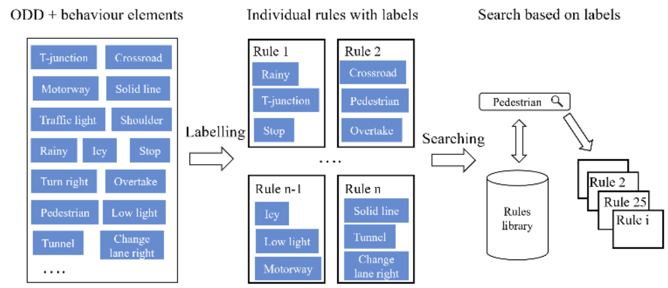

Figure 5. The tagging and querying process of the traffic rule

Underneath the tagging processes for both the rules and scenarios, it utilizes a json schema for organizing the tags, as shown in Listing 1. Within the ***"ontologies"*** section, user can reference one or multiple ontology resources using their unique identifiers, this will create the tagging universe where all the tags belong. In addition, user can also define a small subset where the tagging activity is performed over the whole ontologies. Ontology elements that fall outside such tagging "boundary" will not be considered during the query process. Within the ***'tags'*** section, the relevant tags for individual rules or scenarios are described. In this example, the tag ID '***0***' indicates that *broken line* is present within the scenario or rule alongside any attributes listed in the '***tag_data***' section, the class *broken line* belongs to the reference ontology 1 which is included within the '***ontologies'*** section. Within the ***'metadata'*** section, additional administrative information such as the SAE levels of driving automation or the authorship can be defined.

Listing 1. Top-level json schema for the OpenLABEL scenario tagging

```
{
 "openlabel":{
   "tags": {
     "0":{
       "type": "LaneMarkingBrokenLine",
       "ontology_uid": "1",
       "tag_data":{…}
     },
     …
   },
   "metadata": {…},
   "ontologies": {…},
 }
```

To illustrate the tagging semantics, an example domain hierarchy of *'ODD --> Scenery --> Junction --> Roundabout'* can be used, it means '*Scenery'* is a subclass of '*ODD'*, '*Junction'* is a subclass of '*Scenery'*, and '*Roundabout'* is a subclass of '*Junction'*. User can tag elements at any abstraction levels. If *'Junction'* is tagged using this example hierarchy, the corresponding rule or scenario will only be returned if *'ODD'*, '*Scenery'*, or *'Junction'* is used for querying. It will not be returned if *"Roundabout"* is used for querying, since

'Junction' being present does not necessary indicate *'Roundabout'* being present. Therefore, the inheritance of the tags is one-directional, children nodes inherit parent nodes.

Scenarios and Rules Matching Based on ODD and Behaviors

Previous sections provide the background on how the scenarios and the rules of the road can be decomposed and linked with the same set of domain classes (i.e., ODD & behavior elements). This section will illustrate how to combine the two to create a novel approach for scenario-based testing and to feed into the evaluation process. As can be seen in Figure 6, the workflow starts with a set of *ODD and behavior input*, a set of testing scenarios, and the rules of the road, as indicated by the three elements at the left-hand side of Figure 6. The *ODD and behavior input* in this case can be bounded with a particular system. Upon receiving the input, such ODD and behavior attributes can then be used as matching criteria within a *scenario repository* to filter out the *applicable scenarios*. At the same time such *ODD and behavior input* can also be used to filter out the *applicable rules* within the *rules of the road library*. Please note that the *applicable scenarios* might or might not be able to cover all the *applicable rules* from their ODD and behavior tags as it is dependent on the diversity of the initial scenario library.

Based on the ODD and behavior tags associated with each *applicable scenario*, the relevant rules related to each *applicable scenario* can be determined, these form the *applicable scenarios related rules*. Similarly, using the associated ODD and behavior tags with the *applicable rules*, the relevant scenarios related to each *applicable rule* can be determined, these form the *applicable rules related scenarios*. This bi-directional process allows the matching process between the scenarios and the rules of the road based on ODD and behavior tags, which facilitates the testing efficiency and evaluation process. By comparing how many *applicable rules* are covered by the *applicable scenarios related rules*, one can form a general understanding on the effectiveness of the scenario set from the rules of the road perspective. Similarly, by comparing how many *applicable scenarios* are covered by *the applicable rules related scenarios*, one can form an understanding on the effectiveness of the rules of the road as an evaluation criterion on this particular set of scenarios and system specification. Such analysis, of course, needs to be considered under the wider scenario-based testing evaluation process.

Figure 6. Scenarios and rules of the road matching based on ODD and behavior input

Example

To illustrate the concept, this section will instantiate the framework based on the Automated Lane Keeping Systems (ALKS)[33] use case and an example SUT. For this example, a hypothetical ODD and behavior library for the system will be stated as the initial input; a set of ALKS scenarios will also be introduced, together with a set of rules of the road for evaluation. In the end the rules of the road coverage analysis can be obtained.

System Input

To provide the input into the framework from the SUT (as shown at the top box in Figure 6), the capabilities of the SUT are firstly defined in this section. The capabilities are divided into ODD specification, and behavior library. Using the two-level ODD description format, at the natural language level (i.e., level 1) [34][35][27], the example ODD for a hypothetical SUT can be written as Figure 7.

```
Include: PAS 1883:2020 Operational Design Domain (ODD) taxonomy
for an automated driving system (ADS) - Specification [2020,
https://www.bsigroup.com/en-GB/CAV/pas-1883/]
Base state: Permissive
Extension: N/A

#Composition statements
Unsuitable Drivable area type is [Minor roads]
Suitable Number of lanes is [greater than 2]
Suitable Lane dimension is [greater than 3.7]
Suitable lane type is [Traffic lane]
Suitable Direction of travel is [Left-hand drive]
Suitable Curve is [less than 1/500 m]
Unsuitable Transverse plane is [Undivided, Barriers on edge]
Suitable Drivable area surface type is [Asphalt, Concrete]
Unsuitable Drivable area signs is [Part-time signs]
Cond_1 Conditional Drivable area type is [Motorway]

#Conditional statements
Cond_1 Unsuitable Weather for [Motorway] is rainfall
```

Figure 7. Example ODD for the SUT

It can be seen that the SUT can handle road with traffic lanes and number of lanes greater than 2, with lane width greater than 3.7 meters. The SUT is also a left-hand driving system which can mainly handle straight road geometry with road surface made of asphalt and concrete. It cannot handle undivided roads (bi-directional roads), roads with barrier on edge, and roads with part-time traffic signs. The capability for handling motorway is only valid when the weather is not raining. Based on this ODD description, the system can handle any traffic participant types as defined in the ODD taxonomy, these include pedestrians, vehicles, cyclists.

For the behavior library, the system can perform all the vehicle related behaviors listed in Table 1. In addition, it can handle all behaviors performed by other traffic participant types, such as walking or running towards by a pedestrian, lane changing right cut-in by a vehicle.

Rules of the Road Filtering

The rules of the road filtering is performed based on the ODD and behavior input from the system over a rules of the road library. The ideal rules of the road library would contain all the ADS/ADAS applicable rules. For this example, it is assumed that there are only six rules within the rules of the road library. The six example rules are referenced to the UK highway code [29], they are listed in Table 2.

When mapped against the system ODD and behavior input, Rule 163 and Rule 189 are excluded. Rule 163 is excluded due to its application is only for bi-directional undivided road, i.e., give way to oncoming vehicles. Whereas in the ODD description of the system, the unsuitable drivable area transverse plan is undivided. Rule 189 is excluded due to the road curvature associated with a mini roundabout exceed the curvature stated in the system's ODD. Therefore, upon performing the rules of the rule filtering, four out of the six rules are applicable to the system.

Table 2. Example rules of the road library

Rules	Description (only partial description is referenced)
Rule 162	Before overtaking you should make sure the road is sufficiently clear ahead
Rule 163	Overtake only when it is safe and legal to do so. You should give way to oncoming vehicles before passing parked vehicles or other obstructions on your side of the road
Rule 189	At double mini-roundabouts treat each roundabout separately and give way to traffic from the right.
Rule 198	Give way to anyone still crossing after the signal has changed to green. This advice applies to all crossings.
Rule 206	Drive carefully and slowly when approaching pedestrians who have started to cross the road ahead of you.
Rule 261	You MUST NOT exceed the maximum speed limit for the road and for your vehicle

Scenario Filtering

The scenario filtering based on the system's ODD and behavior input can also be carried out in a similar manner. Assuming there are four scenarios in the whole scenario library. They are S1 - *Cut-out multiple blocking targets*, S2 - *Crossing Pedestrian*, S3 - *Follow Lead Vehicle Emergency Brake*, and S4 - *Turning around a corner with pedestrian crossing*. As shown in Figure 8.

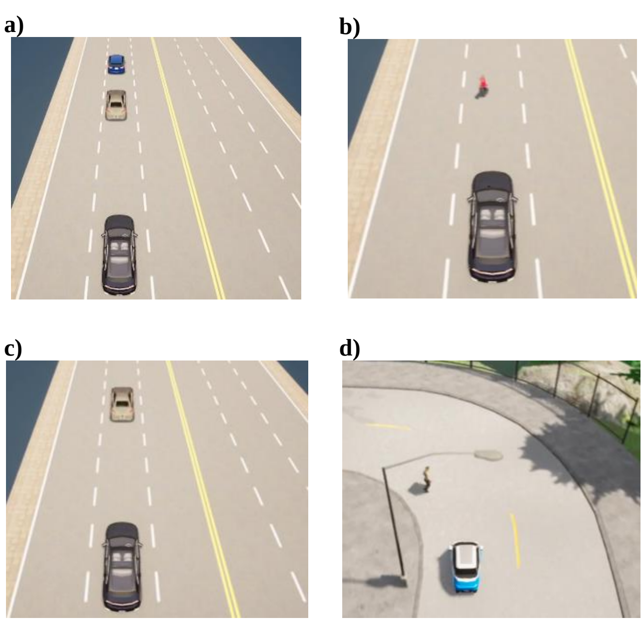

Figure 8. Example scenarios within the scenario library

Figure 8a (S1) illustrates a scenario where the system (the bottom vehicle) is following two front vehicles, the vehicle at the front stops, causing the vehicle in the middle to cut-out and change lane left. Figure 8b (S2) illustrates a scenario where there is a pedestrian crossing the road in front of the system (the bottom vehicle) while the system is driving on a motorway. Figure 8c (S3) illustrates a scenario where the system (the bottom vehicle) is following a lead vehicle (the top vehicle), and the lead vehicle performs an emergency brake. Figure 8d (S4) illustrates a scenario where the system (the bottom vehicle) is navigating through a corner when a pedestrian is crossing the road in front of the system.

Based on the same ODD and behavior input from the system, S4 is excluded due to the presence of a curving road with curvature exceeds the system's ODD. In this case, three out of the four scenarios are applicable for the system.

Scenarios and Rules Matching

Please note that at the operational level, both the rules of the road filtering and the scenario filtering are done systematically utilizing the ASAM OpenLABEL tagging files.

So far, from the rules of the road library, the applicable rules are Rule 162, Rule 198, Rule 206, and Rule 261. From the scenario library, the applicable scenarios are S1, S2, S3.

By comparing the ODD and behavior tags associated with the applicable rules and applicable scenarios, it can be seen that among the applicable rules, Rule 198 is not covered by the applicable scenarios. Rule 162 (overtaking) is covered by S1 and S3; Rule 206 (pedestrian crossing) is covered by S2; Rule 261 (maximum speed) is covered by all three scenarios. On the other hand, Rule 198 requires the presence of a traffic signal, from an operational perspective, this would mean that an ODD tag for the traffic signal needs to be present among the scenario tagging files for the scenarios. Similarly, among the applicable scenarios, all of the scenarios are covered by the applicable rules. S1 is covered by Rule 162 and Rule 261; S2 is covered by Rule 261, Rule 206 and Rule 261; S3 is covered by Rule 162, Rule 261; and S4 is covered by Rule 261, Rule 206 and Rule 261.

It is interesting to see that with the same sets of rules and scenarios, this bi-directional matching process yields different perspectives. In plain language, from a rules of the road point of view, the scenario set cannot fully explore all the applicable rules; on the other hand, all the applicable scenarios can be covered by some rules within the applicable rule set for evaluation.

Logical Flow

At the implementation level, process 1 below details the logical flows for the rules of the road filtering, for the scenario filtering, and for the scenarios and rules matching. Both the rules of the road filtering and the scenario filtering share the same logic, whereas scenarios and rules matching is different, this difference is indicated by the stage input. For rules of the road filtering and scenario filtering, the set of tagged files A represents the input scenario repository or the rules library. For the scenarios and rules matching, the applicable scenarios and applicable rules are represented by tagged files A and tagged files B.

Process 1: ODD_Behavior_Matching

Input: stage, a set of tagged files A, a set of tagged files B (optional), ODD & behavior input

Output: coverage analysis (as input into evaluation framework)

```
if stage is rules of the road filter or scenario filtering:
    for file in tagged files A:
        if file associated tags in ODD & behavior input:
            append applicable files to include file
    return applicable files
else if stage is scenarios and rules matching:
    for file a in tagged files A:
        for file b in tagged files B:
            if file b associated tags in file a associated tags:
                append applicable files a to include file a
    return coverage analysis of tagged files A by applicable
    files a
```

The ODD_Behavior_Matching process illustrated above can then be fitted into the overall workflow shown in Figure 9. It can be seen that the inputs are scenarios repository, SUT ODD & behavior input, and rules of the road library. By applying ODD_Behavior_Matching with stage rules of the road filtering or scenario filtering, the applicable scenarios and applicable rules can be obtained. And by applying ODD_Behavior_Matching with stage scenarios and rules matching twice, bi-directionally between applicable scenarios and applicable rules, the coverage analysis can be obtained, which can be further fed into the overall scenario-based evaluation process.

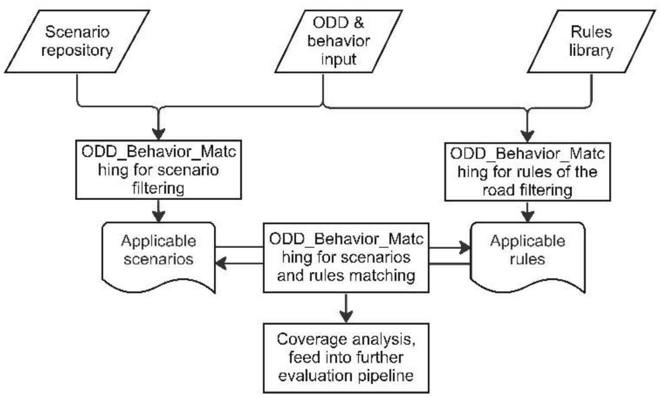

Figure 9. Logical flows of the process

Conclusions

In this paper, a novel approach is presented for evaluating the rules of the road coverage by a pre-defined scenario repository. Underneath the framework is a common ODD and behavior-based domain model. Firstly, the association of individual scenario and rules of the road with their relevant ODD and behavior elements is carried out, this forms the searchable scenario library and rules of the road library. Given any ODD and behavior input from a specific system, the applicable scenarios and rules can then be determined. By further identifying and matching the ODD and behavior elements found within this set of applicable scenarios and rules, another set of related rules and related scenarios can be determined. Such analysis flow facilitates the evaluation of the test efficiency and effectiveness, and provides a scalable way to identify the most relevant scenarios and rules of the road for testing a specific system. The approach has also been input into the EU ADS act on procedure and technical specifications for the type-approval of the automate driving system (ADS) of fully automated motor vehicles [36]. Future work includes the implementation of such workflow across various SAE level systems, ranging from ALKS to parking systems.

References

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles : opportunities , barriers and policy recommendations," *Transp. Res. Part A*, vol. 77, pp. 167–181, 2015.

[2] S. Le Vine, X. Liu, F. Zheng, and J. Polak, "Automated cars : Queue discharge at signalized intersections with ' Assured-Clear-Distance-Ahead ' driving strategies," *Transp. Res. Part C Emerg. Technol.*, vol. 62, pp. 35–54, 2016.

[3] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles , using agent-based model scenarios," *Transp. Res. Part C Emerg. Technol.*, vol. 40, pp. 1–13, 2014.

[4] N. Balfe, S. Sharples, and J. R. Wilson, "Impact of automation : Measurement of performance , workload and behaviour in a complex control environment," *Appl. Ergon.*, vol. 47, pp. 52–64, 2015.

[5] 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 Pr.*, vol. 94, pp. 182–193, 2016.

[6] X. Zhang, S. Khastgir, and P. Jennings, "Scenario Description Language for Automated Driving Systems: A Two Level Abstraction Approach," *IEEE Int. Conf. Syst. Man Cybern.*, 2020.

[7] X. Zhang, S. Khastgir, H. Asgari, and P. Jennings, "Test Framework for Automatic Test Case Generation and Execution Aimed at Developing Trustworthy AVs from Both Verifiability and Certifiability Aspects," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2021-Septe, pp. 312–319, 2021, doi: 10.1109/ITSC48978.2021.9564542.

[8] "Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification," *The British Standards Institution, BSI PAS 1883*. 2020.

[9] M. Scholtes *et al.*, "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment," *IEEE Access*, vol. 9, pp. 59131–59147, 2021, doi: 10.1109/ACCESS.2021.3072739.

[10] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," *arXiv*, vol. 2018-June, no. Iv, pp. 1821–1827, 2018.

[11] E. Esenturk, A. G. Wallace, S. Khastgir, and P. Jennings, "Identification of Traffic Accident Patterns via Cluster Analysis and Test Scenario Development for Autonomous Vehicles," *IEEE Access*, vol. 10, pp. 6660–6675, 2022, doi: 10.1109/ACCESS.2021.3140052.

[12] E. Esenturk, S. Khastgir, A. Wallace, and P. Jennings, "Analyzing real-world accidents for test scenario generation for automated vehicles," *IEEE Intell. Veh. Symp. Proc.*, vol. 2021-July, no. Iv, pp. 288–295, 2021, doi: 10.1109/IV48863.2021.9576007.

[13] 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.

[14] T. Menzel, G. Bagschik, L. Isensee, A. Schomburg, and M. Maurer, "From functional to logical scenarios: Detailing a

keyword-based scenario description for execution in a simulation environment," *IEEE Intell. Veh. Symp. Proc.*, vol. 2019-June, pp. 2383–2390, 2019.
[15] F. Bock, C. Sippl, A. Heinzz, C. Lauerz, and R. German, "Advantageous usage of textual domain-specific languages for scenario-driven development of automated driving functions," *SysCon 2019 - 13th Annu. IEEE Int. Syst. Conf. Proc.*, 2019.
[16] D. J. Fremont, X. Yue, T. Dreossi, A. L. Sangiovanni-Vincentelli, S. Ghosh, and S. A. Seshia, "Scenic: A language for scenario specification and scene generation," *Proc. ACM SIGPLAN Conf. Program. Lang. Des. Implement.*, pp. 63–78, 2019, doi: 10.1145/3314221.3314633.
[17] ASAM e. V, "Openscenario homepage." https://www.asam.net/standards/detail/openscenario/ (accessed Jan. 10, 2023).
[18] 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-Octob, pp. 982–988, 2015.
[19] A. Bruto Da Costa, P. Irvine, X. Zhang, S. Khastgir, and P. Jennings, "Writing Accessible and Correct Test Scenarios for Automated Driving Systems," pp. 1348–1355, 2022, [Online]. Available: http://wrap.warwick.ac.uk/167042
[20] PEGASUS Project, "Pegasus Method," no. 01, 2018, [Online]. Available: https://www.pegasusprojekt.de/en/pegasus-method%0Ahttps://www.pegasusprojekt.de/en/
[21] "ASAM OpenSCENARIO 1.1.1", [Online]. Available: https://www.asam.net/standards/detail/openscenario/
[22] "ASAM OpenDRIVE 1.7", [Online]. Available: https://www.asam.net/standards/detail/opendrive/
[23] A. Anastasio, P. Irvine, X. Zhang, S. Khastgir, and P. Jennings, "Translating Automated Vehicle Test Scenario Specifications Between Scenario Languages : Learnings and Challenges," pp. 14–16, 2022.
[24] SAE International, "SAE J3016 - Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," 2021.
[25] SAE ITC, "AVSC Best Practice for Describing an Operational Design Domain: Conceptual Framework and Lexicon," 2020.
[26] ASAM., "OpenODD Project Proposal," 2020.
[27] ISO, "ISO/DIS 34503 - Road Vehicles - Test scenarios for automated driving systems - Taxonomy for operational design domain," 2022.
[28] "ASAM OpenLABEL1.0", [Online]. Available: https://www.asam.net/standards/detail/openlabel/
[29] "The Highway Code." https://www.gov.uk/guidance/the-highway-code
[30] "ASAM OpenXOntology", [Online]. Available: https://www.asam.net/project-detail/asam-openxontology
[31] "Safety Pool Scenario Database." https://www.safetypool.ai/
[32] P. Irvine, P. Baker, A. Bruto da Costa, X. Zhang, S. Khastgir, and P. Jennings, "Vehicle-to-Everything (V2X) in Scenarios: Extending Scenario Description Language for Connected Vehicle Scenario Descriptions (Accepted)," 2022.
[33] UNECE, "R157," vol. 1958, no. March 1958, 2020.
[34] P. Irvine, X. Zhang, S. Khastgir, E. Schwalb, and P. Jennings, "A Two-Level Abstraction ODD Definition Language : Part I," pp. 17–20, 2021.
[35] E. Schwalb, P. Irvine, X. Zhang, S. Khastgir, and P. Jennings, "A Two-Level Abstraction ODD Definition Language : Part II," pp. 17–20, 2021.
[36] "Automated cars – technical specifications," 2022. https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/12152-Automated-cars-technical-specifications (accessed Jan. 10, 2023).

Contact Information

Mailing address: WMG, University of Warwick, Coventry, United Kingdom. Email address: Jason.Zhang@warwick.ac.uk

Acknowledgments

The work presented in this paper has been supported by UKRI Future Leaders Fellowship (Grant MR/S035176/1), Department of Transport, UK, and Transport Canada. The authors would like to thank the WMG center of HVM Catapult, and WMG, University of Warwick, UK for providing the necessary infrastructure for conducting this study.

Definitions/Abbreviations

ADS	Automated Driving System
ADAS	Advanced Driver Assistance System
ASAM	Association for Standardization of Automation and Measuring Systems
ODD	Operational Design Domain
SAE	Society of Automotive Engineers
SDL	Scenario Description Language
V2X	Vehicle-to-everything