Ontology-based Scenario Generation for Automated Driving Systems Verification and Validation using Rules of the Road

Antonio Anastasio Bruto da Costa, *Member, IEEE,*, Patrick Irvine, Xizhe Zhang, Siddartha Khastgir, *Member, IEEE,*, Paul Jennings

***Abstract*—The verification and validation (V&V) process for Automated Driving Systems (ADS) has undergone a significant transformation in defining the meaning of safety. Initially rooted in the quantity of miles driven, it has now shifted towards emphasizing the quality of test miles. These test miles must effectively capture the full spectrum of behaviours and operational design domains (ODD) of the ADS. To assess an ADS's compliance with specific rules or requirements, a connection must be established between the rules and the scenarios used for testing. In this paper, we propose a targeted scenario generation methodology aimed at testing ADS against formal rules. Our approach leverages ontologies to represent objects and their relationships in a scenario. The rules, are first formally specified, expressed as horn clauses. We then employ a rule transformation process, along with off-the-shelf reasoning tools, to generate corresponding scenarios. These generated scenarios may then utilized to test the ADS's adherence to the specified rules. To illustrate the effectiveness of our methodology, we present an application to example rules derived from both the UK Highway Code and the Vienna Conventions. By utilizing our approach, we enhance the precision and rigor of the verification and validation flow for ADS, ensuring improved safety measures during operation.**

***Index Terms*—scenario, scenario generation, ontologies, rules of the road, verification and validation, automated driving systems.**

I. INTRODUCTION

The on-road readiness of a human driver is assessed by examining their ability to adhere to a set of driving rules. These rules express protocols of driving behaviour on the road and also set an expectation of how other road actors behave and respond to the driver's actions. It is only when the driver is able to demonstrate that they adhere to these rules, that they are issued a license that certifies them to operate a vehicle on the road.

With ongoing developments in vehicle autonomy, actions of the human driver are being replaced by autonomous driving systems (ADSs). The Society of Automotive Engineers (SAE) defines six levels of automated driving, from no automation to full automation, replacing the human driver [1]. Autonomy is encouraged for its potential to reduce on-road accidents causing injuries and fatalities [2], while also improving overall driving characteristics [3]. It is widely accepted that realising these gains requires a philosophy of testing that is based in the quality of miles driven by the ADS, as opposed to the quantity of testing miles [4], [5], exposing the ADS to the myriad conditions it is expected to respond to. This is further asserted through various studies and the adoption of scenario-based testing as the basis of V&V approaches [6]–[11]. A *scenario* is a *'temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Action and events as well as goals & values may be specified to characterise this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time.'* [12].

The evaluation of ADSs in diverse scenarios is essential; however, it lacks a robust mechanism for precisely defining correct ADS behavior. An appealing criterion for evaluation is the existing set of rules of the road, originally designed for human understanding. Nevertheless, for an ADS to be deemed deployable within a specific domain, whether it be a city, region, or country, it must demonstrate strict adherence to the relevant rules of the road. To achieve this, formal codification of these rules is necessary to make them usable for specifying and evaluating ADS compliance with the designated rules.

The set of driving rules serves as a fundamental protocol for driving behavior. However, ensuring the completeness of these specifications presents a well-known challenge in the field of specification design. The question arises: How can we assess the sufficiency of these specifications, particularly in terms of intention? It is crucial to consider the possibility of interactions among active specification rules that could render the set of actions available to the ADS empty, leading to a situation known as a *deadlock*. This concept is a central concern in systems and protocol design and testing [13]–[15]. If the set of fundamental driving actions, governed by the driving rules, results in a deadlock, it indicates the need for augmenting the rules to enable the vehicle to make progress. Additionally, when dealing with situations stemming from driving rules, it becomes imperative to generate scenarios that can effectively test the ADS's adherence to these rules.

This paper contributes to addressing the aforementioned gaps in the following ways:

1) Methodology for Formalising Rules of the Road: We propose a comprehensive methodology for formalising rules of the road. To illustrate the practical application of this methodology, we utilise examples from the United Kingdom (UK) Highway Code [16] and the Vienna Convention [17]. Through these examples, we discuss

---

The authors are with WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom.

Emails: {antonio.bruto-da-costa, patrick.irvine, jason.zhang, s.khastgir.1, paul.jennings}@warwick.ac.uk

concrete challenges towards formalisation of rules and the actions needed to address them.

2) Scenario Generation for ADS Testing against Rules of the Road: We demonstrate how the results obtained from reasoning processes can be utilised to generate purpose-built scenarios for testing Automated Driving Systems (ADS). These scenarios are tailored to evaluate the ADS's performance and adherence to the specified rules, ensuring thorough verification and validation of its safety during real-world operations.

By addressing these key aspects, our work contributes to the advancement of verification and validation processes for ADS, enhancing their safety and reliability for deployment in various domains. In the subsequent sections, we delve into key aspects of our study. Section II delves into related research, while Section III provides an in-depth insight into our rule-set codification methodology. We then introduce our scenario generation framework originating from codified road rules in Section IV. To demonstrate practicality, Section V showcases the application of our approach to select rules from the U.K. Highway Code and the Vienna Conventions. Our observations, challenges, and identified gaps are deliberated upon in Section VI, followed by concluding remarks in Section VII.

II. RELATED WORK

The case for using smart-miles to assess the safety of ADSs, where the quality of distance driven is more important than quantity, has been widely accepted as the way forward [5], [11], [12], [18]. Hence, scenarios are now a cornerstone of the V&V lifecycle. Due to the real-valued nature of parameters representing key scenario elements (such as positions, distances, and lengths; velocity and acceleration; time at which a manoeuvre is executed; heading angles; weather parameters, and so on), and their value combinations, the theoretical number of scenarios for testing becomes infinite. Furthermore, the arrangement of events in a scenario and their timings adds to this complexity. Intelligent mechanisms to generate targeted scenarios is important to handle this difficulty.

In the past, there have been studies aimed at generating scenarios from accident databases [10], [19] or from standards such as the Automated Lane Keeping Standard (ALKS) [20] and others. A number of studies have proposed Ontologies for use with ADS testing and test management [21], [22]. An ontology is a structured representation of knowledge within a specific domain. It defines concepts, their relationships, and attributes, serving as a framework for organizing information. It also facilitates effective information processing, reasoning and manipulation. Ontologies have also been used to generate test cases for ADS to test autonomous functions [22], [23]. The authors in these studies [22], [23] use ontologies, together with combinatorial methods and machine learning to intelligently generate test vectors that cover critical autonomous functions. The work does not however discuss how the ontologies for the critical scenario would be themselves generated, only how the resulting structure is used to generate tests using the proposed combinatorial methods.

Other studies have proposed an ontology designed specifically for scenario definition [7]. The work presents an ontological definition of a scenario founded on object-oriented principles, with the classes, attributes, and properties represented as an ontology. The authors demonstrate that the ontology can be used to construct a detailed scenario description in an object-oriented fashion and can be translated into code using scenario description languages.

The Association for Standardisation of Automation and Measuring Systems (ASAM), as part of it array of standards has developed an ontology concept, ASAM OpenXOntology [24]. The ASAM OpenXOntology provides fundamental definitions, properties, and relations of concepts in ADS. The ontology is publicly available, and well documented. Such ontologies make it easy to have a single consistent definition for terms and concepts across scenario specification languages such OpenSCENARIO [25] and OpenDrive [26].

Recent endeavors have introduced a structured natural language framework for rules of the road, thus facilitating their expression [27]. The study emphasizes the significance of adopting a consistent and structured approach to expressing rules of the road, underlining the importance of well-defined rules. It advocates for seamlessly integrating these rules with testing scenarios to assess ADS capabilities.

However, as the complexity of driving scenarios continues to escalate, a pressing requirement has emerged to synchronize Automated Driving System (ADS) testing with assessments against established road regulations. The focal point of this challenge is to develop scenarios that precisely target specific rules of the road. This article addresses this challenge by employing ontologies and formally articulated rules of the road. Employing a consistent language to express road regulations (as exemplified in Ref. [27]) expedites their translation into an appropriate formal logical structure. This, in turn, enhances the process delineated in this article.

III. FORMALISING RULES OF THE ROAD

Presently, road rules are often penned in unstructured language, yielding incomplete specifications and ambiguous statements. For instance, the U.K. Highway Code's Rule 162 [16] includes phrases like *"sufficiently clear ahead"* and *"suitable gap"*, intended for human comprehension but unsuitable for ADS precision. These terms necessitate clear, objective definitions. Recent efforts propose a structured natural language framework [27] that could rectify this, fostering more consistent handling of such rules for improved ADS understanding.

The process of formalising these rules aims to reveal any incomplete specifications and clarify the semantics of rule terms, making them usable as the gold standard for verification of safe ADS behavior. A further objective is to enable easy use and exchange of rules among various stakeholders involved in ADS safety verification and validation (V&V), including regulators and vehicle manufacturers. The language should be expressive enough to specify the existing rules as they apply to ADSs, facilitating rule analysis, ADS V&V, and scenario generation.

While a complete and exact representation of the rules of the road should consider fuzzy and probabilistic logic interpretations, our methodology focuses on an abstraction level that is practical for our purposes and scalable. In each scenario that an ADS may encounter, rules must be assessed and adhered to at every point in time. Importantly, each individual rule describes the world's state at a singular point in time and therefore does not require memory of past states for evaluation, i.e., they are stateless. Therefore, we concentrate on a first-order logic [28], [29] interpretation of the rules in this article.

We propose the following three step process for translating natural language rules to their codified forms.

1) *Construct a Concept and Predicate Vocabulary:* For a rule, terms relevant to the dynamic driving task are identified. These terms include nouns, verbs, adjectives, pronouns, along with word inflections and terms used for co-referencing. These terms are then compared with standard taxonomies for ADS, such as the ISO 34504 and BSI PAS-1883 ODD taxonomies, and a vocabulary of predicates is developed. Fuzzy terms, synonyms, antonyms are also identified and a normalized predicate vocabulary is constructed.
2) *Reduce the rule to its Minimal Form:* In this step, each rule is meticulously streamlined to its core essence. Extraneous phrases and terms are methodically eliminated, ensuring that only components integral to the rule's essence remain encapsulated within a succinct statement.
3) *Express rule in first-order logic:* Using the predicates and concept vocabulary, express the rule in first-order logic.

Due to the nature of the language used in the highway code, the process of codification is a manual process. It is also important that ADS domain expertise be used to identify and separate terms that are fuzzy - requiring further elaboration and specification.

A rule for an ADS takes the following general form:

$$\mathit{Initialization} \land \mathit{Scenery\text{-}ODD} \rightarrow \mathit{Safety\text{-}Condition}$$

The three components of a rule are described as follows:

- **Initialization:** This sets up the variables that take part in the rule. It specifies the context for a variable. A variable can represent an actor in the scenario, an ODD element (road, road property, weather etc.), position information, and other parameters in the scenario. This part of the rule indicates what each variable represents.
- **Scenery-ODD:** This component then sets up the relations between the variables that require the safety condition to be asserted. For instance, this component could be a check to evaluate if a vehicle is less than the prescribed distance away from the lead vehicle.
- **Safety-Condition:** This component asserts what the vehicle must or must not do in the context of the specified Scenery-ODD. For instance, the vehicle must not drive forward (brake) if too close to the lead vehicle.

IV. SCENARIO GENERATION FOR ADS TESTING AGAINST RULES OF THE ROAD

In the context of Automated Driving Systems (ADS), a scenario represents a sequential progression of scenes over time. The rules governing an ADS are universally applicable in all scenes and at all points in this temporal evolution. Therefore, the rules are continuously in effect throughout the scenario. For the purpose of this article, we focus on non-temporal rules that specify the ADS's actions or restrictions based on defined Operational Design Domains (ODD) or behavioral conditions. These rules are expressed using first-order logic and do not involve implicit temporal constraints, such as time-based counting. Our discussion is thus limited to the behavior governed by these non-temporal rules under specified ODD and behavioral contexts, while acknowledging that performance requirements may incorporate temporal constraints.

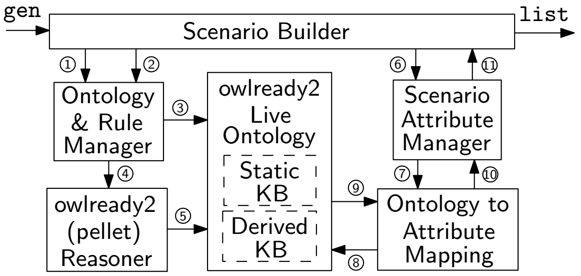

Fig. 1. Scenario generation from Rules of the Road: Methodology Overview

The block diagram presented in Figure 1 provides an overview of the scenario generation process based on codified rules of the road. The entire process is coordinated by the *Scenario Builder* module, which follows a series of steps as reflected in the figure. The broad actions are described in order below:

- *Instruction and Rule Selection:* The *Scenario Builder* receives instructions to generate scenarios from the system or user (`gen`). These instructions specify the rules of interest for which scenarios need to be created. These rules are typically selected based on the specific aspects or behaviors of the ADS under evaluation.
- *Ontology Preparation*: Upon receiving the instruction, the *Scenario Builder* communicates with the *Ontology & Rule Manager*. It instructs the manager to prepare the ontology specifically for scenario generation (*step 1*) and specifies the rules for which scenarios need to be generated (*step 2*). The ontology represents the knowledge base that describes the domain and the relationships between objects relevant to the ADS and the rules of the road.
- *Rule Configuration and Assertion*: The *Ontology & Rule Manager* configures the knowledge base (KB) according to the selected rules (*step 3)*. It initializes the necessary objects, sets up their attributes and relationships, and translates the rules into a form that can be interpreted by the reasoning engine. These rules are then asserted within the KB, which means that they are added as known facts or information.
- *Reasoning and Inference*: The *Ontology & Rule Manager* then requests the *Reasoner* to synchronise the ontology with the newly added assertions (*step 4*). The *Reasoner* is a critical component that applies logical reasoning and inference algorithms to deduce new knowledge from the

asserted rules and existing information within the KB. It resolves any knowledge gaps and infers new relationships and properties between the objects.
- *Knowledge Base Update*: The *Reasoner* updates the KB with the derived knowledge obtained from the inference process (*step 5*). This update includes newly inferred relationships, object properties, and any additional information that was not explicitly stated in the original assertions.
- *Scenario Retrieval*: Once the KB is updated, the *Scenario Builder* proceeds to retrieve scenarios from the *Scenario Attribute Manager* (*steps 6-11*). These scenarios are generated based on the current state of the ontology, which now includes the additional knowledge derived by the *Reasoner*.
- *Abstract Scenario Generation*: The *Scenario Attribute Manager* interacts with the *Ontology to Attribute Mapping* component (*steps 7-10*) to transform the current state of the ontology into abstract scenarios. These abstract scenarios capture the essential elements and interactions between objects.
- *Logical Scenario Generation*: The *Scenario Attribute Manager* further refines the abstract scenarios to create a set of logical scenarios (*step 11*). These logical scenarios represent concrete situations and events that the ADS may encounter while adhering to the specified rules of the road.
- *Output*: The *Scenario Builder* provides the set of logical scenarios as the output of the scenario generation process (`list`).

The scenarios generated by this process may then be used for testing the ADS's behavior, performance, and adherence to the codified rules of the road. They serve as essential evaluation tools for verification and validation during the development and deployment of the Automated Driving Systems.

A. Preparing Rules of the Road for Scenario Generation

We construct our scenario ontology using the taxonomy of the ISO 34503 [30]/BSI PAS 1883 [31].

The *Semantic Web Rule Language* (SWRL) [32] allows for the specification of rules that express logic formulae similar to *Horn Clauses*. A horn statement is of the form $\mathtt{a}_1 \wedge \mathtt{a}_2 \wedge \mathtt{a}_3 \wedge \ldots \wedge \mathtt{a}_n \rightarrow \mathtt{b}$, with its clausal form being $\neg\mathtt{a}_1 \vee \neg\mathtt{a}_2 \vee \neg\mathtt{a}_3 \vee \ldots \vee \neg\mathtt{a}_n \vee \mathtt{b}$, where $\mathtt{a}_i \ (1 \leq i \leq n)$, $\mathtt{b}$ are Boolean variables (*literals*). A horn clause can have atmost one Boolean variable in the consequent of the clause, with all variables in the clause negated except for $\mathtt{b}$. Unlike Horn, SWRL clauses can have multiple positive literals in the clause.

Ontology tools such as Protégé support using SWRL rules to augment the domain ontology. Reasoning engines such as Pellet [33] and HermiT [34] are then used to reason about the ontology in the context of the rules.

Given the ontology, representing knowledge about the domain (ADS, in our case), along with the rules (expressing safety or performance requirements), it then becomes possible to use an automated reasoner (such as Pellet [33] or HermiT [34]) to infer a scenario that will place the ADS in a situation that tests it against the rule or ruleset.

From the perspective of a scenario description, a rule for an ADS can be re-arranged to follow the structure below:

$$Initialisation \wedge ScenarioConds \rightarrow \textbf{SafetyConds}$$

The three components are as follows:
- *Initialisation*: This component, in a rule's antecedent, establishes the variables involved in the rule and defines their context within the scenario. These variables represent various aspects such as actors, elements of the Operational Design Domain (ODD), positional information, weather conditions, and other relevant parameters. By specifying the context of each variable, this part of the rule clarifies their roles in the scenario.
- *ScenarioConds*: This component, in a rule's antecedent, specifies the relationships between scenario parameters that lead to the activation of the safety condition. It acts as a check to determine whether certain conditions, such as vehicle distances, are satisfied within the scenario.
- *SafetyCondition*: This component, in a rule's consequent, asserts the actions or restrictions that the ADS must follow when the specified ScenarioConds conditions are met. For example, it could define actions to avoid collisions or maintain safe distances from other vehicles.

To generate scenarios for each ADS rule, the rule is inverted. The article presents both the non-inverted form, where the scenario conditions are expressed as the antecedent of the rule, and the inverted form, where the safety conditions are swapped with the scenario conditions. The inverted form of the rule takes the following form:

$$Initialisation \wedge \textbf{SafetyConds} \rightarrow ScenarioConds$$

By inverting the conditions in the rule, we can assert scenario conditions based on the safety conditions we want to examine. As a result, the inverted rule enables us to set up scenarios that specifically challenge the ADS to respond to particular safety conditions. By varying these safety conditions, we can thoroughly evaluate the system's performance and verify its adherence to the specified safety requirements. This allows us to systematically generate scenarios that test the ADS behavior under different safety-critical situations, as they relate to the rules. The term *scenario rule* refers to the rule with the swapped conditions.

Rule 1: **No Turning Right:** If a vehicle is in a lane which has a solid line as the right side marking then the vehicle must not change lane right and it must not turn right.
car(c)∧ *laneType(l)*∧ *solidLine(sl)* ∧ *isOn(c,l)*∧ *turnRight(tr)*
∧ *changeLaneRight(lcr)* ∧ *hasRightLaneMarking(l,sl)*
→ *mustNot(c,lcr)* ∧ *mustNot(c,tr)*

Scenario Rule 1: **No Turning Right**
car(c) ∧ *laneType(l)* ∧ *solidLine(sl)* ∧ *turnRight(tl)* ∧
changeLaneRight(lcr) ∧ *mustNot(c,tr)* ∧ *mustNot(c,lcr)*
→ *isOn(c,l)* ∧ *hasRightLaneMarking(l,sl)*

In Rule 1, the predicates *car(.)*, *laneType(.)*, *solidLine(.)*, *changeLaneRight(lcr)* and *turnRight(tr)* impose the requirement that the variables contained as parameters of each predicate are associated with the class type that the predicate refers to. For instance, *car(c)* requires that variable *c* is of type *car*. When all constituent parts of the scenario exist, we use *hasRightLaneMarking(l,sl)* and *isOn(c,l)* to establish the relationships between the variables. This constitutes the scenario condition. The safety condition in the consequent is that the *car c* must not be associated with a lane change right or a turn right manoeuvre. Observe that in the rule, it is the scenario ODD conditions that dictate the safety condition (that the car mustn't do something dangerous), while in the corresponding scenario rule, the scenario terms and safety terms are exchanged to assert the scenario for a situation where the vehicle must adhere to the safety condition.

A similar exercise is used to express the four rules that follow.

Rule 2: **No Turning Left:** If a vehicle is in a lane which has a solid line as the left side marking then the vehicle must not change lane left and it must not turn left.
car(c) ∧ *laneType(l)* ∧ *solidLine(sl)* ∧ *isOn(c,l)* ∧ *turnLeft(tl)* ∧ *changeLaneLeft(lcl)* ∧ *hasLeftLaneMarking(l,sl)*
→ *mustNot(c,lcl)* ∧ *mustNot(c,tl)*

Scenario Rule 2: **No Turning Left**
car(c) ∧ *laneType(l)* ∧ *solidLine(sl)* ∧ *turnLeft(tl)* ∧ *changeLaneLeft(lcl)* ∧ *mustNot(c,tl)* ∧ *mustNot(c,lcl)*
→ *isOn(c,l)* ∧ *hasLeftLaneMarking(l,sl)*

Rule 3: **Safety and Pedestrians:** When a pedestrian is on the same lane as the vehicle and is at its front, the vehicle can not drive forward.
car(c) ∧ *pedestrian(p)* ∧ *drive(d)* ∧ *laneType(l)* ∧ *isOn(c,l)* ∧ *isOn(p,l)* ∧ *isFrontOf(p,c)* → *mustNot(c, d)*

Scenario Rule 3: **Safety and Pedestrians**
car(c)∧ *pedestrian(p)*∧ *drive(d)*∧ *laneType(l)*∧ *mustNot(c,d)*
→ *isOn(c,l)* ∧ *isOn(p,l)* ∧ *isFrontOf(p,c)*

Rule 4: **Reversing on One-Way Roads:** On a one-way road, a vehicle must not reverse if reversing moves it against the driving direction for the road.
car(c) ∧ *laneType(l)* ∧ *reverse(r)* ∧ *unidirectional(ud)* ∧ *isOn(c, l)* ∧ *hasProperty(l, ud)* → *mustNot(c, r)*

Scenario Rule 4: **Reversing on One-Way Roads**
car(c) ∧ *laneType(l)* ∧ *reverse(r)* ∧ *unidirectional(ud)* ∧ *mustNot(c, r)* → *isOn(c, l)* ∧ *hasProperty(l, ud)*

B. Dealing with Action Sequences in Rules

A temporal aspect characterizes certain rules of the road, wherein these rules regulate the conduct of an ADS over a series of sequential actions, such as during overtaking maneuvers. An overtaking maneuver entails a sequence of actions including lane changes to surpass another vehicle, followed by positioning ahead of the overtaken vehicle, succeeded by another lane change. By imposing safety prerequisites prior to executing each action in the sequence, these rules guide these actions. Consequently, an ADS contemplating an overtaking maneuver must assess its ability to successfully execute the planned sequence of actions while adhering to the rule's constraints. However, due to challenges in perceiving the environment, the ADS might be incapable of predetermining the safety of each step before initiating the entire maneuver. In response, the ADS could adopt an optimistic or pessimistic approach. In the pessimistic scenario, the ADS might opt to entirely forgo the maneuver, whereas in the optimistic scenario, it might endeavor to perform as many actions as feasible within safety limits until reaching a point where an action can no longer be deemed safe due to its dynamically evolving surroundings. Multiple scenarios may emerge in which the vehicle can initiate the maneuver, with the quantity of such scenarios contingent upon the length of the rule-defined sequence.

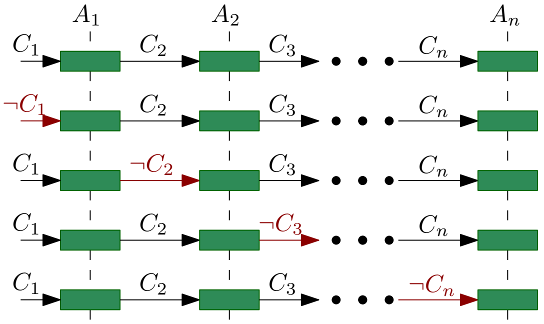

Fig. 2. Series of Rule violation scenarios for rule with n sequential conditions C_1, C_2, ..., C_n, and actions A_1, ..., A_n

A breach of safety regarding such a rule would encompass the ADS erroneously deeming an action in the sequence as safe to execute, contrary to the stipulations of the rule. In alignment with this comprehension, the array of scenarios for testing this rule would encompass the following: a scenario wherein all constraints of the rule are fulfilled, as well as separate scenarios for each instance where an environmental factor violates one of the rule's constraints. This is illustrated in Figure 2, which depicts a representational set of scenarios required to challenge the ADS against a rule governing a sequence of n actions A_1, A_2, ..., A_n. Before each action the rule stipulates a set of conditions C_1, C_2, ..., C_n. The initial scenario, illustrated in the topmost row of the diagram, showcases the fulfillment of all conditions. Subsequent scenarios, however, progressively deviate, wherein each successive instance portrays the non-fulfillment of one of the n conditions.

A rule organized as a series of n sequential conditions and corresponding actions can be logically decomposed into a collection of n sub-rules, each pertaining to a specific segment of the sequence. Each sub-rule serves to define the conditions and actions within its corresponding phase. Such a sub-rule is structured as follows:

Initialisation ∧ *PhaseConditions* ∧ *ScenarioConditions*
→ **SafetyConditions**

Conversely, its inverse form takes the shape:

$$\begin{aligned}&\mathit{Initialisation} \ \wedge \ \textbf{SafetyConditions}\\ &\qquad \rightarrow \ \mathit{PhaseConditions} \ \wedge \ \mathit{ScenarioConditions}\end{aligned}$$

Here, *PhaseConditions* ensure that the scenario conditions align with the designated phase of the sequence. This crucial alignment guarantees that the generated scenarios faithfully adhere to the inherent organization of conditions and corresponding actions. We employ the predicate $inPhase(a, p)$ to confirm that object a, initialised in the rule, pertains to phase p, where p is a numeric designation, and a signifies an ODD or behavior aspect that must be explicitly validated within the operational environment of the ADS. This aspect plays a pivotal role in constructing the scenarios generated during the evaluation process.

V. CASE STUDIES AND RESULTS

In this section, we'll look at real-life examples to show how the method works. We'll start by using a simple example to explain it visually. After that, we apply the method to rules taken from the UK highway code (UKHC) and Vienna conventions (VC) to generate a logical scenario in the WMG-SDL Level 2 scenario language [35], [36].

A. Walkthrough of the Methodology

We initialize the ontology to contain object instances representing objects that may take part in a scenario. The objects are meant to be representative of actual object and actors in the scenario. For our example here, we use a *drive*, *car*, *laneType*, *changeLaneLeft*, *changeLaneRight*, *unidirection*, *laneMarkings*, *pedestrian*, *reverse*, *turnLeft*, *turnRight* objects.

Initially, the object instances are not associated with any properties other than the properties they inherit from their classes. For instance, the location of the pedestrian is unknown at this stage. It is unknown if a rule would result in the involvement of the pedestrian. Objects are instantiated so that they may take part in a scenario, but do not necessarily need to. A rule may require relationships between some objects to be initialized, and may not require relationships between others. Figure 3 depicts the state of the ontology when individual instances of objects have been initialized with no relationships between them asserted.

From the rules in Section IV-A, a scenario can be generated from one or more rules simultaneously. The generated scenario would incorporate the relationships inferred from the chosen rules and the ontology.

Given a collection of rules denoted as R_1, R_2, ..., R_n, let us consider that each rule encompasses a safety condition that necessitates testing. Activation of a rule's safety condition simultaneously triggers the activation of the rule itself. By activating the safety conditions of multiple rules, it becomes feasible to activate multiple rules concurrently. For example, assuming R_3, R_8, and R_{63} symbolize the safety conditions pertaining to rules 3, 8, and 63 respectively, and if there exists an individual referred to as *start* within the *rule* class, the subsequent rule expression would activate rules 3, 8, and 63 as follows:

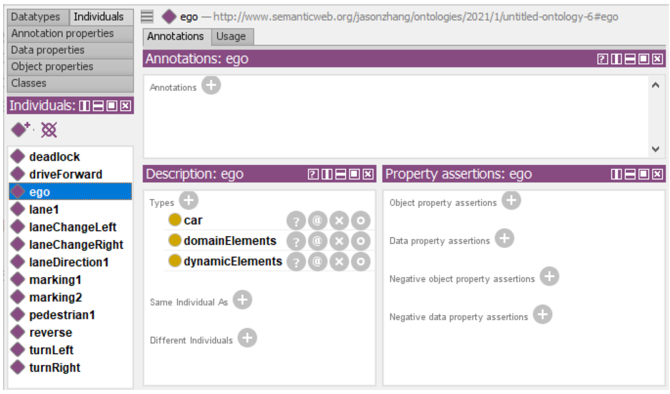

Fig. 3. Screenshot of Protégé showing an empty property assertions list for the car object (named *ego*) on the right pane.

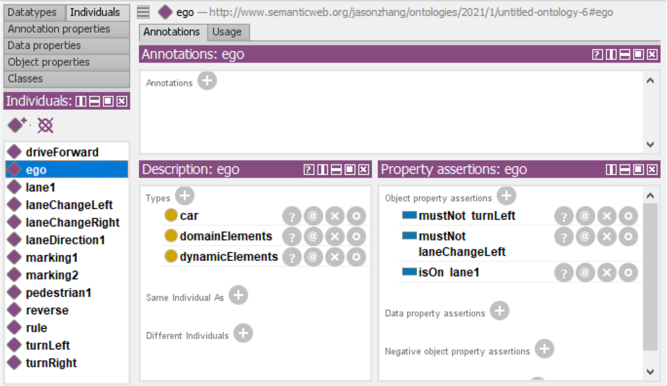

Fig. 4. Screenshot of Protégé showing a non-empty property assertions list for the car object (named *ego*) on the right pane after evaluating Scenario Rule 2: No Turning Left.

$$start \rightarrow R_3 \wedge R_8 \wedge R_{63}$$

In cases where safety conditions reference scenario objects, it is imperative to ascertain the existence of these objects by either testing for their presence or enforcing their inclusion as an integral part of the rule. Verification of their existence can be conducted by appending a test to the antecedent. Consequently, the rule transforms into the following form: $start \wedge testConditions \rightarrow safetyConditions$.

Subsequently, the utilization of an ontology reasoner in conjunction with SWRL rules can be employed to generate the scenario linked with the rule, as well as other ontological conditions. The outcomes of employing this reasoning approach are demonstrated for the ego in Figure 4.When evaluating Scenario Rule 2: No Turning Left, a ripple effect of inferred relations among objects within the scenario becomes apparent. This is visually depicted through populated property assertions for specific individuals, indicating novel object relationships deduced from the rules.

By iteratively examining the properties of all individuals present, a comprehensive collection of property assertions and descriptions can be amassed, encompassing all pertinent actors and elements from the ODD that partake in the generated scenario. In this instance, the generated scenario is functionally described, stipulating that "The car is positioned on lane1, which features a left lane marking identified as marking1, characterized by a solid line."

The outcomes of the reasoning process can be exported as a CSV file, amenable for further processing to generate a scenario in a preferred format. Depending on the level of detail incorporated within the ontology, the resultant scenario may adopt the form of a functional scenario (if the ontology is abstract) or manifest as a logical or concrete scenario, as discussed in previous literature [37].

We now present two case studies of scenario generation from rules. The rules are derived from the UK Highway Code [16] and the Vienna Conventions [17].

B. Pedestrian Crossings (Article 21 VC, Rule 195 UKHC)

```
...
SCENERY ELEMENTS:
DO: Map –roads and junctions network [Network1] as:
Junctions: None
Roads:
R1: START
    Road type [Minor road] as [R1] with zone as [N/A]
    AND speed limit of [30] in an [Urban] environment
    with Number of lanes [2] as [R1.L1, R1.L–1]
    Road traffic direction [Left–handed]
...
    Special road structures [Pedestrian crossing]
    as [PC1] at [10 to 20] distance from START
...
DYNAMIC ELEMENTS:
INITIAL: Vehicle [Ego] in [R1.L1]
    AND Pedestrian [U1] in [R1.L1]
        with a [Longitudinal] offset of [10 to 20]
        AND at relative position [F]
        with relative heading angle [85 to 95] to [Ego]
    AND Global timer [T1] = [0]
WHEN: [Ego] is [Going_Ahead]
    DO: [U1]
        PHASE 1: [Stopped] [–, 0 to 0, 0 to 0] [Ego: 6 to 8, R]
...
END
```

SDL Description 1. Pedestrian Crossing Rule in WMG SDL Level 2

In accordance with Article 21 of the Vienna Convention, *drivers are mandated to halt their vehicles whenever pedestrians are in the process of crossing or stepping onto a designated pedestrian crossing, as indicated by signage or road markings*. This provision, outlined in the Convention, emphasizes the importance of prioritizing pedestrian safety at such crossings. While the full text of the rule contains intricate legal and technical language, the essence of the requirement is distilled in the quoted excerpt above for clarity and succinctness, thereby ensuring the primary focus remains on the crucial obligation of drivers to yield to pedestrians at pedestrian crossings. An equivalent is represented in a part of Rule 195 of the UK highway code, *[a]s you approach a zebra crossing[,] look out for pedestrians waiting to cross and be ready to slow down or stop to let them cross*. On the contrary this representation, as given in the UK highway code, has been provided for human drivers and with that can come a separate set of difficulties when expressing the logic of a rule.

The following rule is articulated using first-order logic, adapted to the SWRL syntax, and is presented as follows:

$$\begin{aligned}&ego(x) \land pedestrian(u) \land laneType(y) \land zebraCrossing(z) \land \\ &isAheadOf(z, x) \land isOn(x, y) \land isOn(u, z) \land stop(s) \\ &\qquad\qquad\qquad\qquad \rightarrow mustDoAction(x, s)\end{aligned}$$

It's important to note that the rule's presentation has been streamlined for the purpose of illustration. More intricate scenario components, like the precise interpretation of `isAheadOf`, may be explicitly defined within the Ontology. In our process of generating scenarios, we make an assumption about a range of longitudinal distances existing between the Autonomous Driving System (ADS) and the pedestrian. This assumption can be further refined to align with specific test objectives as required by the user.

A scenario is generated from this rule and expressed using the WMG Scenario Description Language (Level 2) [35], [36]. The dynamic initialisation within the scenario is reproduced in SDL Description 1.

C. Sequenced Overtaking (Rules 162, 163 UKHC)

In this illustrative instance of a sequential rule, our attention is directed toward depicting overtaking maneuvers as outlined in the UK highway code. Specifically, we center our focus on Rule 162 and the relevant portions of Rule 163. Given that these rules were not originally formulated with consideration for an ADS, our approach involves restructuring the pertinent elements of each rule to correlate them with the specific phases of an overtaking manoeuvre.

Rule 162 delineates the prerequisites that should be met before initiating an overtaking maneuver, encompassing considerations such as the road's visibility, ongoing overtaking actions by other road users, and the presence of an appropriate gap ahead of the vehicle to be overtaken. While the entirety of Rule 163 provides guidelines for safe overtaking, only specific segments are extracted and integrated into our logical framework, aligning with distinct phases of the overtaking sequence. Notably, these extracted sections encompass the *[imperative to] maintain a safe distance from the leading vehicle during the overtaking process, the subsequent swift passage past the overtaken vehicle, and the maintenance of a safe distance to the overtaken vehicle when re-entering the lane*. The careful selection of relevant portions of Rule 163 ensures that our logical representation captures the critical elements influencing the sequence of an overtaking maneuver, while excluding extraneous aspects that pertain to broader contexts and not directly tied to the temporal progression of the overtaking manoeuvre itself.

This rule has a sequential nature, and to develop it into rules, we express the sub-rules as follows:

a) Phase-1: Preliminary conditions and changing lane right: Between the two relevant rules in the highway code that pertain to overtaking, specific conditions and actions occur during the initial phase of the maneuver. In this context, the *first phase* involves several prerequisites. Firstly, *the road ahead*, which pertains to the lane required for the overtaking, *must be appropriately clear*. Additionally, it's necessary to ensure that *vehicles behind you are not in the process of overtaking*, and furthermore, *there must exist an adequate* *gap (referred to as headway) ahead of the vehicle you intend to overtake*. Once these conditions are met, this phase also encompasses the ego vehicle's action of changing lanes, specifically moving into the designated *overtaking* lane. The corresponding rule governing the elements within the scenario is outlined below:

$$\begin{aligned}
&\mathit{ego}(x) \wedge \mathit{laneType}(y) \wedge \mathit{isOn}(x, y) \wedge \mathit{laneType}(z) \wedge \\
&\mathit{isRightOf}(z, y) \wedge \mathit{isClearAhead}(z, \mathit{true}) \wedge \mathit{car}(u) \wedge \\
&\mathit{isAheadOf}(u, x) \wedge \mathit{car}(v) \wedge \mathit{isRearOf}(v, x) \wedge \mathit{overtake}(o) \wedge \\
&\mathit{hasSuitableGap}(u, \mathit{true}) \wedge \mathit{isNotPerforming}(v, o) \\
&\qquad\qquad\qquad\qquad \rightarrow \mathit{canDoAction}(x, o)
\end{aligned}$$

b) Phase-2: Safely passing vehicle: Moving to the subsequent phase, the ego vehicle's actions influenced by these conditions are as follows; the ego's primary action here involves acceleration, enabling it to pass the overtaken vehicle in a prompt manner. This acceleration should ensure a swift overtake while also maintaining a safe distance between the ego vehicle and the overtaken vehicle. During this manoeuvre phase, the ego must check its environment to ensure that the conditions of the overtake continue to hold. The corresponding rule governing the elements within the scenario is outlined below:

$$\begin{aligned}
&\mathit{ego}(x) \wedge \mathit{laneType}(y) \wedge \mathit{laneType}(z) \wedge \mathit{overtake}(o) \wedge \\
&\mathit{isRightOf}(z, y) \wedge \mathit{isOn}(x, z) \wedge \mathit{isClearAhead}(z, \mathit{true}) \wedge \\
&\mathit{car}(u) \wedge \mathit{isAheadOf}(u, x) \wedge \mathit{hasSuitableGap}(u, \mathit{true}) \wedge \\
&\qquad\qquad\qquad\qquad \rightarrow \mathit{canDoAction}(x, o)
\end{aligned}$$

Additionally, it's worth noting that a rule could potentially be deduced to oversee the behavior of the ego vehicle. However, since this aspect doesn't impact the scenario generation process, we omit its representation within this context.

c) Phase-3: Safely re-entering lane: In this concluding phase, the ego vehicle executes a lane change to the left, returning to its original lane. To fulfill the criteria for completing this phase, it's necessary to uphold a secure distance that prevents any abrupt *cut-in* actions from occurring. The corresponding rule governing the elements within the scenario is outlined below:

$$\begin{aligned}
&\mathit{car}(x) \wedge \mathit{laneType}(y) \wedge \mathit{laneType}(z) \wedge \mathit{overtake}(o) \wedge \\
&\mathit{isRightOf}(z, y) \wedge \mathit{isOn}(x, z) \wedge \mathit{isClearAhead}(y, \mathit{true}) \wedge \\
&\mathit{car}(u) \wedge \mathit{isAheadOf}(x, u) \wedge \mathit{hasSuitableGap}(u, \mathit{true}) \wedge \\
&\qquad\qquad\qquad\qquad \rightarrow \mathit{canDoAction}(x, o)
\end{aligned}$$

Utilizing the proposed approach, scenarios are generated based on the aforementioned rule and are represented using the WMG Scenario Description Language (Level 2) [35], [36]. The provided scenario exemplifies a fusion of two distinct instances where potential rule violations might occur during separate phases of the scenario, as dictated by the rule's specific phases. These scenarios position the Ego in a state where aspects of the rule are violated, preventing the successful completion of the overtaking maneuver. If the Ego were to attempt the overtake under these violations, it would be in breach of the established rules.

In phase 1 of the scenario, a violation becomes evident as vehicle V1, initially trailing the ego vehicle, undertakes a lane change to the right, initiating an overtaking maneuver that directly contradicts the conditions outlined in phase 1 of the rules.

```
...
SCENERY ELEMENTS:
DO: Map –roads and junctions network
[Network1] as:
Junctions: None
Roads:
R1: START
    Road type [Minor road] as [R1] with zone as [N/A] AND
    speed limit of [50] in an [Urban] environment
    with Number of lanes [2] as [R1.L1, R1.L–1]
    Road traffic direction [Left–handed]
...
DYNAMIC ELEMENTS:
INITIAL: Vehicle [Ego] in [R1.L1]
    AND Vehicle [V1] in [R1.L1]
        with a [Longitudinal] offset of [–5 to –10]
        AND at relative position [R] with
        relative heading angle [0 to 5] to [Ego]
    AND Vehicle [U1] in [R1.L11] with a
        [Longitudinal] offset of [5 to 10]
        AND at relative position [F] with
        relative heading angle [0 to 5] to [Ego]
    AND Global timer [T1] = [0]
WHEN: [Ego] is [Going_Ahead]
  DO: [V1]
    PHASE 1: [LaneChangeRight_CutOut] [–, 10 to 14, 2 to 5]
    [Ego: 3 to 5, R]
...
    PHASE 1: [Drive_Towards] [–, 6 to 8, 0 to 0] [Ego: –4 to –2, R]
    PHASE 2: [Drive_Towards] [–, 6 to 8, 0 to 0] [Ego: –4 to –2, R]
    PHASE 3: [Drive_Towards] [–, 6 to 8, 0 to 0] [Ego: –4 to –2, R]
  AND: [U1]
    PHASE 1: [Drive_Away] [–, 6 to 8, 0 to 0] [Ego: –4 to –2, F]
    PHASE 2: [Drive_Away] [–, 6 to 8, 0 to 0] [Ego: –4 to –2, F]
        WHILE: [U1] [Longitudinal] offset to Ego is > [2]
    PHASE 3: [Drive_Towards] [–, 8 to 12, 2 to 4] [Ego: –2 to 2, RSR]
END
...
```

SDL Description 2. Sequenced Overtaking Rule in WMG SDL Level 2

A possible second rule violation could transpire in Phase 3 of the scenario. Here, as the ego vehicle initiates its leftward lane change, symbolizing its return to the original lane, it comes within a 2-meter range of the front of vehicle U1. Subsequently, the vehicle U1 accelerates, reducing the gap between itself and the Ego to a degree that qualifies as an *unsafe distance*.

The scenario's dynamic initialization is structured as depicted in SDL Description 2.

VI. DISCUSSIONS

This section discusses our finding and proposals for future avenues of research.

A. Rules of the Road designed for ADS

In this investigation, we conducted an examination of current road regulations across various jurisdictions. Our observations reveal that a majority of these regulations are formulated using unstructured natural language. Such regulations often suffer from incomplete specifications and ambiguous expressions, as exemplified by statements like, 'there is a suitable gap in front of the road user you plan to overtake.' While such linguistic constructs are suitable for human comprehension, allowing for user interpretation of terminologies, conveying the same level of precision to an automated system proves to be laborious.

To facilitate the integration of these regulations into ADS, a process of formal codification becomes imperative. This codification process serves the dual purpose of unveiling inherent issues within the regulations and establishing unequivocal semantics for them. The resultant codified regulations assume significance in the realm of ADS, serving not only for evaluation purposes but also during operational phases—enabling continuous monitoring of the system and safeguarding against unsafe behaviors.

Rule codification entails the transformation of regulations into a formal framework such as the one proposed in Ref [27], ensuring consistency in description suitable for adoption by regulatory bodies and vehicle manufacturers. This endeavor encompasses the identification of all regulations pertinent to ADS, subsequently subjecting them to the codification process. The codified 'rules of the road' thus obtained encapsulate the essential intricacies of pre-existing regulations, thereby facilitating rule analysis, verification, validation (V&V), and scenario generation.

Effective codification mandates precise delineation, accompanied by a collection of unambiguous semantic definitions that elucidate the intended meanings of the codified regulations. Once codified, these regulations are amenable to literal interpretation in alignment with the provided semantic definitions. The methodology expounded in this article employs a meticulous logical interpretation of certain extant regulations.

B. *Concretising Scenarios from Rules*

In view of a logical scenario developed through the approach outlined in this article, a subsequent challenge emerges in discerning precise attribute values for these scenarios. Such values assume a pivotal role in the examination of boundary cases within the rule structure.

One viable approach involves the utilization of Satisfiability Modulo Theories (SMT) solvers, which can be employed to detect corner cases by introducing supplementary goal constraints. Alternatively, the integration of Machine Learning models and techniques from Artificial Intelligence, such as Bayesian Optimization [18], presents another avenue. This approach facilitates the construction of tangible scenarios by leveraging suitable failure objectives.

C. *Expressing Deadlocks for ADS Driving Task*

It is possible that some rules when considered together, lead to a planning deadlock for the ADS. A deadlock within the rule-set is defined by a collection of rules that when taken together can in some situations prevent the ADS from making any progress, i.e. the ADS stops and cannot manoeuvre out of the situation.

To identify deadlock causing scenes, it is necessary to associate rules with actions in a way that allows us to define a deadlock, and then work backward through the rule structure to generate situations that may lead to them. These situations are those that are generated directly from the rules themselves. Recall that rules in the Highway Code may be expressed in the form of,

$$\mathit{Situation} \rightarrow \mathit{Action} \text{ OR } \mathit{Situation} \rightarrow \neg \mathit{Action}$$

In general, for a deadlock to occur, it must be a situation that prohibits any action from being taken. Therefore, to decide that a deadlock has occurred, it must be the case that every action is examined and found to be prohibited by the rule-set. This definition of a deadlock is logically expressed as,

$$\left(\mathit{Actor}(x) \wedge \bigwedge_{a \in \mathbb{A}} \mathit{Action}(a) \wedge MustNot(x, a) \right) \leftrightarrow \mathit{Deadlock}(x)$$

where $\mathbb{A}$ is the set of all actions, 'x' and 'a' are variables. Here, *mustNot* is a predicate that has two parameters, x which is an object, and the action name. If the predicates are all true, that is "x must not change lane and x must not turn and x must not drive etc.", then x is in a deadlock, and vice-versa. The operator symbol $\leftrightarrow$ is used to express a bi-conditional logical statement, where the left and the right side of the operator are taken to be logically equivalent; one cannot happen without the other being true. One cannot be in a deadlock if one can change lane, or turn, or drive; and similarly, if one must not perform any action, a deadlock must exist.

To identify rules that may cause a deadlock, we must be able to reason backward. That is, beginning from the existence of a deadlock, being unable to perform any action, identify which rules are associated with the action, and then work backwards through the implication chain to establish under what circumstances a collection of rules could become applicable to cause the deadlock. This may be achieved using the rule for deadlocks described above, together with the codified set of rules of the road.

Considering the substantial volume of rules involved, it becomes essential to develop a scalable mechanism capable of identifying these deadlocks. This would then enable suitable modifications to be made within the rule-set to prevent such deadlocks effectively.

VII. CONCLUSIONS

This article examines the use of ontologies to generate scenarios that target testing against rules of the road (to express safety or performance requirements). The methodology depends on an ontology for ADS scenarios. Rules written in SWRL are used in conjunction with the ontology to automatically reason and infer relationships among scenario objects. These relationships and instantiations form the scenario that targets the rule. We use the open-source tool Protégé along with the inbuilt HermiT reasoner for generating scenarios. The complexity and abstraction level of the ontology dictate the detail in the scenario. The ontology instance generated by this method can be exported as an XML which can then be further processed to generate scenarios compatible with V&V toolchains.

We demonstrate the methodology on a set of example rules and for one of the examples present how the rule can be used to automatically construct a scenario.

ACKNOWLEDGEMENT

The work presented in this paper has been supported by UKRI Future Leaders Fellowship (Grant MR/S035176/1), Department of Transport, UK (Contract No. TETI0042), and Transport Canada (Contract No. T8080-220112). 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. No new data were created in this study. For the purpose of open access, the author(s) has applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

REFERENCES

[1] Society of Automotive Engineers, "SAE Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, Standard," Apr. 2021.

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

[3] 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, p. 35–54, 2016.

[4] S. Khastgir, B. Stewart, G. Dhadyalla, and P. Jennings, "The science of testing: An automotive perspective," SAE Technical Paper 2018-01-1070, Tech. Rep., 2018.

[5] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings, "Systems Approach to Creating Test Scenarios for Automated Driving Systems," *Reliability Engineering & System Safety*, vol. 215, p. 107610, nov 2021.

[6] 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," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 312–319.

[7] E. De Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. o. d. Camp, S. Kraines, J. Ploeg, and B. De Schutter, "Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022.

[8] 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.

[9] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.

[10] 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.

[11] N. Webb, D. Smith, C. Ludwick, T. Victor, Q. Hommes, F. Favaro, G. Ivanov, and T. Daniel, "Waymo's safety methodologies and safety readiness determinations," *CoRR*, vol. abs/2011.00054, 2020. [Online]. Available: https://arxiv.org/abs/2011.00054

[12] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and Substantiating the Terms Scene , Situation , and Scenario for Automated Driving," *Proc. of the 2015 IEEE 18th 18th Int. Conf. on Intelligent Transportation Systems*, pp. 982–988, 2015.

[13] D. Zöbel, "The deadlock problem: A classifying bibliography," *SIGOPS Oper. Syst. Rev.*, vol. 17, no. 4, p. 6–15, oct 1983. [Online]. Available: https://doi.org/10.1145/850752.850753

[14] S. Singh and S. S. Tyagi, "A review of distributed deadlock detection techniques based on diffusion computation approach," *International Journal of Computer Applications*, vol. 48, no. 9, pp. 28–32, June 2012.

[15] M. Gómez-Zamalloa and M. Isabel, "Deadlock-guided testing," *IEEE Access*, vol. 9, pp. 46 033–46 048, 2021.

[16] U. K. Department for Transport, "The Highway Code, Road Safety and Vehicle Rules," Jan. 2022. [Online]. Available: https://www.gov.uk/guidance/the-highway-code

[17] E. C. for Europe: Inland Transport Committee, "Vienna convention on Road Traffic," Nov 1968. [Online]. Available: https://unece.org/fileadmin/DAM/trans/conventn/crt1968e.pdf

[18] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, "Identification of test cases for automated driving systems using bayesian optimization," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1961–1967.

[19] E. Esenturk, S. Khastgir, A. Wallace, and P. Jennings, "Analyzing Real-world Accidents for Test Scenario Generation for Automated Vehicles," pp. 288–295, 2021.

[20] U. N. E. C. for Europe, "UN Regulation No. 157 - Automated Lane Keeping Systems (ALKS)," 2021. [Online]. Available: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-157-automated-lane-keeping-systems-alks

[21] F. Wotawa, *Testing Autonomous and Highly Configurable Systems: Challenges and Feasible Solutions*. Springer International Publishing, 2017, pp. 519–532.

[22] F. Wotawa and Y. Li, "From ontologies to input models for combinatorial testing," in *Testing Software and Systems*, I. Medina-Bulo, M. G. Merayo, and R. Hierons, Eds. Springer International Publishing, 2018, pp. 155–170.

[23] Y. Li, J. Tao, and F. Wotawa, "Ontology-based test generation for automated and autonomous driving functions," *Inf. Softw. Technol.*, vol. 117, no. C, jan 2020. [Online]. Available: https://doi.org/10.1016/j.infsof.2019.106200

[24] ASAM, "ASAM OpenXOntology," 2022. [Online]. Available: https://www.asam.net/project-detail/asam-openxontology/

[25] ——, "ASAM OpenSCENARIO® Standard," 2021. [Online]. Available: https://www.asam.net/standards/detail/openscenario/

[26] ——, "ASAM OpenDRIVE® Standard," 2021. [Online]. Available: https://www.asam.net/standards/detail/opendrive/

[27] P. Irvine, A. A. B. Da Costa, X. Zhang, S. Khastgir, and P. Jennings, "Structured natural language for expressing rules of the road for automated driving systems," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–8.

[28] I. Chiswell and W. Hodges, *Mathematical Logic*. USA: Oxford University Press, Inc., 2007.

[29] C. C. Leary, *A Friendly Introduction to Mathematical Logic*, 1st ed. USA: Prentice Hall PTR, 1999.

[30] ISO, "ISO/DIS 34503 Road Vehicles — Test scenarios for automated driving systems — Specification for operational design domain," 2023.

[31] BSI, "PAS 1883:2020 Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) — Specification," 2020. [Online]. Available: https://www.bsigroup.com/en-GB/CAV/pas-1883/

[32] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "W3, SWRL: A Semantic Web Rule Language Combining OWL and RuleML," May 2004. [Online]. Available: https://www.w3.org/Submission/SWRL/

[33] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007, software Engineering and the Semantic Web. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826807000169

[34] Data and U. o. O. Knowledge Group, "The New Kid on the OWL Block, HermiT OWL Reasoner," 2022. [Online]. Available: http://www.hermit-reasoner.com

[35] X. Zhang, S. Khastgir, and P. A. Jennings, "Scenario Description Language for Automated Driving Systems: A Two Level Abstraction Approach," in *2020 IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 973–980.

[36] A. A. B. da Costa, P. Irvine, X. Zhang, S. Khastgir, and P. A. Jennings, "Writing accessible and correct test scenarios for automated driving systems," in *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2022, Prague, Czech Republic, October 9-12, 2022*. IEEE, 2022, pp. 1356–1363. [Online]. Available: https://doi.org/10.1109/SMC53654.2022.9945564

[37] C. Neurohr, L. Westhofen, M. Butz, M. Bollmann, U. Eberle, and R. Galbas, "Criticality Analysis for the Verification and Validation of Automated Vehicles," *IEEE Access*, vol. 9, no. i, 2021.

ANTONIO A. BRUTO DA COSTA (S'15), earned his B.Eng. degree in Computer Engineering from Goa Engineering College and his M.Tech and Ph.D. from the Department of Computer Science and Engineering at IIT Kharagpur. He is specialized in formal verification, languages, and causal analysis with over 10 years of experience. Currently, he is the Scenarios lead in the Verification and Validation team at the WMG, University of Warwick, UK, and contributes to international standardization projects for automated driving systems.

PATRICK IRVINE received his B.Eng. degree in Automotive Engineering in 2019 from the School of Engineering, University of Warwick, and Pg.Cert in Automotive Technology in 2021 from WMG, University of Warwick. Since 2019 he has been engaged with research in the field of Validation and Verification, first through a Graduate Training Scheme, and from 2021 as a Project Engineer. He has been contributing to research in verification, languages and scenario generation for the scenario-based safety assurance of automated systems.

XIZHE ZHANG (JASON) received the B.Eng. and the Ph.D. degrees from the Department of Mechanical Engineering in the University of Bath, UK. He is the Simulation Lead at the validation and verification team in WMG, University of Warwick. His expertise covers scenario-based and simulation-based safety assurance workflow for automated systems, ontology design and implementation, language development and toolchain development. He has also been leading the development of several international standardisation projects with the goal of unifying and accelerating activities within the autonomous driving domain.

SIDDARTHA KHASTGIR is a professor at WMG, University of Warwick. He holds a PhD in the area of ADS testing and Human Factors (trust in automation). He is an expert in ADS/ADAS safety verification and has significant experience in the ADS domain include test scenarios generation, safety, simulation-based testing, standardization and international regulations. He has over 10 years' experience in safety verification and validation

PAUL JENNINGS is a professor and the research director at WMG, University of Warwick. He has over 30 years of experience in industrial collaborative R&D projects, having led research projects worth over £60M. Before taking up his role as Research Director at WMG, he set up and led the Intelligent Vehicles group at WMG with a key focus on Safety. He has published over 100 academic papers and has been Principal Investigator on over 30 research grants.