Writing Accessible and Correct Test Scenarios for Automated Driving Systems

Antonio Bruto da Costa
WMG
University of Warwick
Coventry, United Kingdom
antonio.bruto-da-costa@warwick.ac.uk

Patrick Irvine
WMG
University of Warwick
Coventry, United Kingdom
patrick.irvine@warwick.ac.uk

Xizhe Zhang
WMG
University of Warwick
Coventry, United Kingdom
jason.zhang@warwick.ac.uk

Siddartha Khastgir
WMG
University of Warwick
Coventry, United Kingdom
s.khastgir.1@warwick.ac.uk

Paul Jennings
WMG
University of Warwick
Coventry, United Kingdom
paul.jennings@warwick.ac.uk

***Abstract*—For Automated Driving Systems (ADSs), vehicle safety and functional correctness are assessed against scenarios the ADS would encounter - within or outside its operational design domain (ODD). A scenario specifies conditions and events that an ADS is expected to respond to when deployed. Scenario specifications underpin the V&V life-cycle, and are used by a diverse set of stakeholders - from engineers to regulators. Due to the diversity in stakeholder expertise, scenarios must be available at different levels of detail, and further the chance of writing syntactically or semantically incorrect scenarios is high. Present-day Scenario Description Languages (SDLs) need to be supported by technologies to help authors compose scenarios and provide a mechanism for easy translation into executable forms for virtual or real-life testing. This paper addresses these issues by building on the existing two-level abstraction WMG-SDL in the following ways, (1) introducing a human-readable, natural language SDL, replacing the former *Level-1* SDL, and complementing the more detailed *Level-2* SDL, which is now syntax aligned with the ODD Taxonomy defined in ISO 34503, and (2) providing a tool consisting of a parser and a validator to assist writing syntactically and semantically correct scenarios. Our tool may be used within a graphical scenario editing interface or on the command-line. Further, for developers, an object-oriented interface for parsed scenarios enables further development and integration with off-the-shelf ADS simulation and language tools. The tools and technologies described in this paper are to be made open-source.**

***Index Terms*—scenario, scenario definition language, parser-validator, verification and validation, automated driving systems.**

I. Introduction

Over the past decade, developments in vehicle autonomy have been progressing rapidly. The need for autonomy is being increasingly asserted due to potential for reduced on-road injuries and fatalities [1], as well as improved traffic characteristics [2]. In an effort to develop automated driving systems (ADS') that realize a 20% quality improvement over human drivers, it was previously believed that 11 billion training and testing miles would be required. This philosophy has rightly evolved to stress on quality of miles driven over quantity [3]. The philosophy of quality of miles expresses the need to evaluate ADS function and performance under varying conditions that the ADS may find itself when operating, and one that exposes itself to hazards [4]. The operating conditions (environmental, road, and other dynamic conditions) under which an automated driving system (ADS) operates is termed as its Operational Design Domain (ODD). In the verification and validation (V&V) life-cycle for an ADS, *scenarios* are the key assets used to identify failures [3], [5], [6], and may be defined on the basis of the ADS's ODD. Ulbrich et al [6] define a scenario as 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.'*

Due to the widespread adoption of scenario-based testing as a part of the V&V approaches [7]–[11], many Scenario Description Languages(SDLs) have been developed. A SDL is a plain-text format for describing a scenario. Some of the available languages include the WMG two-level abstraction SDL [12], Scenic [13], Fortellix M-SDL [14], GeoScenario [15] and ASAM OpenX Standards [16], [17]. Other specification techniques such as graphical interfaces for scenario generation and testing, including IPG CarMaker [18] and MATLAB toolboxes [19] support ASAM OpenX-Standards for scenario descriptions. At the time of writing this article, with different parties having different preferences in specification, no universal SDL for ADS scenarios exists. While the ASAM OpenX-Standards body has attempted to develop a standard SDL, the existing OpenScenario(v1.1) [16] and OpenDrive(v1.7) [17] languages have had limited adoption due to the diverse set of stakeholders who need to use them and due to internal processes of various organisations. Designing an SDL is a challenging exercise, and most available SDL solutions are single syntax solutions, that try to be everything at once - composable (allowing mixing component specifications across scenarios), extensible, reusable, readable to diverse audiences and expressive (allowing specifying high levels of detail). Readability and expressiveness are mutually opposing characteristics. *Non-technical readers of scenarios (such as regulators) should not be required to understand syntax to be able to assess a scenario-based ADS safety case.* The focus of this article is on the WMG-SDL [12] which proposes having a two-level abstraction plain-text language for scenarios, Level-2 being more detailed than Level-1. WMG-SDL has seen widespread support from by ADS manufacturers and regulators, with the online SafetyPool database [20], [21] supporting searching, sharing, and using scenarios. However, the Level-1 WMG-SDL syntax proposed [12] falls short from being close to a robust natural language description, which is one of the gaps this paper aims to address.

Having a language for specifying scenarios is only one part of the scenario narrative. Writing correct specification requires expertise. The diverse stakeholder audiences that would author, modify and compose scenarios in a SDL makes for a high probability of errors in specification. With increasing attention being given to ADS V&V activities, larger numbers are able to contribute scenarios to the wider Safety Pool^{TM} Scenario Database [21]. The past WMG-SDL [12] did not have the support of scenario authoring tools to enable authors to write and check for correctness of scenarios, nor is there a mechanism to interface WMG-SDL with off-the-shelf ADS testing platforms (virtual or real-life).

This paper addresses these gaps in the following ways:

1) A natural-language SDL that replaces the older Level-1 WMG-SDL [12] to address broader stakeholders. It is then possible to automate the translation of the code-like detailed Level-2 WMG-SDL into the abstract, new Level-1 natural language SDL that is readable for non-verification engineers and non-programmer audiences. The Level-2 WMG-SDL is also now syntax aligned with the ODD Taxonomy defined by the BSI-PAS 1883:2020 and ISO 34503.
2) We develop a framework consisting of parsers and validators for checking the correctness of scenarios written in both Level-1 and Level-2 WMG-SDL. This consists of a domain-specific language tool providing authors with syntax and semantic validation support for writing correct scenarios in WMG-SDL. The tool uses Eclipse Xtext/Xtend [22], [23] to define formal syntax rules and create custom semantic validator checks. The tool can be used as a binary/executable or as part of an Integrated Development Environment (IDE) for online validation.

The rest of this paper is organized as follows: Section II discusses related work. Section III presents our proposal for a natural language SDL, while Section IV presents the architecture we develop using Eclipse Xtext for developing correct scenarios, along with object-oriented extensions for further integration with other off-the-shelf tools. In Section V, we demonstrate the outcomes of using the language parser for preparing scenarios that are correct by specification and with outcomes validated via simulation. Section VI provides conclusions and observations on the work presented herein. Note that the tools and technologies presented in this paper are to be made open-source.

II. Related Work

With increasing automation and system complexities, having complete and correct test scenario specifications is a necessity. Defining a safety-case for ADS's is a fundamental step in safely integrating them into society. The case for using smart-miles to assess safety, where quality trumps quantity, has been repeatedly asserted [3], [6], [24] and scenarios have become the foundation on which the present V&V lifecycle is built. Scenarios define test-cases under which the vehicle is assessed, and an ADS's response to a scenario forms the basis of this safety case. Due to the wide variety of stakeholders involved in the V&V process, four levels of abstraction have been proposed: (1) functional, (2) abstract, (3) logical and (4) concrete [25] . The reason for this is that different stages of the V&V pipeline require analysis performed by experts from varying areas of expertise, each varying in the level of detail with which they describe scenarios [9]. It is also necessary to be able to move between levels of detail, from less detail (functional, representing a higher level of abstraction) to more detail (logical and concrete ready-to-execute scenarios, representing lower levels of abstraction) [5], [12].

These studies [3], [5], [6], [9], [12] elude to the fact that specification writing is an expertise that varies across humans. Imposing a specification language that is very technical and complex to use, would be counter-productive to utilizing scenarios in the V&V pipeline. Writing scenarios in complex languages increases the probability of an human-error, and consequentially incorrect scenarios.

Language development for machine consumption - compilers and translators, have been a subject of study since the late 1940s [26], with advances still being made today. With increasing automation in a multitude of domains, one of the more popular advances in compiler/translator technologies has been in tools for defining Domain-Specific Languages (DSLs). DSLs are languages tailored to specific domains, with gains had in expressiveness and ease of use when compared with general-purpose languages in the broader domain [27]. In particular, the Eclipse Xtext [22] framework is one for developing DSLs that has proven to be powerful, providing a complete parser-validator infrastructure that open-standard Language Server Protocol (LSP) [28].

The Xtext framework provides support for syntax colouring (enhancing readability), syntax and semantic error checking, auto-completion, code folding, viewing code outline structures, and other useful features for domain specific language applications [22]. The contributions presented in this paper are geared towards writing correct, accessible scenarios. In addition to introducing tool support for scenario writing - by creating correctness checker and parsers, we formulate two DSLs, one for an overhauled WMG-SDL Level-1 that allows writing natural-language scenarios, and one for WMG-SDL Level-2. Developing WMG-SDL languages as DSLs in Xtext enables us to treat scenarios as objects, and this opens up WMG-SDL for use in any ADS V&V testing flow in a variety of test environments (simulation or real-world). In the following section, we present our natural-language functional WMG-SDL (Level-1) as it mirrors the more detailed Level-2 scenario specification.

III. Natural Language Scenario Description Language

This section describes our new natural language Level-1 WMG-SDL which allows for writing functional and abstract scenario descriptions. The new language is designed to read more naturally while conveying complex scenario conditions in a well-structured and clear format. WMG-SDL is specified in terms of four elements of the scenario, scenery, scripted actor dynamics, environment and unscripted traffic. Due to a limitations of space, we use examples to demonstrate the changes in syntax, while a complete syntax of the new Level-1 and Level-2 WMG-SDL are available as an online resource [29].

A. Scenery

The *scenery* elements describe stationary items, such as roads and junctions, buildings, as well as state-changing items such as traffic lights, and bridges.

Consider the scenery illustrated in the map in Figure 1. The older Level-1 specification [12] for the scenery is in Figure 2. The same description, written using the new natural language Level-1 SDL specification as shown in Figure 3. The updated syntax is designed to read more naturally.

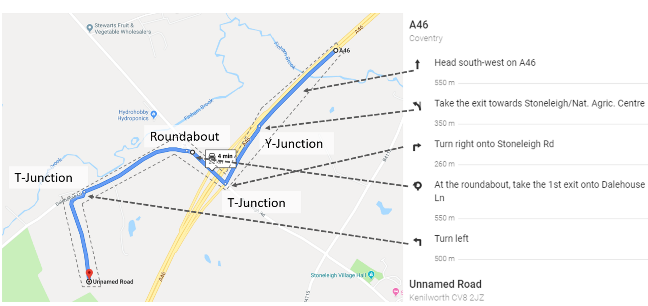

Fig. 1. Road-Junction map of scenery elements around Coventry, *Google Maps, 2022*

B. Scripted Dynamics

In WMG-SDL a component on *dynamic* elements specifies the behaviour of scripted actors. General traffic is defined as a separate component (Section III-D). The scenario does not script or control the behaviour of the Ego vehicle, the Vehicle Under Test (VUT). The dynamics defines the manoeuvre behaviour of other actors (vehicles, bicycles, pedestrians).

Junctions:
[Junction 1] is a [Y-Junction] with connections to [Road 1, Road 2, Road 6]
Connection from [Road 1] to [Road 2] is [To the Left]
Connection from [Road 1] to [Road 6] is [Straight]
[Junction 2] is a [T-Junction] with connections to [Road 2, Road 3, Road 7]
Connection from [Road 2] to [Road 3] is [To the Right]
Connection from [Road 2] to [Road 7] is [To the Left]
Connection from [Road 3] to [Road 7] is [Straight]
[Junction 3] is a [Normal roundabout] with connections to [Road 3, Road 4, Road 8]
Connection from [Road 3] to [Road 4] is [To the Left]
Connection from [Road 3] to [Road 8] is [Straight]
Connection from [Road 4] to [Road 8] is [To the Left]
[Junction 4] is a [T-Junction] with connections to [Road 4, Road 5, Road 9]
Connection from [Road 4] to [Road 5] is [To the Left]
Connection from [Road 4] to [Road 9] is [Straight]
Connection from [Road 5] to [Road 9] is [To the Left]
Roads:
[Road 1, Road 6] are [Radial roads] with [Straight] geometries
[Road 2, Road 3, Road 6, Road 7, Road 8, Road 9] are [Distributor roads] with [Straight] geometries
[Road 4] is a [Distributor road] with a [Curved Right then Straight] geometry
[Road 5] is a [Minor road] with a [Straight] geometry

Fig. 2. Past WMG-SDL Level-1 Scenery Specification for Fig. 1

There is a Y-Junction, Junction1, which has connections with Road1, Road2 and Road6. Road1 to Road2 is to the left. Road1 to Road6 is straight ahead.
There is a T-Junction, Junction2, which has connections with Road2, Road3 and Road7. Road2 to Road3 is to the right. Road2 to Road7 is to the left. Road3 to Road7 is straight ahead.
There is a normal roundabout, Junction3, which has connections with Road3, Road4 and Road8. Road3 to Road4 is to the left. Road3 to Road8 is straight ahead. Road4 to Road8 is to the left.
There is a T-Junction , Junction4, which has connections with Road4, Road5 and Road9. Road4 to Road5 is to the left. Road4 to Road9 is straight ahead. Road5 to Road9 is to the left.
Road1 and Road6 are straight, radial roads.
Road2, Road3, Road6, Road7, Road8 and Road9 are straight, distributor roads.
Road4 is a curved, distributor road. Road5 is a straight, minor road.

Fig. 3. Natural Language WMG-SDL Level-1 Scenery Specification for Fig. 1

These actors' behaviours may cause the VUT to react, and similarly non-VUT actors' behaviours may be triggered by observed behaviours of the VUT.

The dynamic elements are a collection of mutually exclusive blocks, each block containing a set of synchronous behaviours as shown in Figure 4. The mutually exclusive blocks of synchronous behaviours may be triggered asynchronously. In a behaviour block, actors may synchronize behaviours in phases that run parallel in time, or serially in time. The execution of a phase is conditioned on an invariant that must hold true while the phase is active.

To demonstrate the proposed natural language syntax, we use a scenario of a pedestrian crossing the road in front of a vehicle. The past Level-1 syntax, shown in Figure 5, is replaced by the new Level-1 syntax shown in Figure 6.

C. Environment

The environment component defines the weather, particulates, illumination, connectivity capabilities and time of day. The proposed natural language Level-1 SDM syntax for the environment is shown in Figure 8, while the older syntax is shown in Figure 7.

D. Unscripted Traffic

The traffic element of a scenario defines non-scripted, but intelligently controlled traffic. The traffic specification must specify the traffic density (number of expected vehicles in a road section), volume (vehicles generated per unit time),

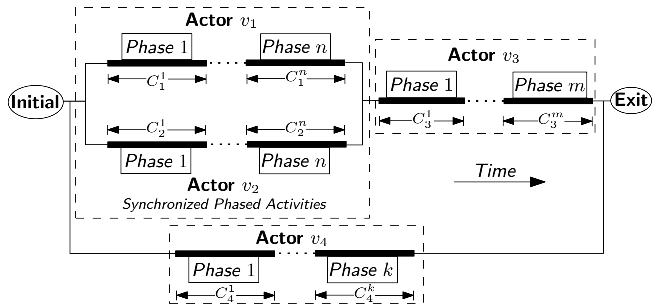

Fig. 4. SDL multi-actor dynamics synchronized by condition-controlled phases, or sequenced in series. Multiple asynchronous blocks of a similar form may be specified. A block's entry and exit are condition-controlled. Actors v_1 and v_2 have manoeuvres/actions/activities that are synchronized as phases running in parallel. Actor v_3 is conditioned to begin its first phase only after the last (n^{th}) phase of v_1 and v_2 are complete. Actor v_4 executes asynchronously of the others.

INITIAL: Vehicle [V1] in [Road 1]
AND Pedestrian [P1] in [Road 1 Pedestrian sidewalk]
WHEN: [V1] is [Going ahead]:
[P1] [Walks towards] [V1] with [Constant] speed at [Front Right] position
[P1] [Walks cross] [V1] with [Constant] speed at [Front] position
[P1] [Walks away] from [V1] with [Constant] speed at [Front left] position
END: [P1] in [Road 1 Pedestrian sidewalk] AND at [Front left] position to [V1]

Fig. 5. Past WMG-SDL Level-1 Dynamics Specification

average speed, source and sink. An example of the contemporary syntax for Level-1 description of traffic between roads R_1 and R_2 is shown in Figure 9.

We propose that traffic specification in Level-1 natural language descriptions be a suffix to the description of the scenery. For instance, for Road1 (R_1) (and similarly R_2), we would state:

Road1 is a straight, distributor road with moderate traffic.

Here the suffix *with moderate traffic* is appended to the specification of the road's scenery element.

The impact of the new WMG-SDL Level-1 syntax on scenario readability is more prominent in descriptions of scripted dynamic and environment elements. In descriptions of the scenery, programmatic structures have been replaced by a more natural to read sentence structure, while retaining the information detail present in the past version of Level-1

There is 1 vehicle, Vehicle1. There is 1 pedestrian, Pedestrian1. Vehicle1 and Pedestrian1 are in Road1. When Vehicle1 is going ahead, Pedestrian1 walks towards Vehicle1 at its front right, walks across in front of Vehicle1 and then walks away.

Fig. 6. Natural Language WMG-SDL Level-1 Dynamics Specification

DO: Environment [ENV1] as:
Wind [Moderate breeze]
Clouds [Heavy]
Particulate condition [Fog]
Rainfall [Moderate rain]
Time [Day]
Illumination [Daylight]

Fig. 7. Past WMG-SDL Level-1 Environment Specification

During the day, there is a light, cloud covered sky, with mist and fog, rain, and a moderate breeze.

Fig. 8. Natural Language WMG-SDL Level-1 Environment Specification

INITIAL: Centroid [C1] in [R2] AND centroid [C2] in [R1]
DO: Traffic [Traffic 1] as:
- Direction [C1] to [C2]
- Density [Moderate]
- Average traffic speed [Moderate]

AND DO: Traffic [Traffic 2] as:
- Direction [C2] to [C1]
- Density [Moderate]
- Average traffic speed [Moderate]

Fig. 9. Past WMG-SDL Level-1 traffic specification

SDL. For unscripted traffic, the earlier description of traffic in terms of centroids has been replaced. Traffic specifications are now coupled with the description of the roads in the scenery.

In the following section, we present our architecture for assisting scenario stakeholders in reading and writing scenarios.

IV. Extensible Architecture for Developing Correct Scenarios

This section presents our DSL architecture, containing tools built to support developing syntactically and semantically correct scenarios. The architecture is based on the Eclipse Xtext framework [22], which allows us to expose and object-oriented scenario structure for Level-1 and Level-2 WMG-SDL scenarios. This enables automation extensions to be added to the tool-suite that use and manipulate the scenario object. Some example purposes of developing such extensions may include, translating WMG-SDL scenarios into other scenario specifications, developing instructions for testing (virtually or on a physical test platform), generating new scenarios, assessing scenario coverage, comparisons against ADS ODDs, and so on.

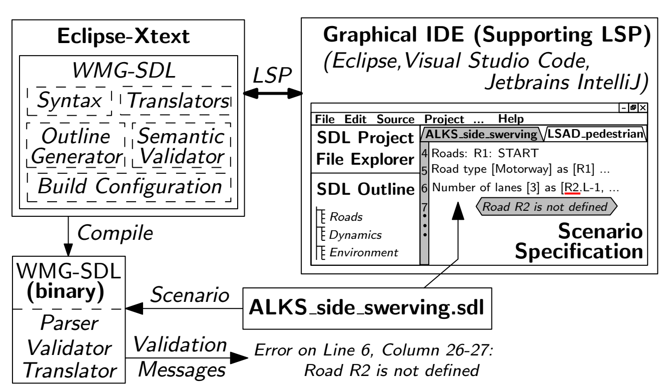

Fig. 10. DSL Architecture for Scenarios in WMG-SDL using Xtext

Figure 10 presents the proposed DSL architecture for developing scenarios. The architecture consists of: (1) WMG-SDL Level-1, Level-2 Syntax Parsers, (2) Scenario Outline

Generator, (3) Semantic Error-Checking Validators, (4) Extensions/Translators and a (5) Build Configuration. Detailed descriptions regarding each component are as follows.

A. WMG-SDL Syntax Parsers

The fabric of a parser consists of terminals (keywords, integers, decimal numbers, and strings) and grammar syntax rules (defining the grammar of a scenario, expressing how a scenario is developed using terminals and other grammar rules). The parsers for natural language and Level-2 WMG-SDLs are defined in Xtext.

We implement syntax rules for the proposed natural-language SDL, as defined in Section III. For Level-2 WMG-SDL, we retain the syntactic structure of contemporary WMG-SDL, making only minor updates to align its vocabulary with the ISO 34503 and BSI-PAS 1883.

The advantage of using Xtext is that it allows us to piggyback an object-oriented class structure atop the grammar definition.

A snippet of an Xtext rule for roads in Level-2 is as follows:

RoadElement : *null*?='None' |
id=**R_ID** ':' 'START' ('['(*startJunction*=**ID**|*startRoad*=**R_ID**)']')?
'Road' 'type' '['*roadType*=**RoadType**']' 'as' '['*roadAltID*=**R_ID**']'
'with' 'zone' 'as' '['*zoneType*=**ZoneType**']' 'AND'
'speed ' 'limit' 'of' '['(*speedLimit*=**DOUBLE**|*noSpeedLimit*?='N/A')']'
'in' ('a'|'an') '['*environmentType*=**EnvironmentType**']' 'environment'
'with' 'Number' 'of' 'lanes' *laneDescription*=**LaneDescription**

The rule above implies a class structure for class **RoadElement** with class attributes defined by those marked in italics, while the attribute data-types (or class-types) are correspondingly marked in bold. The data-types represent terminal rules (such as **ID** or **R_ID** representing string names for general strings or road names, or **DOUBLE** representing decimal numbers), or rules to produce further specification (such as **RoadType**, **ZoneType**, **EnvironmentType**, **LaneDescription**). Hence, when a scenario is parsed, an object is created from the scenario with named-fields that can be processed for various purposes as detailed in later sections.

Such rules are then used in the parser to check the syntactic correctness of a scenario (see Figure 12). We discuss semantic validation in Section IV-C. Detailed grammar rules for the proposed natural language WMG-SDL as well as the revised Level-2 grammar are available as an online resource [29].

B. Scenario Outline Generator

Scenarios for ADS can become complex to specify in any language. This complexity could be due to the length of the specification, or due to the complex interactions and attribute values of scenario components. It is therefore important to, (1) be able to navigate scenarios, and (2) have a high-level summary view of the scenario - with the ability to hide information that may not be relevant to the moment.

A collapsible outline view of the scenario addresses both these needs. The Xtext framework has allowed us to implement a tree-structure hierarchical view of the scenario for both WMG-SDL levels (as shown in the bottom left of Figures 12 and 13). Custom-definitions have been implemented to provide summary outline views that are consistent with the component structure of WMG-SDL, partitioning the scenario at the highest level according to scenery, dynamics, environment and traffic specification. For complex components (such as scenery and dynamics) the outline further breaks this down into an organized structure of roads, junctions, manoeuvres, phased activities, as may be the requirement. Synchronous and asynchronous actor actions are also evident from the outline structure (see Figure 11), allowing authors to navigate the hierarchy as deep as necessary, expanding or collapsing components as they require.

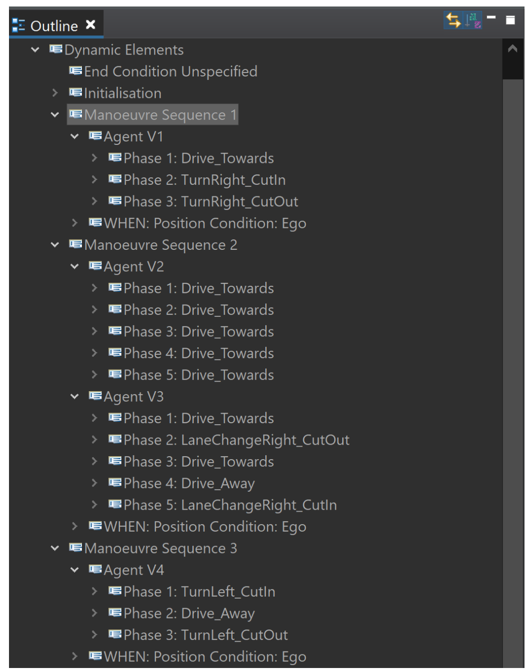

Fig. 11. Scenario Outline Summary describing the outline for a scenario having three asynchronous sets of manoeuvres.

C. Semantic Error-Checking Validators

Beyond following syntactic rules, scenario correctness also requires that the scenario be semantically sound (meaningfully correct). For instance, a scenery element (such as a road) isn't defined in the scenario, but is used while positioning a vehicle, or perhaps a measurement is made relative to the unknown element; this would represent a semantic error. Such errors must be brought to the notice of the author in a manner that communicates the error and provides guidance on how to fix it.

We develop detailed semantic validation checks that assess the correctness of junctions and roads, ensuring that they are correctly defined with no missing elements; we compare the scenery with the dynamics ensuring that the dynamic specifications use correct scenery elements, and highlight any issues that result in the scenario being incorrect.

An example of a semantic error highlighted during level-2 scenario development is shown in Figure 13.

D. Scenario Extensions/Translators

The SDL class definition that is generated from our grammar definition, described in Section IV-A, is used during parsing to parse the scenario into an SDL object instance. This object instance may be used to extend the functionality of the tool and interface with other tools in the V&V pipeline, such as for scenario generation, translation or execution.

For instance, we have developed an extension for translating WMG-SDL Level-2 scenario specifications into ASAM OpenSCENARIO (OSC) and OpenDRIVE (ODR) scenario specification standards. OSC and ODR languages are XML-based languages, making them intrinsically machine readable, and are hence supported by simulation tools such as esmini [30]. However, scenarios in OSC and ODR quickly, even for the simplest of scenarios, become complex and lengthy to develop. They are therefore difficult to read. By enabling a translation from WMG-SDL to OSC/ODR, WMG-SDL inherits the benefits afforded by tools that support OSC/ODR. This enables scenarios to be written in easy to read, extensible, navigable, tool-supported WMG-SDL, with the benefits of simulation and execution support of OSC/ODR.

The scenario object structure developed enables developers to create tools (for generating, translating or executing scenarios) for use in the V&V pipeline.

E. Build Configuration for Executability

The architecture and tools developed may be used in a graphical integrated development environment (IDE), or may be executed as a standalone Java binary. In order to build a Java binary, we develop a configuration that orchestrates how syntax and semantic validation errors are displayed and how any extensions are used (for instance generating scenarios in ASAM OpenX standards).

F. Bringing it all together

Some of the features described above are visible in Figures 12 and 13.The scenario outline pane on the bottom left of the IDE window provides easy navigation through the scenario, along with a high-level hierarchical scenario breakdown, further assisting scenario inspection.

It is our intention to make the scenario tool-suite (including codified grammars, parsers-validators and supporting components - translators, outline generators) open-source.

V. Language Parser Case Studies

We use our scenario framework to develop scenarios in the the UN Regulation for Automated Lane Keeping Systems (ALKS) [31], which describes the necessary functional requirements to be fulfilled for the approval of an ALKS. In this article we present a selection of the scenarios in the natural language Level-1 WMG-SDL and their corresponding Level-2 descriptions at the logical scenario level. The

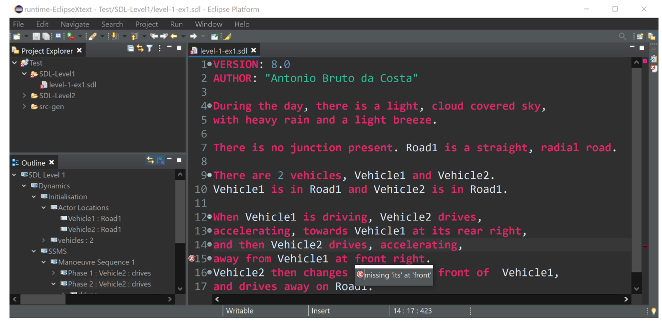

Fig. 12. Graphical IDE (Eclipse) for developing natural-language WMG-SDL Level-1 scenarios. A syntactic error is detected on line 15 of the natural-language scenario, with a suggested fix provided.

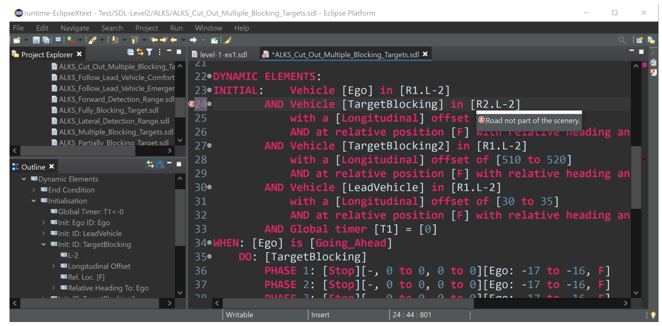

Fig. 13. Graphical IDE (Eclipse) for developing logical WMG-SDL Level-2 detailed scenarios. A semantic error is detected on line 24 of the scenario, with a semantic cause provided, indicating use of road label that is not defined in the scenery.

scenarios are developed with the intention of assessing the ADS dynamic driving task under different conditions.

A. Field of View - Lateral Range Detection

This scenario evaluates the ADS's capabilities to detect another road user beside it, within its lateral detection area, up to at least the full width of the adjacent lane (clause 4.6 [31]). Figure 14 presents the scenario specified in natural language SDL, while Figure 15 specifies the same scenario in Level-2 SDL.

During the day, there is a light, clear sky and a light breeze. There is no junction present.
Road1 is a straight, motorway. Also, Road1 has broken lane markings. There are 2 vehicles, Ego and SideVehicle. Ego and SideVehicle are in Road1. When Ego is driving ahead, SideVehicle changes lane left towards Ego at its right, and drives ahead.
SideVehicle then changes lane right ahead of Ego, and drives away.

Fig. 14. Field of View - Lateral Range Detection: Natural Language SDL

B. Multiple Blocking Targets

This scenario evaluates the ADS's capabilities to avoid a collision with a road user or object blocking the lane it is in, while its driving speed is up to the maximum specified speed of the ADS. This specific scenario uses multiple consecutive obstacles blocking the lane (clause 4.2.2 (g) [31]). Figure 16

SCENERY ELEMENTS:
DO: Map - roads and junctions network [Network1] as:
Junctions: None
Roads:
R1: START
Road type [Motorway] as [R1] with zone as [N/A] AND speed limit of [70] in a [Rural] environment with
Number of lanes [3] as [R1.L-1, R1.L-2, R1.L-3]
Road traffic direction [Right-handed]
Lane type [Traffic lane]
Lane markings [Broken line]
Horizontal road geometry [Straight]
Vertical road geometry [Level plane]
Transverse road geometry [Divided] with [No] roadside feature
Length [9000 to 11000] AND Lane width [3.4 to 3.7]
END

DYNAMIC ELEMENTS:
INITIAL: Vehicle [Ego] in [R1.L-2]
AND Vehicle [SideVehicle] in [R1.L-3] with a [Lateral] offset of [-1.75 to -1.75]
AND at relative position [SR] with relative heading angle [0 to 5] to [Ego]
AND Global timer [T1] = [0] AND Local timer [t1] = [0]
WHEN: [Ego] is [Going_Ahead]
DO: [SideVehicle]
PHASE 1: [LaneChangeLeft_Towards][-, 15 to 16, 0 to 0][Ego: -0.1 to 0.1, SR]
WHILE:[SideVehicle] [Lateral] offset to [R1.L-3] != [0]
PHASE 2: [LaneChangeLeft_Towards][-, 15 to 16, 0 to 0][Ego: -0.1 to 0.1, SR]
WHILE:[SideVehicle] [Lateral] offset to [R1.L-3] < [0.5]
PHASE 3: [Drive_Away] [-, 15 to 16, 0 to 0][Ego: -0.1 to 0.1, SR]
WHILE: [t1] < [5]
PHASE 4: [LaneChangeRight_Away][-, 15 to 16, 0 to 0] [Ego: -0.1 to 0.1, SR]
WHILE: [SideVehicle] [Lateral] offset to [R1.L-3] != [0]
PHASE 5: [Drive_Away] [-, 15 to 16, 0 to 0][Ego: -0.1 to 0.1, SR]
END: [T1] == [40]

ENVIRONMENT ELEMENTS:
DO: [Env1]
Wind [0 to 0.2] Cloudiness [0 to 1] Particulates [None] Rainfall [None: N/A]
Snowfall [None: N/A] Time of the day [03:00 to 06:00]
Illumination [Day] with [Sun] as light source at [10 to 30] degree elevation AND [F] position

Fig. 15. Field of View - Lateral Range Detection: Level-2 SDL

presents the scenario specified in natural language SDL, while Figure 17 specifies the same scenario in Level-2 SDL.

All the scenarios presented have been validated by the parser. The intended scenario behaviours are then verified using our OSC/ODR translator extension – by simulating the translated OSC/ODR scenario in esmini, and by examining the simulation log to ensure that the WMG-SDL events occur as expected and are timed correctly. The latter is used to examine synchronization.

During the day, there is a light, clear sky and a light breeze. There is no junction present.
Road1 is a straight, motorway. Also, Road1 has broken lane markings.
There are 3 vehicles, Ego, TargetBlocking and TargetBlocking2. Ego, TargetBlocking and TargetBlocking2 are in Road1.
When Ego is driving ahead, TargetBlocking is stopped ahead of Ego, and TargetBlocking2 is stopped ahead of TargetBlocking.

Fig. 16. Multiple Blocking Targets: Natural Language SDL

VI. Conclusions

This paper recognizes scenario-based testing to test safety performance and driving characteristics of ADS's over human drivers. Scenarios must be available in different forms to address the variety of ADS stakeholders. Authors of scenarios require assistance in writing correct scenarios. Correctness of scenario specifications is multi-faceted. We provide a two-pronged approach. We update the Level-1 WMG-SDL syntax to read more naturally, which intrinsically ensures the specifications reflect the intended scenario. We also provide a parser to check scenario syntax and a semantic validator to ensure that a scenario is semantically valid. The parser-validator architecture works with open-standard Language Server Protocol (LSP), enabling our tool to communicate with graphical integrated development environments (IDEs) to visually manage and develop scenario projects. Scenarios in our framework are parsed into objects, which can then be used by tool developers to create extensions for scenario generation, translation and execution. As a motivating example, the object-interface extension mechanism has enabled us to develop a translation of WMG-SDL Level-2 scenarios into concrete ASAM OpenX (OpenScenario and OpenDrive) executable scenarios. The regulation on Automated Lane-Keeping Systems (ALKS) has been used as a case study to develop scenarios. It is intended that the parser tool-chain be made open-source to enable wider use of the tool.

DYNAMIC ELEMENTS:
INITIAL: Vehicle [Ego] in [R1.L-2]
AND Vehicle [TargetBlocking] in [R1.L-2]
with a [Longitudinal] offset of [495 to 505]
AND at relative position [F] with relative heading angle [0 to 5] to [Ego]
AND Vehicle [TargetBlocking2] in [R1.L-2]
with a [Longitudinal] offset of [510 to 520]
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: [TargetBlocking]
PHASE 1: [Stop][-, 0 to 0, 0 to 0][Ego: -55 to -65, F]
AND: [TargetBlocking2]
PHASE 1: [Stop][-, 0 to 0, 0 to 0][Ego: -55 to -65, F]
END: [T1] == [40]

Fig. 17. Multiple Blocking Targets: Level-2 SDL. Note that Scenery and Environment components of the scenario have been omitted since they are identical to those in Fig. 15.

Acknowledgement

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.

References

[1] 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.

[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, p. 35–54, 2016.

[3] 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.

[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] 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 Intelligent Vehicles Symp., Proc.*, vol. 2019-June, pp. 2383–2390, 2019.
[6] 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 Int. Conf. on Intelligent Transportation Systems*, pp. 982–988, 2015.
[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," in *2021 IEEE Int. Intelligent Transportation Systems Conf. (ITSC)*, 2021, pp. 312–319.
[8] 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.
[9] 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.
[10] 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 Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.
[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.
[12] 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.
[13] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: A Language for Scenario Specification and Scene Generation," in *Proc. of the 40th ACM SIGPLAN Conf. on Programming Language Design and Implementation*. ACM, 2019, pp. 63–78.
[14] Fortellix, "Open M-SDL: Measurable Scenario Description Language," 2022. [Online]. Available: https://www.foretellix.com/open-language/
[15] R. Queiroz, T. Berger, and K. Czarnecki, "GeoScenario: An Open DSL for Autonomous Driving Scenario Representation," 2019. [Online]. Available: https://geoscenario.readthedocs.io/
[16] ASAM, "ASAM OpenSCENARIO® Standard," 2021. [Online]. Available: https://www.asam.net/standards/detail/openscenario/
[17] ——, "ASAM OpenDRIVE® Standard," 2021. [Online]. Available: https://www.asam.net/standards/detail/opendrive/
[18] I. Automotive, "CarMaker," 2022. [Online]. Available: https://ipg-automotive.com/en/products-solutions/software/carmaker/
[19] I. The MathWorks, "MATLAB Automated Driving Toolbox," 2022.
[20] U. o. W. WMG, "World's largest public scenario database for testing and assuring safe Autonomous Vehicle deployments," 2021. [Online]. Available: https://warwick.ac.uk/newsandevents/pressreleases/worlds_largest_public/
[21] ——, "Safety Pool^{TM} Scenario Database," 2021. [Online]. Available: https://www.safetypool.ai/database
[22] Eclipse, "Language Engineering for Everyone," 2021. [Online]. Available: https://www.eclipse.org/Xtext/
[23] L. Bettini, *Implementing Domain Specific Languages with Xtext and Xtend - Second Edition (2nd. ed.)*. Packt Publishing, 2016.
[24] 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 Conf. (ITSC)*, 2019, pp. 1961–1967.
[25] 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.
[26] J. Hennessy and M. Ganapathi, "Advances in compiler technology," in *Ann. Rev. Comput. Sci.*, vol. 1, 1986, pp. 83–106.
[27] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.*, vol. 37, no. 4, p. 316–344, dec 2005.
[28] Microsoft, "Language Standard Protocol," 2022. [Online]. Available: https://microsoft.github.io/language-server-protocol/
[29] U. o. W. WMG, "Proposed Two-Level Scenario Description Language," 2022. [Online]. Available: https://docs.safetypooldb.ai/docs/sdl
[30] E. Knabe *et al.*, "Environment Simulator Minimalistic (esmini)," 2022. [Online]. Available: https://github.com/esmini/esmini
[31] 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