Translating Automated Vehicle Test Scenario Specifications Between Scenario Languages: Learnings and Challenges

Antonio Anastasio Bruto da Costa, Patrick Irvine, Xizhe Zhang, Siddartha Khastgir and Paul Jennings

WMG, University of Warwick, Coventry, CV4 7AL, United Kingdom,
e-mail: {antoniobruto-da-costa, patrick.irvine, jason.zhang, s.khastgir.1, paul.jennings}@warwick.ac.uk

Abstract - *The Verification and Validation (V&V) lifecycle for an Automated Driving System (ADS) has evolved to use scenarios as its basis for evaluating functional correctness and its safety. A scenario describes scenery (road and junction layout), environmental conditions, and behaviour of road-actors (vehicles or pedestrians). Due to the variety of V&V stakeholders, scenario descriptions must be accessible, easy to specify, readable, and executable in simulation. The existing standards for scenario specification are the ASAM OpenX (OpenScenario and OpenDrive) languages which use the Extensible Markup Language (XML). The inherent structure of XML affects ease of specification and readability; nonetheless, they have wide simulation tool support. The two-level WMG-SDL scenario concept addresses the problem of ease of specification and readability, but scenarios written therein are (until now) not compatible with ASAM OpenX languages. This article bridges this gap by providing a methodology and tool for translating scenarios in WMG-SDL to OpenX equivalents. The tool uses the Eclipse xText framework for parsing WMG-SDL and implementing the scenario translator. We discuss how different syntactic elements in WMG-SDL are translated into OpenX and associated challenges. The translation is applied to benchmark scenario sets, (1) Automated Lane Keeping System (ALKS) (UNECE Reg. 157) scenarios, and (2) Low-Speed Automated Driving (LSAD) (ISO 22737) scenarios.*

Keywords: *Automated Driving Systems, Scenarios, Translators*

1. Introduction

With increasing automation and system complexities in Automated Driving Systems (ADSs), having complete and correct test scenario specifications is a necessity. Defining a robust safety-case for ADSs is a fundamental step in their safe introduction into society. The case for using smart-miles to assess safety, where quality of scenarios trumps quantity of scenarios, has been repeatedly asserted (Gangopadhyay, et al., 2019; Khastgir, et al., 2021; Ulbrich, et al., 2015) and scenarios have become the foundation on which the present Verification & Validation (V&V) lifecycle is built. Scenarios define conditions under which the vehicle is assessed, and an ADS's response to a scenario provides evidence for the safety case. Due to the wide variety of stakeholders involved in the V&V process, four levels of abstraction for scenarios have been proposed: (1) functional, (2) abstract, (3) logical, and (4) concrete (Neurohr, et al., 2021). 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 (Menzel, Bagschik, and Maurer, 2018). 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) (Menzel, et al., 2019; Zhang, Khastgir, and Jennings, 2020).

These studies (Khastgir, et al., 2021; Menzel, et al., 2019; Menzel, Bagschik, and Maurer, 2018; Ulbrich, et al., 2015; Zhang, Khastgir, and Jennings, 2020) 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 throughout the V&V pipeline. Further, the greater the complexity of scenarios used in testing, the greater the chances of having complex specifications, and consequentially incorrect scenarios.

Scenario Description Languages (SDLs) for ADSs form the basis for assessing the function of ADSs. A scenario expresses the dynamic, environmental, traffic, and road-junction scenery in which an ADS is required to operate. While there exist numerous SDLs, the WMG-SDL (Zhang, Khastgir, and Jennings, 2020) has become popular, with the online Safety Pool™ Scenario Database supporting searching, sharing, and using scenarios. However, due to the wide variety of scenario description languages, and the increasing need for simulation tools to support them, the ASAM Standards body developed the OpenX languages, OpenScenario (OSC) (ASAM, 2021b) and OpenDrive (ODR) (ASAM, 2021a), for describing behaviour and scenery respectively. In their present form, the languages, OSC 1.x and ODR 1.6, are XML-based. Furthermore, a newer version of OpenScenario is currently being developed, OSC 2.0. A number of simulation tools support a subset of these languages, with the open-source "Environment Simulator Minimalistic (*esmini*)" tool (Knabe, et al., 2022) providing the widest support for the OpenX languages. While XML-based languages are well-structured and machine supported, their inherent structure affects ease of readability and development of scenarios.

On the other hand, the WMG-SDL has two specification forms, an abstract level 1 specification in which scenarios are expressed in structured natural language, but lack the detail required for simulation, and a (logical scenario) level 2 specification which is concise and readable and allows for detailed expressions of scenarios. Furthermore, WMG-SDL 8.0 (WMG, 2022) allows the scenery and behaviour descriptions in a scenario to be expressed together, which makes cross-referencing easier. With the increasing use of the description-friendly language WMG-SDL, its usability and support in a variety of simulation tools is critical for wider adoption. We use the detailed logical scenario WMG-SDL level 2 specification for translation to OpenX scenarios. In this article, when we refer to WMG-SDL, the reader should read this as the level 2 WMG-SDL language.

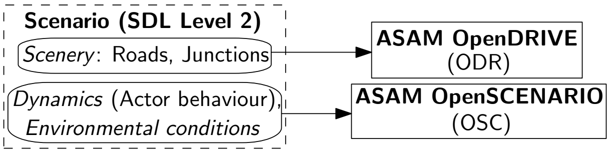


Figure 1: A singular WMG-SDL Scenario Description is translated into two OpenX specifications, an OpenSCENARIO and an OpenDRIVE specification.

To bridge the gap between WMG-SDLs ease of specification and readability, and simulation support for OpenX languages, this work proposes a methodology for translating scenarios specified in WMG-SDL Level 2 to OpenX equivalents, as shown in Figure 1. Due to the differences in the execution semantics of both languages, this work explores how such a bridge between the two languages may be formed.

The translator from WMG-SDL to OpenX is written using Eclipse Xtext (Eclipse, 2021). Using Xtext, we first define a parser for WMG-SDL Level 2. A WMG-SDL scenario specification is parsed into an object structure representing the language's syntax tree. This tree is then explored to construct the semantic equivalent OSC and ODR scenario specifications. Developing WMG-SDL languages in Xtext enables us to treat scenarios as objects, and this opens up WMG-SDL for use in any ADS V&V testing flow, for translators to be written to other scenario specification languages for a variety of test environments (simulation or real-world).

This article is structured as follows. Section 2 introduces related work on scenario languages. Section 3 describes the translation methodology for scenarios in WMG-SDL into scenarios in ASAM OpenX languages. Section 4 presents our results. We discuss our findings in Section 5, and provide concluding remarks in Section 6.

2. Related Work

The quality of testing has become key in the effort of developing safe and correct ADSs. This philosphy deviates from the earlier belief that to realize a 20% quality improvement over human drivers, 11 billion training and testing miles would be required (Kalra and Paddock, 2016). This philosophy has rightly evolved to stress on quality of miles driven over quantity (Khastgir, et al., 2021). The operating conditions (environmental, road, and other dynamic conditions) under which an ADS is designed to operate is termed as its Operational Design Domain (ODD) (Society of Automotive Engineers, 2021). In the V&V life-cycle for an ADS, scenarios are the principal assets used to identify failures in an ADS (Menzel, et al., 2019; Ulbrich, et al., 2015), and may be defined on the basis of the ADS's ODD. Ulbrich et al (Ulbrich, et al., 2015) defines 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 part of V&V approaches (De Gelder, et al., 2022; Esenturk, et al., 2022; Fremont, et al., 2020; Menzel, Bagschik, and Maurer, 2018), a number of scenario description languages have been developed. Some of the more well known among these include the two-level abstraction WMG-SDL (Zhang, Khastgir, and Jennings, 2020), Scenic (Fremont, et al., 2019), Fortellix M-SDL (Fortellix, 2022), GeoScenario (Queiroz, Berger, and Czarnecki, 2019) and ASAM OpenX Standards (ASAM, 2021a; 2021b). Among these, WMG-SDL is widely supported in the online Safety Pool™ Scenario Database (WMG, 2021a; 2021b) providing searching and sharing functionalities for WMG-SDL scenarios. On the other hand, the ASAM OpenX scenario specifications find wide support among simulation tools.

3. Methodology

Our methodology is twofold. In the first stage, we parse the SDL into an object structure that can be searched and manipulated. In the second stage, a mapping is identified, linking WMG-SDL components to OpenX language components. The core challenge in identifying the syntax structures in ODR/OSC required to implement WMG-SDL grammar components, is achieving semantic equivalence. Furthermore, we were restricted to using only those syntactic structures of OSC/ODR that were supported by the simulator in use (esmini). Other off-the-shelf scenario simulators also provide selective support for OSC/ODR. Hence, the syntax mapping process is iterative, with every translation iteration being tested to ensure that the actions specified in WMG-SDL are manifesting as expected in a simulation of the translated OSC/ODR specifications. The WMG-SDL specification has four core components:

1. Scenery (Roads and Junctions)
2. Dynamics (Actors, their positions, and actions)
3. Environment (Time of Day, Weather, Cloud state, Illumination conditions, etc.)
4. Unscripted Traffic (Location and Density)

Component 1 (Scenery) is translated into an ODR specification, while Components 2 to 4 are translated into an OSC specification. In this section, we describe how the different components listed above are translated while discussing the challenges we overcame. The OSC specification is dependent on the scenery naming convention used in the ODR. In WMG-SDL, scenery and dynamics are described in the same scenario specification, which is not the case with OSC and ODR specifications. For ease of reading, in each component's translation, we will refer to both OSC and ODR and present their dependencies.

The translation of the scenery in WMG-SDL into an ODR specification involves a geometrical translation of the WMG-SDL's descriptive road-junction network specification into concrete coordinates in ODR. This involves interpreting specifications in relative and functional terms in WMG-SDL (such as road X has three segments, one straight of length l1, one curved of length l2 curvature c2, and one straight of length l3) into co-ordinates specifying the x-y-z and heading angles of each road segment along with how they are connected.

There are three core algorithmic challenges in translating WMG-SDL into OSC. There are, (1) Overcoming the difference (and ease) in the specification structure for synchronous/asynchronous actions, (2) Translating high-level abstract manoeuvres in WMG-SDL into a series of sub-manoeuvres in OSC, and (3) The mechanism of indexing roads and lanes that introduce challenges in implementing vehicle turning manoeuvres.

We overcome these challenges in our work and describe exact translations where possible, and workarounds where an exact translation is not possible, and therefore the potential gaps and difficulties in writing scenario specifications in OSC/ODR. In some instances where the treatment of scenario elements between OSC/ODR and WMG-SDL are identical, we use syntactic translations (for instance, for aspects of the traffic and environment). For more complex components in WMG-SDL (such as scenery and dynamics), where there is not a one-to-one mapping between syntactic elements in both languages, the translations are semantic. This means that we use OSC/ODR syntax components to build manoeuvre dynamics and roads/junctions in a way that allows us to uniformly translate scenarios in WMG-SDL.

We implement the translation algorithm using xText and xtend. A parser for WMG-SDL, implemented in xText, parses the WMG-SDL into an object structure which is then explored systematically to construct the OSC/ODR specification. The construction of the OSC/ODR specifications is implemented in xtend. The toolflow required to translate WMG-SDL into OSC/ODR specifications can be executed as a binary (a Java JAR) or in the Eclipse IDE.

This section is structured based on the WMG-SDL scenario specification syntax. The WMG-SDL syntax partitions a scenario into scenery, dynamics, environment, and traffic. As shown in Figure 1, the scenery is translated into an ODR specification, while the latter three components are grouped into an OSC specification. We use extracts from scenario specifications in WMG-SDL and ODR or OSC (as required) to demonstrate the translation and associated challenges. The units of measurement for WMG-SDL mirror that of ASAM OpenX standard (ASAM, 2021a) unless otherwise specified. We use the term *Ego* to refer to the vehicle under test (the ADS).

3.1. Scenery

The scenery is comprised of descriptions for roads and junctions in the scenario. Roads and junctions are defined in the ASAM ODR specification and referenced in the ASAM OSC specification. This section describes the translation of scenery specifications in WMG-SDL into ODR.

3.1.1. Roads

In the ODR, a *Road* element defines the geometry and layout of the road. This includes the road's length, curvature, number of lanes, etc. The road is detailed using sub-elements such as *type*, *planView*, *lateralProfile*, *elevationProfile* and *lanes*. The type element describes the road material and the speed limit of the road. The ODR *planView*, *lateralProfile*, and *elevationProfile* elements describe the geometry of the roads. The *lanes* element lists the different lanes that form part of the roads.

Roads and lanes in ODR are numbered. Given the direction of the road, the lanes to the right of the centerline are numbered starting with a -1, with negative numbers, increasing negatively towards the outermost right lane. Similarly, to the left of the centerline, lanes are numbered with a positive 1, with positive numbers, incrementing in value up to the leftmost lane. This numbering is independent of traffic direction and depends solely on the direction of the road's centerline.

The translation of the scenery in WMG-SDL into ODR requires us to consider the syntax and semantics of OSC manoeuvres within the scenery.

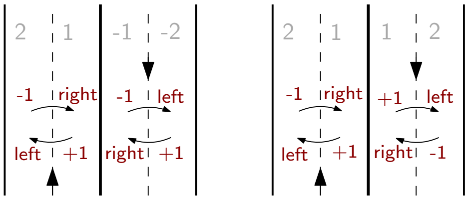
In OSC, lane changes are applied to an actor's position by either indicating a lane displacement relative to the vehicle's current position or indicating a destination lane ID to change position to. During a lane-change manoeuvre it is not possible at runtime to know where the vehicle is (its lane) to apriori (before simulation) specify a destination lane (in terms of an absolute lane position) for the vehicle. Hence lane movements must be defined using a relative displacement. Furthermore, in OSC left and right lane change movements have different displacements, not to mention that this is also dependent on the direction of traffic. For a vehicle moving in the direction of the centerline, with right-handed traffic, positive lane changes indicate a left lane change whereas negative values indicate a right lane change. For a vehicle moving against the direction of the centerline, positive lane changes indicate a right lane change whereas negative values indicate a left lane change. Left-handed traffic would then mirror these lane change actions. Figure 2(a) depicts OSC lane change manoeuvres concerning traffic direction for left-handed driving.

To treat lane manoeuvres uniformly, independent of where the actor is on the road, in our translation from WMG-SDL to OSC/ODR we apply the following road transformations:

- A road possessing positive and negative lanes is translated to two separate roads, one for each direction of traffic, and having opposing START and END definitions.
- For right-handed traffic all lanes are negatively numbered. For left-handed traffic all lanes have positive numbers.

This transformation allows us to consistently, independent of traffic direction, use negative signed lane displacements to achieve right lane-change manoeuvres and positive signed lane displacements to achieve left lane-change manoeuvres.

The transformation is depicted in Figure 2.



(a) Classic lane-change in OSC with inconsistent lane offsets (b) Modified Road Structure with consistent lane offsets

Figure 2: Road transformations to consistently specify lane-change manoeuvres in OSC/ODR.

In WMG-SDL, roads are specified using constraints, while in ODR complete road geometries must be defined. For instance, the WMG-SDL description of road *R1* in Figure 3 describes a road with two lanes, one on each side of the centerline. The road has a horizontal geometry consisting of three segments, named *S1*, *CR1*, and *S2*, with the middle segment being curved, with radii of curvature specified where needed (a N/A is used otherwise). The translation from WMG-SDL to ODR involves solving WMG-SDL constraints to form road and junction geometric coordinates and road-segment headings used in ODR.

Figure 4 shows the translation of the horizontal road geometry in WMG-SDL into its ODR specification. The WMG-SDL horizontal road geometry is translated into the ODR specification of the *planView*, which requires concrete geometries to be specified. Further, the WMG-SDL road *R1* is translated into two ODR roads, namely *R1* and auxiliary road *AR1*. The two ODR roads represent the two sides of the centerline for the WMG-SDL definition of *R1*. This is in accordance with the transformations described earlier and depicted in Figure 2.

Roads:
R1:
START
Road type [Minor road] as [R1] with zone as [N/A] AND speed limit of [30] in a [Rural] environment with
Number of lanes [2] as [R1.L-1, R1.L1]
Road traffic direction [Right-handed]
Lane type [Traffic lane]
Lane markings [Broken line]
Horizontal road geometry [S1:Straight, CR1: Curved, S2:Straight] with curvature radius of [S1:N/A, CR1:3.05 to 4.57, S2:N/A]
Vertical road geometry [Level plane]
Transverse road geometry [Divided] with [No] roadside feature
Roadway edge features [Pavement]
Fixed road structures [Trees, Buildings, Street lights : {spacing: 20 to 30, height: 4.5 to 12}]
Length [S1:10 to 12, CR1:7 to 9, S2: 10 to 12] AND Lane width [3.4 to 3.6]
END

Figure 3: WMG-SDL: Description for a turning road.

3.1.2. Junctions

The translation for junctions is syntactic and poses fewer geometric challenges than the translation for roads. Once the list of roads is translated into the ODR specification, WMG-SDL junctions may be

```
<OpenDRIVE>
  <road id="1" junction="-1" length="30.0" name="R1" rule="RHT">
    <planView>
      <geometry hdg="0.0" length="11.0" s="0.0" x="0.0" y="0.0">
        <line/>
      </geometry>
      <geometry hdg="0.0" length="8.0" s="11.0" x="11.0" y="0.0">
        <arc curvature="0.26246719160104987"/>
      </geometry>
      <geometry hdg="2.099737532808399" length="11.0" s="19.0"
      x="14.289332419748945" y="5.73260038291595">
        <line/>
      </geometry>
    </planView>
    <lanes>
  </road>
  <road id="2" junction="-1" length="30.0" name="AR1" rule="RHT">
    <planView>
      <geometry hdg="5.2413301863981925" length="11.0" s="0.0"
      x="8.738517665923368" y="15.229360649907651">
        <line/>
      </geometry>
      <geometry hdg="5.2413301863981925" length="8.0" s="11.0"
      x="14.289332419748947" y="5.732600382915951">
        <arc curvature="-0.26246719160104987"/>
      </geometry>
      <geometry hdg="3.1415926535897936" length="11.0" s="19.0"
      x="11.000000000000004" y="8.881784197001252E-16">
        <line/>
      </geometry>
    </planView>
    <lanes>
    <type type="Minor road">
  </road>
</OpenDRIVE>
```

Figure 4: ODR: Translation of road R1's geometry from Figure 3.

translated to the *junction* element syntax in ODR. Similar to the translation for roads where a declarative description of a road must be translated into concrete geometries in ODR, a junction in WMG-SDL translates into a series of road connections in ODR. ODR requires each junction to specify the roads connecting and the links between lanes. Roundabouts in WMG-SDL are translated into groups of junctions in ODR.

The singular challenge in defining junctions, is defining smooth connections between roads at a junction. This requires the definition of connecting roads. A junction is treated as the connection of multiple roads to a single reference road. This is necessary to compute the geometries of the roads as they approach/leave a junction. We first set one of the roads arriving at the junction to be the reference road and calculate its end-point using standard geometry. This end-point acts as the starting point of the connecting roads. A connecting road is a road segment that connects the reference road to other roads at the junction and is characterized by the smooth curve required to link roads at the junction. The ending point of the connecting road becomes the starting point of the roads in the WMG-SDL junction road list.

Consider Figure 5. The reference road RR (with heading θ) has an end-point B(x,y). Assume *R2* is one of the roads meeting at the WMG-SDL junction, making an angle α relative to *RR*. We generate the smooth connecting road *BE* with radius of curvature r. Using trigonometry projections, with knowledge of the radius r, the point on the circle B and the subtended angle α, the end-point of the connecting road $E(x', y')$ may be calculated as,

$$x' = 2.r.Sin(\alpha/2).Cos(\theta + \alpha/2) \quad (1)$$

$$y' = 2.r.Sin(\alpha/2).Sin(\theta + \alpha/2) \quad (2)$$

The heading of the connecting road is calculated to be $\theta + \alpha$.

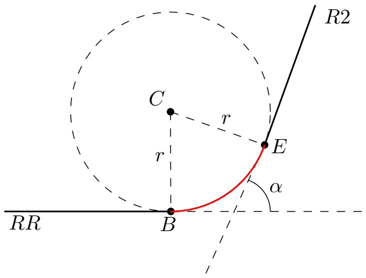


Figure 5: ODR Translation for the geometry of Geometry for Connecting Roads.

3.2. Dynamics (scripted)

In this section, we describe the translation of actor dynamics specified in WMG-SDL into OSC. Due to limitations of space, we are unable to provide code in OSC/ODR corresponding to WMG-SDL. Instead, for each component of an actor's dynamics, we describe its mapping from WMG-SDL into OSC elements. We also mention which translations are not possible due to syntactic limitations of OSC or for which we are unable to find semantic alternatives.

The more complex component of the scenario specification to translate is actor dynamics. Dynamics are specified in the OSC specification. In WMG-SDL, the scripted dynamic elements of the scenario are specified as a series of manoeuvre sequences for each actor. These sequences may be required to execute synchronously or asynchronously.

In WMG-SDL, the dynamic elements are grouped into an initialization, a collection of synchronous activities, and a stopping condition. The dynamics in OSC are described within a *Storyboard*, which consists of three core components, (1) *Init* - describing where the actors are placed and their initial dynamics when the scenario begins, (2) *Story* - describing the activity of actors during the scenario, (3) *StopTrigger* - specifying the conditions, if true, cause the scenario to halt immediately. In the following text, we describe how each of these components is constructed from the WMG-SDL scenario.

3.2.1. Init

The initialization in WMG-SDL is broken up into three components, (1) The Ego Initialization, (2) Vehicle and Pedestrian initializations - relative to the Ego, and (3) Timer Declarations.

For instance, the WMG-SDL single line initialization of *Ego* in the second lane, right of the centerline, on road *R1*, as shown in Figure 6, is translated into OSC code in Figure 7. Similarly the translation of the relative position of *SideVehicle* is depicted in Figure 8. In the translations, when an absolute position of the actor is known, either in terms of its road-lane position or co-ordinate position, the OSC *LanePosition* element is used. Whereas, if a relative position is known, the OSC *RelativeLanePosition* element is used.

3.2.2. Story

In WMG-SDL, all activity is organized as a collection of Synchronised Serial Manoeuvre Sequences (SSMSs). An actor's activity is a sequence of phased manoeuvres belonging to the actor. In Figure 9,

```
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]
```

Figure 6: WMG-SDL: Initialization of Dynamic Elements.

```
<Private entityRef="Ego">
  <PrivateAction>
    <TeleportAction>
      <Position>
        <LanePosition roadId="1" laneId="-2" offset="0.0" s="0.0"/>
      </Position>
    </TeleportAction>
  </PrivateAction>
  <PrivateAction>
    <LongitudinalAction>
      <SpeedAction>
        <SpeedActionDynamics dynamicsShape="step" value="0.0"
        dynamicsDimension="time"/>
        <SpeedActionTarget>
          <AbsoluteTargetSpeed value="15.5"/>
        </SpeedActionTarget>
      </SpeedAction>
    </LongitudinalAction>
  </PrivateAction>
</Private>
```

Figure 7: OSC: Translated Initialization for *Ego* from Figure 6.

an SSMS is represented by the dashed box that groups the various actors' manoeuvre sequences. The phases are numbered from '1' to the number of phases in the SSMS. A *WHEN* condition triggers an SSMS to begin, which causes all actors' manoeuvre sequences to begin at Phase 1. Phases across different sequences, but within the same SSMS, having identical index values, operate synchronously. Any two SSMSs taken together may operate asynchronously from each other.

For an actor, a phase consists of a manoeuvre, manoeuvre parameters, and a *WHILE* invariant condition that must hold while that phase is in operation. In Figure 9, the invariant for phase i and actor j is represented by the symbol C^i_j. A phase is considered 'complete' once the active component of all actors' manoeuvres, in the same phase, have completed, or when any *WHILE* condition linked to that phase is invalidated. Hence, so long as all *WHILE* conditions of the phase hold, any actor having completed its active manoeuvre component continues with a default drive action until all actors in the same phase

```
<Private entityRef="SideVehicle">
  <PrivateAction>
    <TeleportAction>
      <Position>
        <RelativeLanePosition entityRef="Ego" dLane="-1" ds="0.0"
        offset="-1.75" />
      </Position>
    </TeleportAction>
  </PrivateAction>
  <PrivateAction>
    <LongitudinalAction>
      <SpeedAction>
        <SpeedActionDynamics dynamicsShape="step" value="0.0"
        dynamicsDimension="time"/>
        <SpeedActionTarget>
          <AbsoluteTargetSpeed value="15.5"/>
        </SpeedActionTarget>
      </SpeedAction>
    </LongitudinalAction>
  </PrivateAction>
</Private>
```

Figure 8: OSC: Translated Initialization for *SideVehicle* from Figure 6.

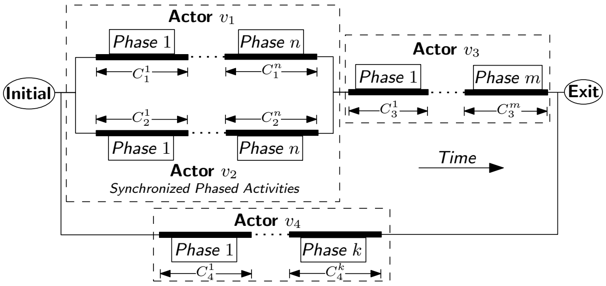


Figure 9: WMG-SDL: Execution Semantics of Phased Manoeuvre Sequences.

complete their respective active manoeuvre components. A drive-only (stopping, accelerating or braking to reach a target velocity) phase, unlike manoeuvres requiring lateral motion (for instance lane changes) is unique. Hence, the completion of a phase containing a drive manoeuvre necessarily requires either a *WHILE* condition associated with the drive manoeuvre to become invalid OR once the active 'speed change' component of the drive action is complete, the drive phase is considered complete.

In WMG-SDL, each SSMS begins with a triggering condition, the *WHEN* condition. If this condition is satisfied, then the actors may begin executing the manoeuvres defined within the SSMS. Table 1 summarizes the hierarchy of OSC elements used for translating the two types of *WHEN* conditions in WMG-SDL. The first, being movement-related (going ahead or being stopped), while the other is based on the vehicles position on a road or lane.

Similarly, manoeuvres in WMG-SDL can be translated into OSC elements as described in Table 2. WMG-SDL manoeuvres are expressed as *X_Y*, comprised of an absolute manoeuvre (*X*) and a relative manoeuvre(Y). Relative manoeuvre components of a WMG-SDL manoeuvre, such as Cut in, Cut out, Towards, and Away, cannot be translated into OSC because relative actions aren't supported in OSC.

In OSC, activity is grouped into asynchronous *Stories*. Synchronization is achieved by explicitly triggering stories or contained acts (groups of manoeuvres) using triggering conditions. Table 3 describes the translation of WMG-SDL phased manoeuvres into OSC elements. To achieve synchronization between phases, the OSC Act associated with a phase is conditioned to start only when all earlier phases, across all actors, within the same SSMS, have ended.

Furthermore, WMG-SDL makes use of timers to synchronize activities across actors. Two types of timers are used, global timers (shared across actors' phases, and SSMSs), and local times (local to a phase and reset when there is a phase change). Global timers have a syntactic equivalent in OSC, which uses simulation time, and supports conditions defined on it. Unfortunately, OSC does not have a mechanism for defining timers local to any other element within a scenario. OSC associates each activity (a group of manoeuvres, actions, and events) with a name and a state. The state of the activity describes whether it has started, is running, or has stopped or ended. It is possible to specify timings relative to when a change of state occurred for such elements. We use this to implement local timers of WMG-SDL. A local timer is used to condition the end of a phase

WMG-SDL *WHEN*	OSC Element
The beginning of an OSC Maneuvre is marked using a StartTrigger hierarchically containing the following elements: StartTrigger - ConditionGroup - Condition - TypeOfCondition	
Going Ahead Stopped	ByEntityCondition - EntityCondition - SpeedCondition
Road or Lane	ByEntityCondition - EntityCondition - ReachPositionCondition - Position - LanePosition

Table 1: WMG-SDL WHEN Manoeuvre Condition → OSC Elements

WMG-SDL Manoeuvre	OSC Action
Drive_{Towards,Away,CutIn,CutOut}	SpeedAction
Stop_{Towards,Away,CutIn,CutOut}	
Reverse_{Towards,Away,CutIn,CutOut}	
LaneChangeRight_{Towards,Away,CutIn, CutOut}	laneChangeAction
LaneChangeLeft_{Towards,Away,CutIn, CutOut}	
TurnRight_{Towards,Away,CutIn,CutOut}	RoutingAction
TurnLeft_{Towards,Away,CutIn,CutOut}	

Table 2: SDL Manoeuvre → OSC Action Map

in WMG-SDL. We translate this into a condition on the OSC *Act* corresponding to the phase. The condition requires the *Act* to stop after the limit on time defined on the WMG-SDL local timer for that phase has expired. This condition is defined as a delay after the *Act* began.

3.2.3. StopTrigger

We define two types of StopTriggers for OSC, as it relates to WMG-SDL. These are as follows:

1. Global Timer: WMG-SDL requires a scenario to end if its global timer reaches a limit. In OSC, the global timer is simulation time, and this condition is translated into a StopTrigger for the Story. The trigger condition is the simulation time having reached the limit defined in WMG-SDL.
2. Collisions: In WMG-SDL, a scenario ends whenever there is a collision with the Ego. In OSC, this can be expressed using the *CollisionCondition*.

3.3. Environment

The environment specification in WMG-SDL is translated into the OSC specifications pertaining to the time of day, weather and road condition. These attributes are direct translations of the syntax of environment in WMG-SDL. An example of the environment in WMG-SDL and its equivalent in OSC are shown in Figures 10 and 11.

3.4. Traffic (unscripted)

Traffic in OSC is modelled using OSC *TrafficActions* which allow for the specification of traffic sources and sinks, and traffic swarms. The traffic generated

WMG-SDL Element	OSC Element	Translation
SSMS	Story	Each SSMS is mapped into an OSC Story.
Phase	Act (Story)	For each actor's manoeuvre phase, within its manoeuvre sequence, an OSC Act is created. This is because it is possible to trigger the beginning and end of an Act, which allows us to synchronize phases across all actors within the same SSMS.
WHEN (SSMS)	StartTrigger (Act_1)	The WMG-SDL SSMS *WHEN* condition is mapped into the StartTrigger of the first act (which we refer to as Act_1) corresponding to the first phase for every actor's manoeuvre sequence.
WHILE (Phase)	StopTrigger (Act)	The StopTrigger of all acts are defined by their respective phase's WHILE condition.
Phase Synchronization	StartTrigger (Act)	The StartTrigger for every other phase is designed to be a group of conditions. Each condition asserts that the OSC Act associated with the earlier phase has reached its *completedState*, *endTransition*, or *stopTransition* state. These states are implemented in OSC simulators as states of an OSC Act.

Table 3: Translation of WMG-SDL Elements into OSC Elements for synchronous and asynchronous activities.

```
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
```

Figure 10: WMG-SDL: Environment definition.

will follow a path decided by their corresponding driver model, i.e. their manoeuvres are not explicitly scripted or specified. The translation of WMG-SDL traffic to OSC traffic is syntactic.

In OSC, traffic is specified as part of the initialization block. A *TrafficAction* is defined for each traffic source (*TrafficSourceAction*) and sink (*TrafficSinkAction*). However, OSC requires a traffic vehicle controller model to be specified for the automated traffic. The tool *esmini* contains a few controller examples, any one of which may be used as the traffic controller.

The traffic source and sink require a radius (the radius of the traffic source where vehicles appear around a specific position), rate (the rate on which vehicles appear at the source location in vehicles/sec), and an optional starting velocity (in m/s) to be specified. Additionally, the position and definition of the vehicle controller must be specified. WMG-SDL does not currently specify the radius of traffic sources and sinks and we, therefore, assume this to be half the width of the road on which traffic is spawned (for a source) or destroyed (for a sink).

```xml
<GlobalAction>
  <EnvironmentAction>
    <Environment name="">
      <TimeOfDay animation="true" dateTime="2022-04-11T03:00:52.577+0100"/>
      <Weather cloudState="free">
        <Sun intensity="10000.0" azimuth="0.0" elevation="20.0" />
        <Precipitation precipitationType="dry" precipitationIntensity="0.0" />
        <Wind direction="0.0" speed="0.1" />
      </Weather>
    </Environment>
  </EnvironmentAction>
</GlobalAction>
```

Figure 11: OSC: Translated Environment definition from Figure 10.

4. Results

We test our translation on two benchmark scenario sets, (1) Automated Lane Keeping System (ALKS) scenarios (UNECE Reg. 157), and (2) Low-Speed Automated Driving (LSAD) scenarios (ISO 22737). We use 14 ALKS scenarios describing, crossing pedestrians, cut-ins with no collision, cut-in with an unavoidable collision, cut-out with a fully blocking target, cut-out with multiple blocking targets, following a lead vehicle comfortably, following a lead vehicle that performs an emergency brake, forward detection range test, fully blocking target, lateral detection range test, multiple blocking targets, partially blocking target, swerving lead vehicle, and swerving side vehicle.

We use 13 LSAD scenarios describing varying driving areas (blocked, shrinking, unblocked), pedestrian positions (moving, stationary, as an obstacle non-occluded and hazardous, non-occluded and moving in the same direction, as an obstacle occluded), and similarly for pedal cyclists as an obstacle.

We begin with scenarios written in WMG-SDL. Each WMG-SDL specification is translated into an ODR specification containing the road-junction network, and an OSC specification describing the traffic, environment, and any scripted behaviours of entities in the scenario. WMG-SDL descriptions are verified against the translated OSC/ODR specification using visual inspection by playing the specification in *esmini*, and examining the simulation log to ensure that WMG-SDL events occur as expected and are timed correctly. The latter is used to examine synchronization.

We selected a group of individuals who are familiar with WMG-SDL and understand its semantics but were not involved in developing the translation to execute our scenarios in *esmini* and assess whether the scenario is subjectively performing as expected. The translation was found to work on the ALKS and LSAD benchmark examples. A selection of translation results is made available on the Safety Pool™ Scenario Database knowledge-base (WMG, 2022).

5. Discussion

WMG-SDL allows validation engineers, regulatory specialist and system engineers to describe scenarios at a functional and logical level, containing sufficient detail for simulation. Translating WMG-SDL into OSC/ODR generates specifications that are lengthy and hard to read. Since writing lengthy, hard to read, and hard to debug specifications by hand is undesirable, it seems easier to express scenarios in WMG-SDL and utilize translators to convert a high-level functional description into low-level concrete scenarios in OSC/ODR.

Due to the richness of specification allowed by the

OpenX standard, it may be possible to translate the same SDL specification into OpenX formats in several different ways. In this article we choose translations that we feel best map individual WMG-SDL specifications into their corresponding OpenX syntactic structures, while listing down all assumptions made in the translation process. In some cases, a direct translation is not possible (for instance - translating some WMG-SDL manoeuvres into a series of atomic OpenScenario actions, translating road structures into OpenDrive, using timers in a scenario). In such cases we mention this clearly, along with the reason why a mapping may not be directly found.

It is worth mentioning that there are efforts to develop OSC 2.0 which as a language is a complete departure from the XML specification syntax and is not backward compatible with OSC 1.x. Since our translation is rooted in existing standards – as supported by simulation tools, scenarios translated using the mapping provided herein will not be applicable for OSC 2.0. However, once the OSC 2.0 standard is published, a similar exercise will be undertaken.

Overall, the translation from WMG-SDL to OSC and ODR is not always a syntactic translation because the two languages do not share semantically equivalent language elements. The translation we develop in this article is a semantic one. Furthermore, the target languages do not fully support expressing all elements of WMG-SDL. For instance, relative manoeuvres in WMG-SDL, such as cut-ins and cut-outs cannot be expressed in OSC.We envisage such challenges occurring in translating other languages to OSC and ODR too.

6. Conclusions/Implications

Various SDLs are currently used in the scenario-based testing workflow for ADS. In this paper, we demonstrate a methodology to translate scenarios in WMG-SDL to OSC/ODR specifications, in order to enhance the compatibility between the languages. The presented translation is semantic since the two languages are not syntactically equivalent. Furthermore, the target languages, OSC/ODR, do not fully contain the syntactic elements required to implement the full gamut of syntax in WMG-SDL and also have selective support from simulation tools. The translation is evaluated using benchmark ALKS and LSAD scenarios simulated using the *esmini* tool. The translation demonstrates that it is possible to write high-level functional descriptions of scenarios in WMG-SDL and automate their translations into low-level OpenX specification standards OSC/ODR. This allows scenarios to be written in a readable form that is easy to share, debug and maintain.

Acknowledgements

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. A selection of openly available scenarios from this study are available on the Safety Pool™ Scenario Database knowledge-base (WMG, 2022).

References

ASAM, 2021a. ASAM OpenDRIVE® Standard. Available at: ¡https://www.asam.net/standards/detail/opendrive/¿.

— 2021b. ASAM OpenSCENARIO® Standard. Available at: ¡https : / / www . asam . net / standards / detail / openscenario/¿.

De Gelder, E., et al., 2022. Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework. *IEEE Trans. on Intelligent Vehicles*, p. 1.

Eclipse, 2021. Language Engineering for Everyone. Available at: ¡https://www.eclipse.org/Xtext/¿.

Esenturk, E., Wallace, A. G., Khastgir, S., and Jennings, P., 2022. Identification of Traffic Accident Patterns via Cluster Analysis and Test Scenario Development for Autonomous Vehicles. *IEEE Access*, 10, pp. 6660–6675.

Fortellix, 2022. Open M-SDL: Measurable Scenario Description Language. Available at: ¡https : / / www . foretellix . com / open-language/¿.

Fremont, D. J., Dreossi, T., Ghosh, S., Yue, X., Sangiovanni-Vincentelli, A. L., and Seshia, S. A., 2019. Scenic: A Language for Scenario Specification and Scene Generation. In: *Proc. of the 40th ACM SIGPLAN Conf. on PLDI*, pp. 63–78.

Fremont, D. J., Kim, E., Pant, Y. V., Seshia, S. A., Acharya, A., Bruso, X., Wells, P., Lemke, S., Lu, Q., and Mehta, S., 2020. Formal Scenario-Based Testing of Autonomous Vehicles: From Simulation to the Real World. In: *2020 IEEE 23rd Int. ITSC*, pp. 1–8.

Gangopadhyay, B., Khastgir, S., Dey, S., Dasgupta, P., Montana, G., and Jennings, P., 2019. Identification of Test Cases for Automated Driving Systems Using Bayesian Optimization. In: *IEEE ITSC*, pp. 1961–1967.

Kalra, N. and Paddock, S. M., 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract*, 94, pp. 182–193.

Khastgir, S., Brewerton, S., Thomas, J., and Jennings, P., Nov. 2021. Systems Approach to Creating Test Scenarios for Automated Driving Systems. *Reliability Engineering & System Safety*, 215, p. 107610. ISSN: 0951-8320.

Knabe, E., et al., 2022. Environment Simulator Minimalistic (esmini). Available at: ¡https : / / github . com / esmini / esmini¿.

Menzel, T., Bagschik, G., Isensee, L., Schomburg, A., and Maurer, M., 2019. From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment. *IEEE Intelligent Vehicles Symp., Proc.*, 2019-June, pp. 2383–2390.

Menzel, T., Bagschik, G., and Maurer, M., 2018. Scenarios for development, test and validation of automated vehicles. *arXiv*, 2018-June(Iv), pp. 1821–1827. ISSN: 23318422.

Neurohr, C., Westhofen, L., Butz, M., Bollmann, M., Eberle, U., and Galbas, R., 2021. Criticality Analysis for the Verification and Validation of Automated Vehicles. *IEEE Access*, 9(i).

Queiroz, R., Berger, T., and Czarnecki, K., 2019. GeoScenario: An Open DSL for Autonomous Driving Scenario Representation. Available at: ¡https://geoscenario.readthedocs.io/¿.

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

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., and Maurer, M., 2015. Defining and Substantiating the Terms Scene , Situation , and Scenario for Automated Driving. *Proc. of the 2015 IEEE 18th ITSC*, pp. 982–988.

WMG, U. o. W., 2021a. Safety Pool^{TM} Scenario Database. Available at: ¡https://www.safetypool.ai/database¿.

— 2021b. World's largest public scenario database for testing and assuring safe Autonomous Vehicle deployments. Available at: ¡https://warwick.ac.uk/newsandevents/pressreleases/worlds_largest_public/¿.

— 2022. Proposed Two-Level Scenario Description Language and ASAM OpenX Scenario Translations. Available at: ¡https:// docs.safetypooldb.ai/docs/sdl¿.

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