Manuscript version: Author's Accepted Manuscript
The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:
http://wrap.warwick.ac.uk/164739

How to cite:
Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:
Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

Vehicle-to-Everything (V2X) in Scenarios: Extending Scenario Description Language for Connected Vehicle Scenario Descriptions*

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

Abstract— **The move towards connected and autonomous vehicles (CAVs) has gained a strong focus in recent years due to the many benefits they provide. While the autonomous aspect has seen substantial advancement in both the development and testing methodologies, the connected aspect has lagged behind, especially in the verification & validation discussions. Integrating connectivity into the development and testing framework for CAVs is a necessity for ensuring the early deployment of cooperative driving systems. A key element within such a framework is a test scenario, which represents a set of scenery, environmental conditions, and dynamic conditions, that a system needs to be tested in. However, the connectivity element is not present in any of the current state of the art scenario description languages (SDLs) that are publicly available. This leaves a gap within the CAV development ecosystem. To accommodate for, and accelerate the development of, connected vehicle systems and their verification and validation methods, this paper proposes a novel V2X extension to the previously published two-level abstraction SDL. The extension enables communications between vehicles, infrastructures, and further additional entities to be specified as part of the scenario and be subsequently tested in virtual testing or real-world testing. Eight new V2X attributes have been added to the SDL. An example set of syntax and semantic definitions are presented in this paper targeting two different abstraction levels – level 1 aims at the abstract scenario level for non-technical end-users such as regulators, and level 2 aims at the logical and concrete scenario level for end-users such as simulation test engineers.**

I. INTRODUCTION

The recent advancements in CAVs are driven by their many benefits, however their safety assurance poses challenges [1]. Kalra et. al. estimated that CAVs need to be driven 11 billion miles to prove they are 20% better than human baseline [2]. This has shifted their safety assurance approach from 'quantity' driven to 'quality' driven within the industry and academia [3]. Purposely defined scenarios have become the predominant elements to represent such 'quality', leading towards a scenario-based testing approach which investigate how a system perform against its design requirements and how it fails [4][3]. A common definition of scenario, in the context of CAV testing is proposed by Ulbrich et al [5] as:

'A scenario describes the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Action & 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.'

There exist a number of scenario generation methods: data-driven approaches (e.g. accident analysis [6]) and knowledge based approaches (hazard analysis [3] or ontological approaches [7]). However, in order to communicate with various stakeholders in the ADS landscape, a common language for scenario definition is required. Several efforts have been invested to design and develop languages for specifying scenarios, such as OpenX (OpenScenario and OpenDrive) [8], Scenic [9], and two-level SDL [10]. However, the capability of describing the connectivity element details is lacking in all of current scenario languages.

Connectivity has become a ubiquitous technology for CAV, and it is now widely accepted that some form of V2X connectivity will be essential to achieve the previously stated benefits of moving towards CAV [11]. In addition to improved safety, V2X would also bring about improved traffic throughput using Vehicle to Infrastructure (V2I) communications and Vehicle to Vehicle (V2V) communications; enabling connected autonomous functions, such as Cooperative Adaptive Cruise Control (CACC) [12] and Green Light Optimal Speed Advisory (GLOSA) [13] systems. Significant developments in technologies such as these make describing scenarios that involve communications technology a logical necessity in ADS testing. This is further highlighted by the V2X test cases published in the 5GAA whitepaper, Safety Treatment in V2X Applications [14].

There is currently a gap in the provision made by scenario definition languages in terms of sufficiently capturing the detail of V2X capabilities. There is a selection of scenario description formats currently available which can be executed in various simulation environments. Particularly of note are the OpenX contributions such as OpenScenario [8] and OpenDrive [15]. As a two-level abstraction scenario description language (SDL) targeting multiple groups of end users, the two-level SDL provides a format that offers a complete coverage along the V model. The motivation of this piece of work was to create a readable, exchangeable and far-reaching extension for scenario description including V2X capability and enable further connectivity development across the industry. This paper proposes an extension to 'SDL Scenario Description Language for Automated Driving Systems' [10] by incorporating V2X elements. The remainder of this paper will cover related work in the field of V2X technology and scenario language development in Section II, an introduction to SDL in Section III, and the development of a language extension for V2X capabilities in SDL in Section IV.

---

*Research supported by UK Research & Innovation (UKRI) Future Leaders Fellowship, Transport Canada and the Department for Transport, HM Government.

All authors are with WMG, University of Warwick, Coventry, CV4 7AL, UK. (e-mails: {patrick.irvine; pete.baker; antonio.bruto-da-costa; jason.zhang; s.khastgir.1; paul.jennings} @ warwick.ac.uk).

II. RELATED WORK

A. *Understanding connectivity in scenarios*

V2X in the context of CAV application is an area of research experiencing an influx of interest in recent years. Several papers have attempted to summarise the state of the art signal technologies [16][11], security concerns related to V2X [17][11], and the potential benefits it offers [18]. Signal technologies and protocol assessment are particularly popular areas of research. Work in this area appears to consider connectivity separately from established testing and safety assurance methods utilized elsewhere in the field. For instance a comparison of the communication system protocols ITS-G5 using IEEE 802.11p and C-V2X using 3GPP [19], brings to light specific performance discrepancies under testing. The major communication standards for connectivity have been compared and debated at great length and newer solutions such as 3GPP continue to emerge [20]. Security concerns and potential for attacks were raised of these communications interfaces [11]. This area is widely discussed under the cyber security topic within the context of CAV. The different methods of attack are categorised in [21] with potential countermeasures suggested in [22]. A suitable method for testing these conclusions in the context of scenarios has not yet been devised within industry or academia.

V2X covers the capability for vehicles to interact with four types of communication being to vehicles (V2V), infrastructure (V2I), pedestrians (V2P) and network (V2N) [20], each application of communications for CAVs leans naturally to the use of varied communications protocols be it, DSRC/IEEE 802.11p, 3GPP C-V2X, ETSI or TTA. The application of all communications categorized above are safety critical, work from Wang et al [20] gives examples of messages transmitted using each protocol that could contribute to a reduction in traffic accidents from collision warnings to wrong-way driving warnings. The above (along with specific test cases in [14]) exemplify a small selection of the safety cases which will need to be described and tested as scenarios, outlining further the importance of sufficient provision in scenario description languages. Be it through security concern, system protocol performance comparison, or technology introduction, all methods of V2X communication will be required to be tested using scenarios during their safety assurance, therefore a suitable method of description is a necessity.

B. *Scenario description languages*

Several scenario description languages exist for the scenario-based testing activities within the CAV domain. Common, executable and exchangeable language format for representing the scenarios across different abstraction levels have been the direction where industry and academia are aiming. For a scenario to be relevant for CAV, the language must take into account of the Operational Design Domain (ODD) of the system. ODD, as defined by SAE J3016, represents the operating conditions under which a given driving automation system or feature thereof is specifically designed to function [23]. ODD covers the environmental and scenery elements that could affect a system and are crucial to consider in the scenarios design. The BSI PAS 1883 ODD taxonomy for an automated driving system attempts to define and provide end users with a set of baseline attributes that should be considered during the system development and testing process [23].

When considering the which format of scenario description to follow it is important to consider the desired stakeholder being targeted by the outcome. There are several stakeholder groups to be considered who will come int contact with scenarios at various times during the validation and verification process. More technically minded end-users of scenarios could include ADS technology developers, simulation test engineers and real-world test engineers. Whilst less technically minded end users such as regulators, local government and the public will also have contact with scenarios. Several open formats exist for describing road layouts and other relevant scenery objects related to the ODD attributes, for example OpenDrive [15], OpenStreetMap [24] and Lanelet2 [25]. OpenDrive [15], when combined with OpenScenario 1.x [8], an open source scenario description language, has demonstrated execution capabilities across various simulators, however, as documented in a recent study the human readability aspect of OpenScenario 1.x and OpenDrive is limited [26]. Scenic [9] is another open source scenario description format which has received increasing attention, execution capability is being developed [27], but there is also the need to cover the more abstract level of scenarios with an emphasis on human readability. The two-level SDL [10] published previously aims at providing both a human readable format as well as a format that can be executed in simulators such as CARLA [27]. Furthermore, the two-level SDL format can be converted into the low level OpenScenario 1.x and OpenDrive format, which utilizes the OpenX standards and further increases its application usages. As indicated in the same study [26], OpenScenario 1.x and OpenDrive are great low level executable formats, it would be advantageous to combine with a structured, human readable, high level texture format. Similar to OpenDrive and OpenScenario 1.x, other common and available description formats put emphasis on executability and less on human readability.

III. TWO ABSTRACTION LEVEL SCENARIO DESCRIPTION LANGUAGE (SDL)

An extension for the inclusion of V2X into scenarios should aim to reach as large an audience as possible. Developing a framework for connectivity in scenarios would be useful for execution purposes, but also for communicating a complex technical component of CAV systems to those who are not necessarily technically minded but will still have involvement with scenarios. For the description of V2X to be accessible to as many stakeholders as possible, the two-level SDL language provided the obvious choice for a base language to be extended. As discussed in Section II, there are multiple languages which offer a solution for scenario description, but few focus on human readability. As a result of this, the remainder of this paper builds on top of a previous publication of the two-level SDL; extending it to incorporate the V2X description capability within the language. A brief overview of the two-level SDL will be provided in this section, next section will focus on the V2X extension.

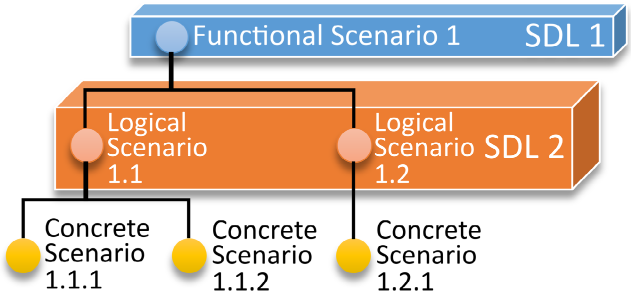

Figure 1. SDL abstraction levels mapped to items of the validation and verification process for ADSs.

The two-level SDL aims to be both human and machine readable, and to be used as a common, understandable and exchangeable format across a wide variety of end users. Considered end users include the public, regulator, system developer, requirement engineer, and test engineer. While synergy exists, there are also competing requirements on the language format, i.e. executable vs human readable, abstract vs detailed. Inspired by the previous three-level abstractions of scenario (functional, logical, and concrete), and the recent four-level abstractions (with the addition of abstract scenario between functional and logical [28][4][29]), the SDL is therefore developed into a two abstraction level. Level 1 aims at functional and abstract level, focuses on structured natural language representation; and level 2 aims at the logical and concrete level, focuses on executability and increasing detail levels. Through a programmatic detailing process, SDL level 1 can be converted to level 2, and through abstracting the reverse can be achieved. Furthermore, SDL level 2 can also be converted into OpenX standard for wider application usage.

From the language model perspective, both levels of SDL consist of scenery, environment, and dynamic aspect [10]. Scenery elements contains the road and junction network, as well as additional road structures within the scope of ODD. Environment elements focused on the weather, lighting conditions, and time of the day. The dynamic elements describe the scenario actors as well as their behaviors. The behaviors description is divided into two parts, the initialization, and the maneuver stage. In the initialization, each actor is located spatially within its environment and given starting positions relative to the established scenery for the scenario. In the maneuver sequence that follows, the initialized actors can perform phased action sequences described using the maneuvers from the SDL behavior library. An example of the dynamic elements of a scenario being described in SDL Level 2 can be seen in the Figure 2. As previously mentioned, SDL level 1 takes the same features used in SDL level 2 and describes them qualitatively, at a higher level of abstraction more suited for non-technical users.

IV. DEVELOPMENT OF V2X EXTENSION IN SDL

The V2X extension can build upon the foundation of SDL to enable scenarios that include connectivity. An important part of adding V2X into SDL was to align with the targets that were already laid out, having a common structure, being executable and understandable. Following research into the required attributes for ADAS system connectivity, a suitable syntax extension was developed.

Initialisation Stage

INITIAL: Vehicle [Ego] in [R4.L-1]
AND Vehicle [V1] in [R2.L-1] AND at relative position [FSL] with relative heading angle [175 to 185] to [Ego]
AND Vehicle [V2] in [R2.L-2] AND at relative position [FSL] with relative heading angle [175 to 185] to [Ego]

Manoeuvre Stage

WHEN: [Ego] is [Going_Ahead] in [R4.L-1]
DO: [V1]
PHASE 1: [Drive_Towards] [-, 8.9 to 11.2, -3 to 3] [Ego: 11.2 to 15.6, FSL]
PHASE 2: [TurnLeft_CutIn] [CR1, 15 to 20, -3 to 3] [Ego: 4.5 to 9, FSL]

AND: [V2]
PHASE 1: [TurnRight_CutIn] [CR1, 15 to 20, -3 to 3] [Ego: 4.5 to 9, FSL]
PHASE 2: [TurnRight_CutOut] [CR1, 20 to 25, -3 to 3] [Ego: 11.2 to 15.6, FSL]

Figure 2. 'Dynamic Elements' example of an SDL scenario description.

A. *Attribute Selection*

Literature regarding the use of connectivity to support AV functions was carried out and an advisor working in the field of connectivity was consulted to arrive at a suitable set of attributes to describe connectivity in scenarios. The result of this process is a selection of 8 attributes which are considered critical for describing scenarios involving connectivity at the detailed executable abstraction level: SDL level 2. The eight attributes that were finalised and introduced into the V2X extension for SDL are: Communications Capability, Transceiver Directionality, Casting Type, Transmission Type, Transmission Size, Transmission Time (of flight), Transmission Signal Strength, Message Type.

The first three attributes (as above) that will be discussed are used within the dynamic initialisation stage of an SDL description; and are illustrated in Figure 3. The 1st attribute is the Communications capability attribute. Appearing in the dynamic initialisation stage for actors (and the scenery elements for connected traffic lights), this attribute is necessary to establish which participants in the scenario are connectivity capable. This is integrated into the existing initialisation sequence for the actors in the scenario to avoid confusion. The 2nd attribute is the transceiver directionality, which specifies what directions the transceiver can transmit and receive signals from. This is described in relation to the actor’s direction, e.g., F, R (for front and rear transceivers). Only one attribute exists related to transmission location in the initialisation, this highlights the necessity to integrate the V2X extension into SDLs established system for locating actors and infrastructure into the scenario (the methods by which this occurs are expanded on later in the syntax description).

The third attribute is the casting type, which is utilised in both the initialisation and manoeuvre stages of the SDL description for actors. When initialising the actors in the scenario, Casting type states the casting capability of each actor. In the manoeuvre stage, the casting type details which casting type is being used by the actor, and as a result what the intended audience of the signal is. There are three types, unicast, multicast and broadcast [30]. Unicast is one-to-one communication, while multicast is one-to-many. Both unicast and multicast expect a ‘handshake’ to indicate that the message has been received. In a broadcast, the actor doesn’t expect to get a ‘handshake’, and hence it is vital for one actor to be able to send certain messages to other actors that might only relate to their specific manoeuvres, for example in the case of a collision warning message. The (S) in the figure below indicates that the Communications Capability attribute is also used in the scenery elements section.

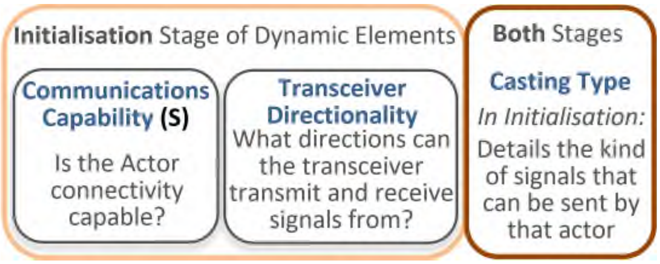

Figure 3. Connectivity related attributes included in the Initialisation phase of the Dynamic elements description.

The last 5 attributes to discuss all fall within the manoeuvre stage of the dynamic elements; and can be seen in Figure 4. The transmission type is a description of the method by which the data is sent i.e., by 5G, DSRC, Bluetooth, FM etc. It is important to define the type of transmission that is being used as the receiving actor will need to be receiving it in the same type in order to utilise the information. The transmission size details the size of the data being transmitted, which could range between approximately 100 bytes and 9000 bytes depending on the complexity of the message. The transmission time indicates the time of flight of the signal. This could be necessary for a multitude of testing purposes, as well as being necessary for a system which is expecting ‘handshake’ will occur following a unicast/multicast message. The transmission signal strength defines the initial strength of the signal, the environment elements and actors’ proximity will determine the final strength of the received signal. Finally, the message type details what type of message is being sent, whether that be location data, a collision warning, collision averted, etc. This is of vital importance to the scenario, as it will likely inform the actions of the system under test. As the ‘Casting Type’ appears in both the initialisation stage and the manoeuvre stage (as discussed earlier), there are 6 attributes in total that are involved in the manoeuvre stage.

A. *Scenery Description*

An alteration was necessary for the Scenery Elements section of SDL to allow for V2I interactions to be captured. V2I currently can be seen primarily in the form of ‘connected infrastructure nodes’ which can take many specific forms, with a primary form being Connected traffic lights. These needed to be inserted into the syntax where possible. Note that the syntax below is only given for SDL level 2 as at the SDL level 1 abstraction, here is not a necessity to localise the communications units in the description, as there is in Level 2. The Junction syntax has a simple alteration in the form of the addition of the optional use of ‘Connected’ before traffic lights to show the communications capability.

Whilst ‘connected traffic light’ provisions are made in the syntax of the junction description, infrastructure nodes used for connectivity are typically roadside objects and are therefore included in the road description syntax. ‘Communications unit’ and ‘Communications Sensor’ were added to the existing special road structures list and a more specific locating syntax was introduced to allow for a distance to be specified from road start or junction. V2X extensions to the SDL Road syntax is presented in Figure 6.

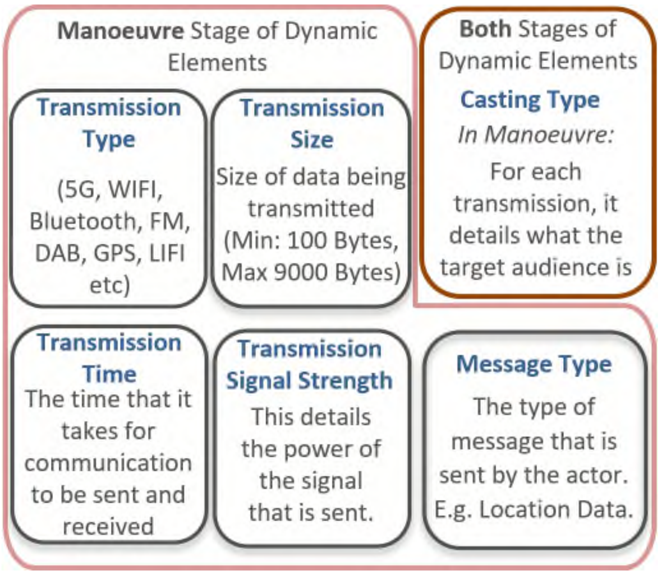

Figure 4. Connectivity related attributes included in the Manoeuvre phase of the Dynamic elements description.

((Connected) opt Traffic light TRAFFIC_LIGHT as [ID -> R_ID_x.L_ID_x:]

Where:
R_ID_x.L_ID_x: Road ID x and lane ID x. E.g., R1.L1, R2.L2.
TRAFFIC_LIGHT: Description of traffic light at the junction.
opt: Indicates optional statement.

Figure 5. Altered SDL junction syntax including connected traffic lights. The highlighted sections are the additions that were necessary to the original syntax.

Road syntax
('Special' 'road' 'structure' '[SRST]' 'as' '[SRS_ID ']' 'at' '['DOUBLE']' 'distance' 'from' ('START')/(‘ID’))*

Where:
SRST: List of Special Road Structures. E.g., Pedestrian Crossing, Sensing Unit, Communications Unit, custom description as a string in quotes.
SRS_ID: Special Road Structure ID.

Figure 6. Expanded SDL Road description syntax including the option to locate 'connected' infrastructures.

C. *Dynamic Description*

Communications contain actions that can be likened to the absolute manoeuvres that already feature in SDL. They can directly interfere with/dictate the scenario itself and therefore must be considered in parallel with the physical manoeuvres that occur.

Communications phases have been introduced where necessary into the Dynamic description of a scenario. Communication phases are not necessary components and are not mapped one to one with the manoeuvre phases of a scenario. Communications phases occur in parallel with the associated physical manoeuvre phase:

i.e. PHASE 1: Manoeuvre activity would happen in parallel with COMMS 1: Communication activity.

However, a communications phase is not required for every physical manoeuvre phase

i.e. PHASE 1: Manoeuvre activity can occur followed by PHASE 2: Manoeuvre activity which occurs in parallel to COMMS 2: Communication activity.

Communications phases can be included fluidly into the story that the scenario and omitted if not necessary.

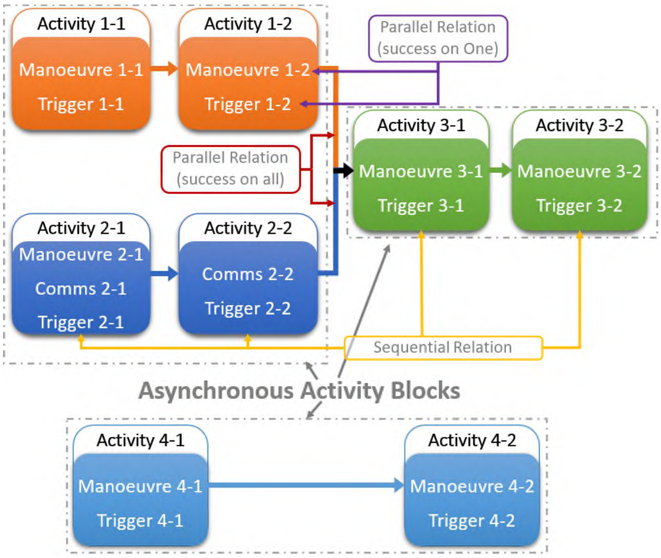

Figure 7. Graphical representation of parallel and sequential manoeuvre sequences, including the optional communications phases in the lower branch.

INITIAL:

There are/is INT ActorType(s), ID_1 ((, ID_2 …) and ID_x).
IDx ((, IDx …) and IDx) are/is connectivity capable.
ID_1 is in RoadID ((, ID_2 is in RoadID, …) and ID_x is in RoadID).

Where:
IDx: Agent ID x. E.g., Vehicle1, Vehicle2, Ego, ConnectedTrafficLight1. The ID may not have any spaces.
RoadID: Road ID x. E.g., Road1, Road2…
ActorType: i.e., vehicle, pedestrian, cyclist, connected structure.
(content): Brackets around *content* imply optional repetition of *content* depending on number of actors.

Figure 8. The revised initialisation syntax for SDL Level 1, including the provision to define communications capability.

In terms of syntax, the functionality for describing V2X communication needed to be flexible enough for allowing description of widely varied communication, as would be likely in safety related testing scenarios.

The attributes identified as necessary during the dynamic section of the SDL description at level 2 are Casting type, Message type, Transmission size, Transmission signal strength, Transmission type, Transmission time (of flight) and Transceiver directionality. At SDL level 1, only a confirmation of V2X capability for the actors is given in the initialisation, and the syntax for the comms phase is restricted to describe Casting type and Message type.

As mentioned earlier, the dynamic elements of a scenario are described in two parts in SDL. Actors are first initialised within the scenario, following this, a behaviour tree style activity phase is entered, see Figure 7, which exemplifies this for a scenario described using SDL level 2. There is little difference when using SDL Level 1, except that at this level of abstraction the phases are not named and rather described sequentially.

To introduce connectivity into the initialisation and activity phases the following syntax has been introduced; stating whether the actor has communication (comms) capabilities and hence if it does, the transceiver's directionality and any casting types that are able to be used are also defined. The syntax for the initialisation in Level 2 can be seen in Figure 9, with the additions for connectivity element description highlighted.

INITIAL :
A_TYPE [A_ID_1] in [R_ID_1.L_ID_1] (with comms capability and [TD] mounted transceivers for [CT] signals) opt (at [AP]) opt (at heading angle [AH]) opt
AND A_TYPE [A_ID_2] in [R_ID_2.L_ID_2] (with comms capability and [TD] mounted transceivers for [CT] signals) opt
(AND with a [D_1] offset of [O_1] to [A_ID_a] (AND with a [D_2] offset of [O_2] to [A_ID_b]) opt) opt
AND at relative position [RP] with relative heading angle [RA] to [A_ID_c]
(AND Global timer [Tx] = [0]) opt
(AND Local timer [tx] = [0]) opt

Where:
A_ID_x: Agent ID x. E.g., V1, V2, Ego, CTL2, SU1, CU1.
A_TYPE: Actor type. E.g., Vehicle, Pedestrian, Cyclist, Connected Traffic Light, Sensing Unit, Communications Unit
AP: Absolute position described in two dimensions: **X , Y**.
AH: Absolute heading angle described as a range **H_1 to H_2**.
CT: Casting type for the mounted transceivers. E.g., Unicast, Multicast, Broadcast.
R_ID_x.L_ID_x: Road ID x and lane ID x. E.g., R1.L1, R2.L2.
RP: Relative position described in the relative position compass. E.g., F, SR, FSR.
RA: Relative heading angle described in the heading angle compass. E.g., 180, 270.
TD: Transceiver directionality described in the relative position compass. E.g., F, SR, FSR.
D_x: Directional can either be longitudinal or lateral.
O_x: Offset value, can be positive or negative to indicate direction.
Tx: Global timer, which runs throughout the scenario.
tx: Local timer, which runs within each manoeuvre phase.
opt: Indicates an optional statement.

Figure 9. Level 2 revised initialisation syntax, with V2X aspects highlighted

ID$_{1}$ CastingType MessageType message of ID$_{2}$ [(, ID$_{3}$ …) and ID$_{4}$] [to ID$_{5}$].

and CastingType MessageType message of ID$_{2}$ [(, ID$_{3}$ …) and ID$_{4}$] [to ID$_{5}$].

Where:
IDx: Agent ID x. E.g., V1, V2, Ego. The ID may not have any spaces.
CastingType: Casting type used by the scripted agent in this interaction. E.g., unicasts, multicasts, broadcasts.
MessageType: Message type that is sent by the scripted agent. E.g., Location, Collision warning, Collision averted.
[content]: Square brackets around *content* implies optional use, depending on the level of detail desired by the end user. In some scenarios this detail may not be necessary.

Figure 10. Two syntax variations for describing a communications phase in SDL level 1, with the relevant variable explained below.

PHASE X: …

(COMMS (X)?: Using transmission type [TT], casting type [CTU], transmission size [TS], transmission time [TTIME], transmission strength [TSS]; Send: [MT] of [A_ID_x] (to [A_ID_x], Acknowledgement [Message Received: Status]. Acknowledgement [Message Read: Status].) opt

(WHILE: …) opt

Where:
A_ID_x: Relative agent ID. The same scripted agent can be A_ID_1, A_ID_2 and A_ID_3.
CTU: Casting type used by the scripted agent in this interaction. E.g., Unicast, Multicast, Broadcast.
MT: Message type that is sent by the scripted agent. E.g., Location data, Collision warning, Collision averted.
Status: Yes/No
RS: Relative speed of the agent relative to A_ID_1.
RP: Relative position to the agent relative to A_ID_1.
TS: Transmission size that the scripted agent is using in this interaction.
TSS: Transmission signal strength that the scripted agent is using in this interaction.
TT: Transmission type that the scripted agent is using in this interaction. E.g., 5G, WIFI, Bluetooth, FM, DAB, GPS, LIFI, Hand Gestures, Car Light Flashing.
TTIME: Transmission time that it takes for communication to be sent and received.
opt: Indicates an optional statement.

Figure 11. The proposed communications phase syntax for SDL Level 2 can be seen here as COMMS X, with the relevant variables explained below.

The syntax for initialisation at the higher level of abstraction in Level 1 can be seen in Figure 8. The syntax for manoeuvre phases remains unchained for regular actions. Communication phases in SDL level 2 have been included, between the existing phased manoeuvre sequence and the optional while condition, using the syntax in Figure 11.

The syntax for the description of connectivity phases has two possible formats in SDL level 1. As covered earlier, level 1 of SDL uses a structured natural language inspired approach to defining the scenario in a way that resembles a story. If the communications phase occurs sequentially, directly following a manoeuvre activity then the first line of the syntax in Figure 10 is used. The second line of syntax in Figure 10 describes an event that occurs in parallel to a manoeuvre activity.

D. *Case Study: Example 1*

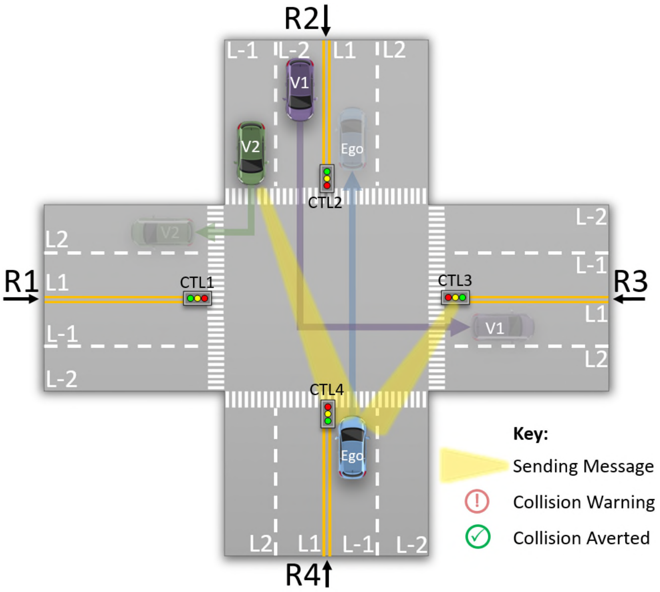

Figure 12. Example 1 represented graphically. It displays a simple case of V2X use in a scenario at a crossroads with 3 vehicles.

In Figure 12 the subject vehicle (Ego) is going straight on at a crossroads whilst another vehicle (V1) coming from the opposite direction is turning left across the subject vehicle. A third vehicle (V2) that is also opposite the Ego is turning right at the crossroads. There are four sets of traffic lights, all of which are connectivity capable and all of the vehicles on the road are connectivity capable. Both V2 and Connected Traffic Light 3 (CTL3) have location data of V1, which they send directly to the Ego in real time.

There are 4 phases in this example (as seen in the SDL Level 2 below). In the first 2 phases, the location data of V1 is sent to the Ego by V2 and CTL3.

Then in phase 3, a collision warning of V1 is sent to the Ego by CTL3; V2 doesn't send this information as it is now out of range of V1 and the Ego. Finally, in phase 4, the collision averted message is sent to the Ego by CTL3 as V1 has completed its left turn.

There are 3 vehicles, Ego, Vehicle1 and Vehicle2. Ego, Vehicle1 and Vehicle2 are connectivity capable. There is 1 connected infrastructure, ConnectedTrafficLight3. Ego is in Road4, Vehicle1 and Vehicle2 are in Road2 and ConnectedTrafficLight3 is in Road3.

When Ego is going ahead, Vehicle1 drives towards Crossroads1, Vehicle2 turns right across Ego at its front left and broadcasts Location message of Vehicle1, ConnectedTrafficLight3 broadcasts Location message of Vehicle1.

Next, Vehicle1 turns left across Ego at its front, Vehicle2 drives away from Ego, and ConnectedTrafficLight3 unicasts Collision warning message of Vehicle1 to Ego.

Next, Vehicle1 drives away from Ego and ConnectedTrafficLight3 unicasts Collision averted message of Vehicle1 to Ego.

Figure 13. SDL Level 1 dynamic description for Example 1, involving V2V and V2I communication.

INITIAL: Vehicle [Ego] in [R4.L-1] with comms capability and [F, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals

AND Vehicle [V1] in [R2.L-1] with comms capability and [F, SL, SR, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals AND at relative position [FSL] with relative heading angle [175 to 185] to [Ego]

AND Vehicle [V2] in [R2.L-2] with comms capability and [F, SL, SR, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals AND at relative position [FSL] with relative heading angle [175 to 185] to [Ego]

AND Traffic Light [CTL3] in [R3.L-1] with comms capability and [F, SL, SR, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals AND at relative position [FSR] to [Ego]

WHEN: [Ego] is [Going_Ahead] in [R4.L-1]

DO: [V1]

PHASE 1: [Drive_Towards] [-, 8.9 to 11.2, -3 to 3] [Ego: 11.2 to 15.6, FSL]

PHASE 2: [TurnLeft_CutIn] [CR1, 15 to 20, -3 to 3] [Ego: 4.5 to 9, FSL]

PHASE 3: [TurnLeft_CutOut] [CR1, 20 to 25, -3 to 3] [Ego: 11.2 to 15.6, F]

PHASE 4: [Drive_Away] [CR1, 20 to 25, 0 to 3] [Ego: 0 to 4.5, FSR]

AND: [V2]

PHASE 1: [TurnRight_CutIn] [CR1, 15 to 20, -3 to 3] [Ego: 4.5 to 9, FSL]

COMMS 1: Using transmission type [5G], casting type [Broadcast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Location data] of [V1].

PHASE 2: [TurnRight_CutOut] [CR1, 20 to 25, -3 to 3] [Ego: 11.2 to 15.6, FSL]

COMMS 2: Using transmission type [5G], casting type [Broadcast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Location data] of [V1].

PHASE 3: [Drive_Away] [CR1, 20 to 25, 0 to 3] [Ego: 0 to 4.5, FSL]

PHASE 4: [Drive_Away] [CR1, 20 to 25, 0 to 3] [Ego: 0 to 4.5, FSL]

AND: [CTL3]

COMMS 1: Using transmission type [5G], casting type [Broadcast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Location data] of [V1].

COMMS 2: Using transmission type [5G], casting type [Broadcast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Location data] of [V1].

COMMS 3: Using transmission type [5G], casting type [Unicast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Collision warning] of [V1] to [Ego]. Acknowledgement [Message Received: Yes]. Acknowledgement [Message Read: Yes].

COMMS 4: Using transmission type [5G], casting type [Unicast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Collision averted] of [V1] to [Ego]. Acknowledgement [Message Received: Yes]. Acknowledgement [Message Read: Yes].

END: [Ego] in [R1.L1]

Figure 14. SDL Level 2 dynamic description for Example 1, involving V2V and V2I communication.

B. *Case Study: Example 2*

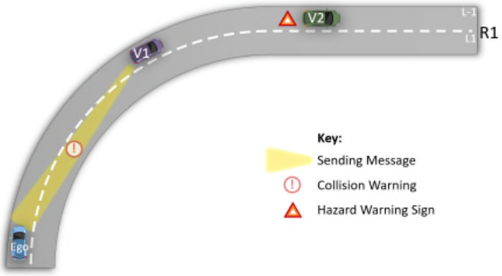

Figure 15. Safety critical V2X scenario of a hazard around a blind corner.

Figure 15 displays an example where the subject vehicle (Ego) is going straight along a bi-directional road which curves to the right. Due to the curvature of the road, Ego does not have a clear line of sight of the road ahead. There are two vehicles ahead of Ego in this scenario, V1 is positioned just passed the crest of the turn and has a clear line of sight to V2. V2 is broken down in the middle of the lane and has a hazard warning sign placed behind it in the lane. V1 and Ego are connectivity capable vehicles.

There are 2 phases in this example; occurring while Ego drives straight in the lane. In the first phase V1 broadcasts the location data of V2. In the second phase, after V1 has stopped behind V2, it unicasts a collision warning message to the Ego vehicle of V1 and V2's stationary presence in the road.

There are 3 vehicles, Ego, Vehicle1 and Vehicle2. Ego, Vehicle1 and Vehicle2 are in Road1.

When Ego is going ahead, Vehicle1 drives away from Ego at its front right and broadcasts location message of Vehicle2 and Vehicle2 is stopped in front of Vehicle1.

Next, Vehicle1 is stopped in front of Ego and unicasts Collision warning message of Vehicle2 to Ego.

Figure 16. SDL Level 1 dynamic description for Example 2 involving V2V and V2I communication.

INITIAL: Vehicle [Ego] in [R1.L-1] with comms capability and [F, SL, SR, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals

AND Vehicle [V1] in [R1.L-1] with comms capability and [F, SL, SR, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals AND at relative position [FSR] with relative heading angle [130 to 140] to [Ego]

AND Vehicle [V2] in [R1.L-1] with comms capability and [F, SL, SR, R] mounted transceivers for [Unicast, Multicast, Broadcast] signals AND at relative position [FSR] with relative heading angle [130 to 140] to [Ego]

WHEN: [Ego] is [Going_Ahead] in [R1.L-1]

DO: [V1]

PHASE 1: [Drive_Away] [-, 18 to 22, -5 to 0] [Ego: -5 to 5, FSR]

COMMS 1: Using transmission type [5G], casting type [Broadcast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Location data] of [V2].

PHASE 2: [Stopped] [-, 0 to 0, 0 to 0] [Ego: -9 to 0, F]

COMMS 2: Using transmission type [5G], casting type [Unicast], transmission size [100-1500], transmission time [30ns-50ns], transmission strength [5-11 dBm]. Send [Collision warning] of [V2] to [Ego]. Acknowledgement [Message Received: Yes]. Acknowledgement [Message Read: Yes].

AND: [V2]

PHASE 1: [Stopped] [-, 0 to 0, 0 to 0] [V1: -9 to 0, FSR]

PHASE 2: [Stopped] [-, 0 to 0, 0 to 0] [V1: 0 to 0, F]

END: [Ego] in [R1.L-1]

Figure 17. SDL Level 2 dynamic description for Example 2, involving V2V

V. CONCLUSION

This paper presents an extension to the two-level SDL to include connectivity elements, such that it can describe complex cooperative automated driving scenarios involving V2X communication. The language provides the user with the ability to specify signal details, along with the ability to convey the messages in the story of the scenario. Preceding the development of this language extension, no documented scenario definition language existed for describing scenarios that involve connectivity aspects. The proposed language extension fits into the established two-level abstraction model, and as such can be used to describe scenarios involving V2X to regulators etc. at the high abstraction of level 1, and maintains the executable feature associated with level 2, allowing developers to test scenarios involving complex cases of communication. The proposed extension incorporates eight V2X attributes in the SDL which include: Communications Capability, Transceiver Directionality, Casting Type, Transmission Type, Transmission Size, Transmission Time (of flight), Transmission Signal Strength, Message Type. The extension establishes a syntax for describing connected infrastructure within the scenery elements of the scenario, as well as describing connected actors; allowing for any number of safety critical communications messages to be described and simulated. The benefit of achieving a common and capable description format for describing connectivity in scenarios is that it sets a precedent for involving connectivity into the existing testing framework as is established for verifying and validating ADSs. By providing multiple levels of abstraction, inherently complex connectivity dependent automated systems can be described in scenarios at a suitable level for the reader, maintaining readability and executability where necessary. As more potential Cooperative Automated Driving Systems which include V2X communication arise from industry, the need to test them grows and the proposed format will allow for easy inclusion into an established language for scenario description.

ACKNOWLEDGMENT

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 this study.

REFERENCES

[1] S. Khastgir, S. Birrell, G. Dhadyalla, and P. Jennings, “Identifying a gap in existing validation methodologies for intelligent automotive systems: Introducing the 3xD simulator,” in *Proc. of the IEEE Intelligent Vehicles Symposium 2015*, 2015, pp. 648–653.

[2] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?,” *Transp. Res. Part A Policy Pract.*, vol. 94, no. December, pp. 182–193, 2016, doi: 10.1016/j.tra.2016.09.010.

[3] S. Khastgir, S. Brewerton, J. Thomas, and P. Jennings, “Systems Approach to Creating Test Scenarios for Automated Driving Systems,” *Reliab. Eng. Syst. Saf.*, doi: doi.org/10.1016/j.ress.2021.107610.

[4] T. Menzel, G. Bagschik, L. Isensee, A. Schomburg, and M. Maurer, “From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2019-June, pp. 2383–2390, 2019, doi: 10.1109/IVS.2019.8814099.

[5] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, “Defining and Substantiating the Terms Scene , Situation , and Scenario for Automated Driving,” 2015, doi: 10.1109/ITSC.2015.164.

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

[7] E. de Gelder *et al.*, “Ontology for Scenarios for the Assessment of Automated Vehicles,” 2020.

[8] ASAM e.V, “OpenSCENARIO, ASAM.” ASAM e.V., 2020.

[9] D. J. Fremont, X. Yue, T. Dreossi, A. L. Sangiovanni-Vincentelli, S. Ghosh, and S. A. Seshia, “Scenic: A language for scenario specification and scene generation,” *Proc. ACM SIGPLAN Conf. Program. Lang. Des. Implement.*, pp. 63–78, 2019, doi: 10.1145/3314221.3314633.

[10] X. Zhang, S. Khastgir, and P. Jennings, “Scenario Description Language for Automated Driving Systems: A Two Level Abstraction Approach,” 2020.

[11] M. Hasan, S. Mohan, T. Shimizu, and H. Lu, “Securing Vehicle-to-Everything (V2X) Communication Platforms,” *IEEE Trans. Intell. Veh.*, vol. 5, no. 4, pp. 693–713, 2020, doi: 10.1109/TIV.2020.2987430.

[12] Z. Wang, G. Wu, and M. J. Barth, “A Review on Cooperative Adaptive Cruise Control (CACC) Systems: Architectures, Controls, and Applications,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2884–2891, doi: 10.1109/ITSC.2018.8569947.

[13] V. Nguyen, O. T. T. Kim, T. N. Dang, S. Il Moon, and C. S. Hong, “An efficient and reliable Green Light Optimal Speed Advisory system for autonomous cars,” in *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2016, pp. 1–4, doi: 10.1109/APNOMS.2016.7737260.

[14] 5GAA, “Safety Treatment in V2X Applications,” 2021.

[15] M. Dupuis, E. Hekele, and A. Biehn, “OpenDRIVE® Format Specification, Rev. 1.5,” *OpenDRIVE*, no. M, pp. 1–133, 2019.

[16] J. Wang, Y. Shao, Y. Ge, and R. Yu, “A survey of vehicle to everything (V2X) testing,” *Sensors (Switzerland)*, vol. 19, no. 2, pp. 1–20, 2019, doi: 10.3390/s19020334.

[17] A. C. Serban, E. Poll, and J. Visser, “A security analysis of the ETSI ITS vehicular communications,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11094 LNCS, pp. 365–373, 2018, doi: 10.1007/978-3-319-99229-7_31.

[18] S. Darbha, S. Konduri, and P. R. Pagilla, “Benefits of V2V communication for autonomous and connected vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1954–1963, 2019, doi: 10.1109/TITS.2018.2859765.

[19] V. Mannoni, V. Berg, S. Sesia, and E. Perraud, “A comparison of the V2X communication systems: ITS-G5 and C-V2X,” *IEEE Veh. Technol. Conf.*, vol. 2019-April, 2019, doi: 10.1109/VTCSpring.2019.8746562.

[20] X. Wang, S. Mao, and M. X. Gong, “An Overview of 3GPP Cellular Vehicle-to-Everything Standards,” *GetMobile Mob. Comput. Commun.*, vol. 21, no. 3, pp. 19–25, 2017, doi: 10.1145/3161587.3161593.

[21] M. Amoozadeh, H. Deng, C. N. Chuah, H. M. Zhang, and D. Ghosal, “Platoon management with cooperative adaptive cruise control enabled by VANET,” *Veh. Commun.*, vol. 2, no. 2, pp. 110–123, 2015, doi: 10.1016/j.vehcom.2015.03.004.

[22] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, “A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures,” *ACM Comput. Surv.*, vol. 48, pp. 1–31, 2016, doi: 10.1145/2897166.

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

[24] M. Haklay and P. Weber, “OpenStreetMap: User-Generated Street Maps. Pervasive Computing,” *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, 2008, doi: 10.1109/MPRV.2008.80.

[25] F. Poggenhans *et al.*, “Lanelet2: A high-definition map framework for the future of automated driving,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1672–1679, doi: 10.1109/ITSC.2018.8569929.

[26] E. Knabe, “SimS. Simulation Scenarios. Public report,” 2019.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” no. CoRL, pp. 1–16, 2017.

[28] T. Menzel, G. Bagschik, and A. M. Maurer, “Scenarios for Development, Test and Validation of Automated Vehicles,” *IEEE Intell. Veh. Symp. Proc.*, vol. 2018-June, no. Iv, pp. 1821–1827, 2018, doi: 10.1109/IVS.2018.8500406.

[29] 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, doi: 10.1109/ACCESS.2021.3053159.

[30] M. Fallgren *et al.*, “Multicast and Broadcast Enablers for High-Performing Cellular V2X Systems,” *IEEE Trans. Broadcast.*, vol. 65, no. 2, pp. 454–463, 2019, doi: 10.1109/TBC.2019.2912619.