National Physical Laboratory

NPL REPORT MS 55

Standardisation of Vehicle Simulation Interfaces for VTE

Bartlomiej Cieszynski

APRIL 2024

Standardisation of Vehicle Simulation Interfaces for VTE

Bartlomiej Cieszynski

Informatics Group
Data Science Department

ISSN 1754-2960

https://doi.org/10.47120/npl.MS55

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

Approved on behalf of NPLML by Maurizio Bevilacqua,

Senior Engineer

EXECUTIVE SUMMARY

This report acts as a stepping stone towards the standardisation of road-based Connected Autonomous Vehicle (CAV) development, to enhance the progress towards an Intelligent Transportation System (ITS). It provides an overview of simulation Virtual Test Environments (VTEs), simulation methods, and currently used Vehicle to Vehicle (V2V) and Vehicle to Everything (V2X) communication technologies and protocols, with an emphasis on their associated interfaces. A brief statement of Validation and Verification (V&V) methods is presented followed by an introduction to the current standardisation efforts in the CAV and V2X area.

The simulation overview consists of the application of Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL), and Hybrid Vehicle-in-the-Loop (ViL) approaches as well as a description of a VTE. Each simulation method is summarised through a description of a widely used case study; for instance, simulation software (LGSVL) composed of a simulation engine (Gazebo) and a system operation framework (ROS) for the case of SiL. The simulation methods are linked to an appropriate stage of CAV and V2X technological development via consideration of system requirements and project resource allocation, such as cost and time.

SiL methods emphasise the development of software, focusing on the development of independent vehicle decision making (AD Stack software) and transferring information between vehicles, pedestrians, and infrastructure. HiL focuses on the validation and development of hardware by inputting simulation data into the components such as sensors, integrating the virtual and real-world, allowing for more flexible and robust testing. Hybrid methods such as ViL combine both SiL and HiL implementations to test a range of simulated scenarios in a controlled real-world environment.

This report focuses on two approaches to V2X communications, the Dedicated Short Range Communication (DSRC) and Cellular Vehicle to Everything communication (C-V2X). DSRC is primarily used for short range quick communications, defined by the IEEE 802.11p protocols, however, does not extend to external networks. C-V2X is based on cellular technologies developed by 3GPP. The application of 4G Long Term Evolution (LTE) or 5G New Radio (NR) technologies allows for both direct and external network communications; however, both struggle to overcome latency issues generated by signal band overloads.

The goal of this report is to provide a clear explanation of the current state of simulation methods and V2X technologies. It also aims to highlight the interfaces associated and the necessity of introducing standardisation procedures for CAV and V2X development.

GLOSSARY

Terms and Abbreviations			
Access Point	AP	Packet Data Convergence Protocol	PDCP
Application Programme Interface	API	Physical	PHY
Artificial Intelligence	AI	Physical Broadcast Channel	PBCH
Association for Standardisation of Automation and Measuring Systems	ASAM	Physical Downlink Control Channel	PDCCH
Automated Driving System	ADS	Physical Downlink Reference Signal	PDSCH-DMRS
Automatic Repeat reQuests	ARQ	Physical Downlink Shared Channel	PDSCH
Autonomous Driving	AD	Physical Layer Convergence Procedure	PLCP
Autonomous Vehicles	AV	Physical Layer Management Entity	PLME
Basic Service Set	BSS	Physical Medium Dependent	PMD
British Standardisation Institute	BSI	Physical Random Access Channel	PRACH
Cellular Vehicle-to-Everything	C-V2X	Physical Sidelink Broadcast Channel	PSBCH
Channel Impulse Response	CIR	Physical Sidelink Control Channel	PSCCH
Channel Transfer Function	CTF	Physical Sidelink Feedback Channel	PSFCH
Connected Autonomous Vehicles	CAV	Physical Sidelink Shared Channel	PSSCH
Control Channel	CCH	Physical Uplink Control Channel	PUCCH
Controller Interface Device	CID	Physical Uplink Reference Signal	PUSCH-DMRS
Cyclic Prefix	CP	Physical Uplink Shared Channel	PUSCH
Dedicated Short Range Communication	DSRC	Quality of Service	QoS
Distribution System	DS	Radio Frequency	RF
Downlink	DL	Radio Link Control	RLC
Downlink Control Information	DCI	Radio Resource Control	RRC
European Telecommunications Standards Institute	ETSI	Roadside Unit	RSU
Evolved Node B	eNB	Robot Operating System	ROS
Extended Service Set	ESS	Second Generation CID	CID II
Field of View	FOV	Service Channel	SCH
Frequency Range	FR	Shared Device Access Protocol	SDAP
Generation NodeB	gNB	Sidelink	SL
Graphical User Interface	GUI	Sidelink Control Information	SCI
Hardware-in-the-Loop	HiL	Sidelink Primary Synchronisation Signal	S-PSS

High Definition	HD	Sidelink Secondary Synchronisation Signal	S-SSS
High Definition Render Pipeline	HDRP	Society of Automotive Engineers	SAE
High Level Architecture	HLA	Software-in-the-Loop	SiL
Hybrid Automatic Receive Request	HARQ	Standalone	SA
Independent BSS	IBSS	Station Device	STA
Input/Output	I/O	Synchronisation Block	SSB
Institute of Electrical and Electronics Engineering	IEEE	System Under Test	SUT
Intelligent Transportation System	ITS	The 3rd Generation Partnership Project	3GPP
International Organisation for Standardisation	ISO	The Institute of Electrical and Electronics Engineers Standards Association	IEEE-SA
Internet of Things	IoT	Ultra-Reliable Low-Latency Communication	URLLC
Internet Protocol version Six	IPv6	Uplink	UL
Joint Architecture for Unmanned Systems	JAUS	User Equipment	UE
Logical Link Control	LLC	Validation and Verification	V&V
Logical Protocol Data Unit	LPDU	Validation, Verification, and Uncertainty Quantification	VVUQ
Long Term Evolution	LTE	Vehicle Dynamics Model	VDM
Master Information Block	MIB	Vehicle-in-the-Loop	ViL
Media Access Control	MAC	Vehicle-to-Everything	V2X
Model-in-the-Loop	MiL	Vehicle-to-Infrastructure	V2I
Multimedia Broadcast Multicast Services	MBMS	Vehicle-to-Network	V2N
New Radio	NR	Vehicle-to-Pedestrian	V2P
Non-Standalone	NSA	Vehicle-to-Vehicle	V2V
Onboard Unit	OBU	Virtual Test Environment	VTE
Open Dynamics Engine	ODE	Visual Line-of-Sight	VLOS
Open Systems Interconnection	OSI	WAVE Basic Server Set	WBSS
OpenGL Utility Toolkit	GLUT	WAVE Short Message Protocol	WSMP
Operational Design Domain	ODD	Wireless Access for Vehicular Environments	WAVE
Orthogonal Frequency-Division Multiplex	OFDM	World Magnetic Model	WMM

CONTENTS

1 INTRODUCTION

Connected Autonomous Vehicles (CAVs) have been an active area of research within the past few decades. The possibility of reducing road risk, improvement of traffic management and significantly increasing ease of access is an exciting prospect for both industry and governing bodies. Autonomous Vehicles (AVs) are the foundation of building an Intelligent Transportation System (ITS), an Internet of Things (IoT) connecting vehicles, pedestrians, and infrastructure in a 'smart' manner. The automation of such an essential aspect of today's society however, requires incredible levels of system and hardware integrity, reliability and upkeep. Current research seeks to solve problems revolving around sensor quality and attenuations due to external factors, or communication protocol Quality of Service (QoS) and latencies. The connection of vehicles allows them to overcome Visual Line-of-Sight (VLOS) and range limitations of on-board sensors for obstacle detection. This generates demand for the development and improvement of Vehicle-to-Vehicle (V2V) and Vehicle-to-Everything (V2X) communication methods, to enable collaborative perception between connected vehicles.

The current most utilised approaches to V2X include the Dedicated Short Range Communication (DSRC) and Cellular-V2X (C-V2X). The DSRC protocol used for inter-vehicular communication is standardised across several layers, part of a system architecture responsible for certain functions, and specifications. The Physical (PHY) and Media Access Control (MAC) layers follow the Institute of Electrical and Electronics Engineers (IEEE) 802.11p [1] standard, adapted for automotive communications from previously defined WiFi protocols. The network and security protocols come from the Wireless Access for Vehicular Environments (WAVE) protocols consisting of IEEE 1609.x [2] and Society of Automotive Engineers (SAE) J2735 [3] standards. In Cellular-V2X (C-V2X), the application of cellular bandwidths and communication technologies is dictated by the 3GPP releases for 3G, 4G Long Term Evolution (LTE) and most recently 5G New Radio (NR) V2X [4]. The development of the specifications set by 3GPP over the past years can be seen Figure 1 below. Since the DSRC protocols are an extension of the IEEE 802.11 WiFi protocols, amendments to the these protocols are usually published with respect to WiFi rather than V2X applications. A recent summary of advancements prioritising the application to V2X technologies can be found in the IEEE 802.11bd-2022 protocols [5].

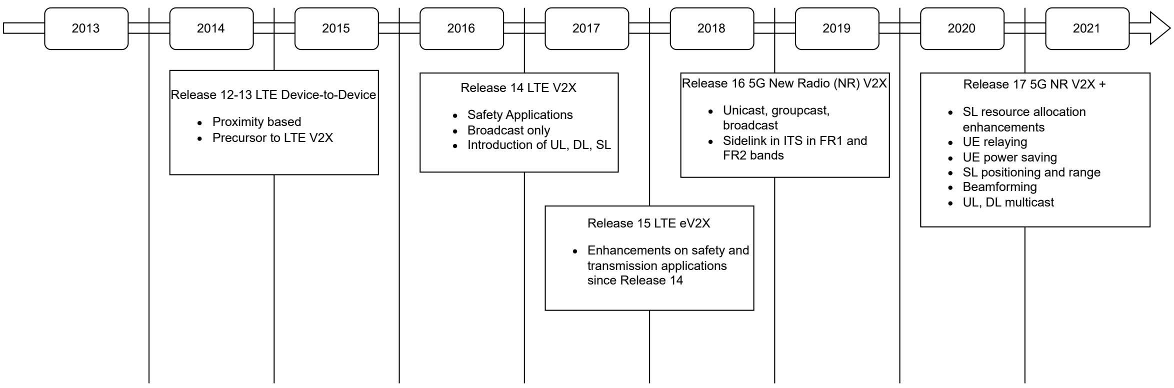


Figure 1: Diagram displaying the timeline of C-V2X development specified by the 3GPP Releases. Each box summarising key concepts discussed within the release. Inspired by [6].

Deploying autonomous transportation systems and intelligent infrastructure necessitates rigorous validation of the software and hardware to ensure safety and reliability. This drives a need for the standardisation of development and testing procedures across the industry, including Validation and Verification (V&V) methods to provide certainty and confidence in these systems before real-world operation. This report will provide an introduction to simulation Virtual Test Environments (VTEs), Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL), and Hybrid Vehicle-in-the-Loop (ViL) simulation methods; providing case studies of currently used and accepted software, hardware, and approaches. The report will then provide an overview to the technologies utilised in DSRC and C-V2X by providing a description of protocol architecture and responsible channel interfaces. A brief introduction to V&V methods will be provided along with a discussion of the current state of standardisation within CAV and V2X development. Both V2X and CAV technologies are regulated by standardisation bodies such as the British Standardisation Institute (BSI), International Organisation for Standardisation (ISO), and the European Telecommunications Standards Institute (ETSI).

2 AUTONOMOUS VEHICLE STATE OF THE ART

In recent years, CAVs have emerged as the fundamental technology required for the development of an ITS. The desire for creating a ‘smart’ transportation infrastructure has generated a huge demand for advancements in sensor technologies, artificial intelligence, and communication systems among both industry leaders and academia.

2.1 Simulations

The appeal of the ITS lies in its potential to address road safety, drive economic advancements and mitigate environmental impacts. The unification and automation of transportation would give the ability to anticipate and avoid accidents caused by interactions with pedestrians and infrastructure. However, integrating extensive automation into critical infrastructure systems necessitates extremely high standards for software and hardware integrity, security, and traceability. For autonomous vehicles especially, which could profoundly impact daily life, this is imperative for industry-wide standardisation of development, testing, and validation procedures to assure safety and reliability. The quality and robustness of all components used in autonomous systems require certification through rigorous testing and simulation-based on Validation and Verification (V&V) methods discussed in section 2.2.4. The following subsections will discuss Software-in-the-Loop (SiL), Hardware-in-the-Loop (HiL), and Hybrid Vehicle-in-the-Loop (ViL) simulations as stages of CAV and V2X development, followed by a description of the Virtual Test Environment (VTE) in which components are simulated.

2.1.1 Software-in-the-Loop

In simulations, the term ‘in-the-loop’ refers to incorporating a particular component of the developed product inside a simulation. In the case of SiL, it refers to the initial designing of sensor, engine, and autonomous driving (AD) software and interfaces which all need to collaborate in a quick, efficient and reliable manner. The software, also referred to as an AD Stack, enables the AV to perceive the environment, make decisions and navigate without human intervention. It includes control systems, mapping and other algorithms such as machine learning or sensor fusion. Developing these components and the interfaces between them is primarily conducted within SiL simulations and with the use of open-source simulator software such as LG Simulator (LGSVL) [7] and CARLA [8], or commercial simulators such as rFpro [9] and NVIDIA Drive Constellation [10]. Simulators operate and utilise framework systems such as the Robot Operating System (ROS) [11] through the use of physics engines such as Unreal Engine [12] or simulation engines such as Gazebo [13], and mapping software such as OPENDRIVE [14]. The simulators are typically accessible through Python or C++ application programme interfaces (APIs) [15]. The combination of the above listed software is used for generation of SiL simulation frameworks for AV testing [16].

LGSVL is a simulator tailored for development of software systems due to its ability to generate high definition sensor output data and VTEs [15], which enables high confidence testing. Additionally, it also has applications to HiL and V2X simulations which will be further discussed in sections 2.1.2 and 2.2. It utilizes Unity's game engine [17] and High Definition Render Pipeline (HDRP) technologies for simulation, and is capable of generating photo-realistic VTEs by simulating fine tuned detail such as high definition (HD) mapping, traffic or weather conditions throughout time-of-day [7]. It contains a communication bridge which enables direct communication with the AD system tested, (AD stack), and additionally is capable of integrating with open-source AD system platforms such as Apollo [18] or Autoware [7][19]. A flow diagram representing the steps incorporated in the use of LGSVL have been outlined in the Figure 2 below. Specific LGSVL workflows as well as possible application to component modelling and simulating scenarios are available and covered within [7].

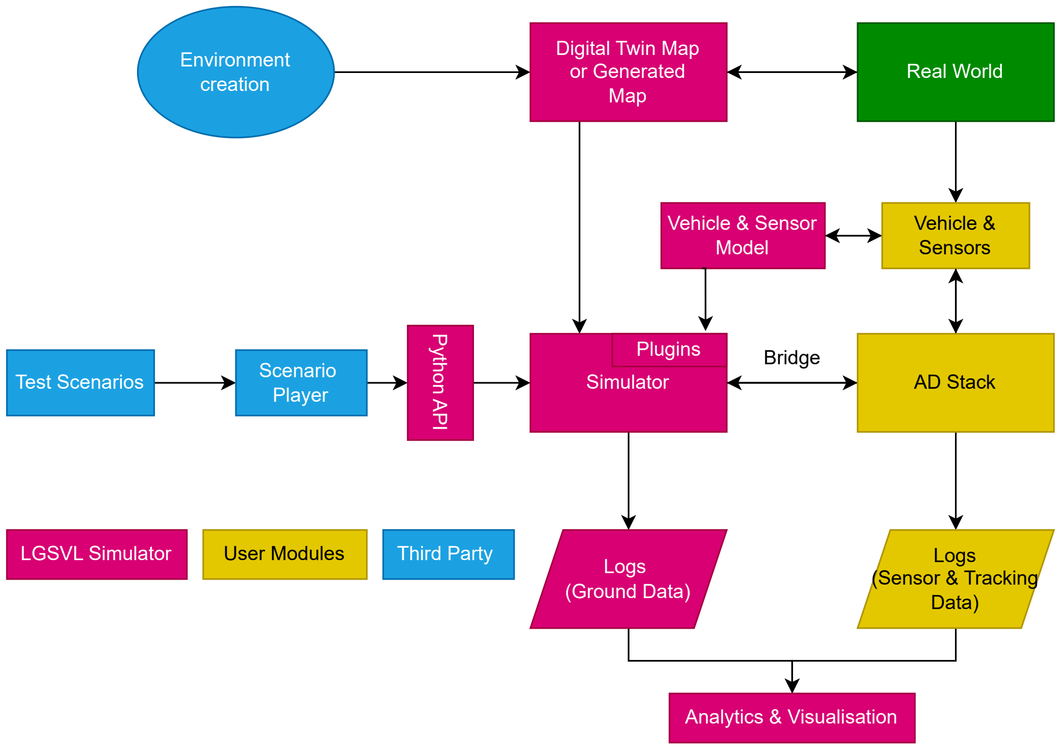


Figure 2: A flow chart representing the procedures used by the LGSVL Simulator. The diagram shows at which stage certain aspects of the VTE are adjusted and which subsystems are directly linked. Inspired by [7]

AD stacks are complex and rapidly evolving, developing them therefore requires integrating a myriad of fundamental software interfaces, spanning driver-level control, to machine learning components. The depth and range of topics exceed realistic capabilities for individual researchers or projects, hence, creating a demand for a framework for robotic software development such as ROS, in both industry and academia. The design of ROS can be summarised by a set of goals: Peer-to-peer, Tools-based, Multi-lingual, Thin, Free, and Open-Source [11]. Consisting of a number of different hosts, and hence a peer-to-peer topology, allows the system to avoid slow data traffic flows from a central server in a heterogeneous network. The topology requires lookup mechanisms which allow for interfacing during runtime, which is performed by a large number of tools within ROS. The use of said tools extends to all operations within ROS such as measuring bandwidth utilisation, plotting message data, and auto-generating documentation among others [11]. The operations within ROS are language-neutral and can be supported by various languages such as C++, Python, Octave and LISP. The messaging layer is configured by XML-RPC, which can be integrated easily into other languages. Since drivers used within software developing frameworks may be useful or reusable elsewhere, ROS implements a 'thin' ideology, where all algorithms are placed within libraries called upon execution, allowing for easier extraction of code for other purposes [11]. Lastly, debugging of stacks implemented through ROS has been facilitated by making it publicly available, allowing for simultaneous debugging of software and hardware components.

The development of AD stacks built on operation frameworks such as ROS occurs within simulation software which use physics engines, such as Unreal, or simulation engines, such as Gazebo. Gazebo is a 3D simulation engine, capable of simulating vehicles in a robust and highly photo-realistic VTE through the Open Dynamics Engine (ODE) [20]. The VTE closely resembling the real world visually improves the accuracy of visual camera simulation, however, that doesn't mean that it is such a good representation in other parts of the electromagnetic spectrum required for other sensors such as LiDARs or RADARs. To integrate AD stacks into simulator engines, specific interface packages, such as gazebo_ros_pkgs in the case for ROS and Gazebo, must be implemented for configuration [15]. Gazebo is a popular choice among simulation engines as it is capable of simulating dynamic environments and gives control of virtually every aspect within the simulation. However, it struggles to create large and complex environments due to lack of use of newest technologies in modern engines such as Unreal or Unity [7] [15]. The Gazebo architecture consists of a physics engine, visualisation software and world frameworks. It utilises the ODE which includes features such as collision detection, mass and rotational functions, and use of numerous joints [13]. The engine and visualisation interfaces cooperate together to generate the VTE. The control over a simulation requires either a User Interface (UI), for the case of Gazebo provided by OpenGL Utility Toolkit (GLUT) [21], or an API script. A model, any object that maintains physical representation composed of a rigid body, joints, sensors and interfaces to facilitate the flow of data [13], interacts with external interfaces through the UI and APIs to run the simulation. The use of external interfaces is flexible as numerous drivers such as Player's Vector Field Histogram [22] or Adaptive Monte-Carlo Localisation [23] may be integrated within the VTE. Case studies, limitations and precise details on Gazebo design may be found within [13].

2.1.2 Hardware-in-the-Loop

HiL simulations are the secondary step in the development of AVs. Once the desired AD stack has been approved via V&V procedures, later discussed in section 2.2.4, the next step is to integrate it with hardware. Rather than focusing on the development of software, HiL simulations focus on Systems Under Test (SUTs) which can refer to any component of a system in the context of quality testing, such as an onboard sensor. In comparison to SiL, rather than testing software, HiL simulations ensure AV hardware components such as sensors operate correctly and within the defined standards. HiL simulations still partly take place on simulator software such as LGSVL or other platforms commonly developed on ROS and Gazebo [24], however, rather than conducting everything virtually, the simulator is connected and inputs data into a real external SUT as depicted within Figure 3 below. The set up of the simulation causes the SUT to not be able to differentiate between a real and a simulated input, allowing it to feedback into the simulation as it would in a real AV [7]. The following paragraphs will briefly discuss the aims and architectural structure of HiL simulations followed by a description of hardware to simulation interfaces.

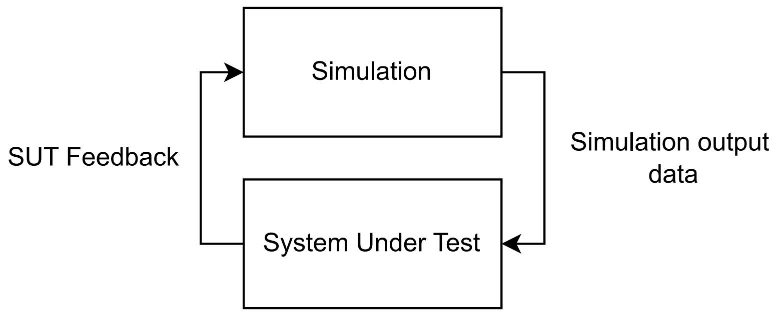


Figure 3: A simple diagram depicting the structure of a HiL simulation. Inspired by [25]

HiL simulations aim to observe hardware behaviour such as data inputs/outputs and execution upon integration with an AD stack [25]. There are two main approaches to HiL simulations; Open and Closed loop. Open loop simulations run independently without using previously logged data from the SUT. While simulation data is archived for future analysis, it does not get fed back to influence subsequent test runs [25]. In contrast, closed loop simulations directly utilize prior SUT output data to inform calculations in subsequent iterations, creating a real-time feedback loop [26] [27]. When designing HiL simulations it is vital to ensure it accepts a variety of SUT configurations, can perform both closed and open simulations, and that it is open and scalable. The simulator should flexibly integrate with varying SUTs, and scale without needing entire redesigns for minor system changes, ensuring affordable enhancements [25]. SUT complexity can vary from individual components such as RADARs, LiDARs, or cameras, to full vehicles as discussed in section 2.1.3 [24]. The architectural frameworks of HiL are closely related to those developed for the purpose of SiL. Details of such frameworks can be viewed in the works of [24], [25], and [27].

Since the simulation frameworks for HiL and SiL are similar, the major challenge when developing a HiL simulation is how to integrate the hardware into the simulator. The hardware and simulator require an interface which will allow for the transmission of data inputs and outputs both in Open and Closed loop simulations. To account for this bridge, a Controller Interface Device (CID) needs to be implemented within the loop [26]. The speed and quality of CIDs can be classified via generations. First generation CIDs use slow speed communication interfaces such as RS-232 (standard serial communication) and tend to implement only 32 or fewer input/output (I/O) channels [26]. On the other hand, second generation CIDs (CID II) utilise a motherboard-daughterboard design built around a microcontroller linked through USB as its data communication interface [26]. Increasing to 64 I/O channels through a 24/12 volt interface allows for high speed real time communication of large datasets which are required in HiL simulations. CIDs mediate the interface between hardware and simulations, however, they are not capable of directly communicating with simulations models which must be accounted for through in-built simulator packages. CID to simulator interface software must be capable of reading, writing, formatting, synchronising, and transmitting data and must consider issues such as communication initialization, information configuration and error handling. Details of these considerations have been explained within [26] for the case of CORSIM [28], an alternative simulator.

2.1.3 Vehicle-in-the-Loop

Vehicle-in-the-Loop (ViL) simulations, a form of Hybrid simulations, are an advanced extension of a HiL simulation which allow for the addition of physical testing. As mentioned previously an SUT can consist of any number of components working together, hence setting up an entire vehicle as the SUT is one of the final stages of AV simulation testing. The test incorporates a real vehicle connected to a model of itself generated within a VTE; where the modelled vehicle encounters obstacles or specific road scenarios and sends data to the real vehicle which then responds accordingly. ViL tends to be the shortest and final stage of AV testing and validating due to its related expenses and requirements.

Similar to HiL, ViL takes place both in the real world and the virtual. A ViL testbed consists of a VTE, a vehicle model, a physics engine, sensor models, visualization tools, interface tools and code generation capability [24]. Some of these components such as a physics engine, visualisation and interface tools and VTEs are embedded within simulation software such as LGSVL. Once again, similar simulation software, such as those created on ROS and Gazebo through the control of APIs, may be used for the virtual aspect of the testing. The complexity of ViL simulations stem from the incorporation of a precise vehicle model and the VTE. The vehicle's general structure such as visual geometry, configuration and connection of parts, and specification of sensor placement is often represented in the Unified Robot Description Format, utilising XML Macros [24]. External factors such as kinematic and dynamic constraints, or other physical attributes are defined within the VTE through the simulation or physics engine. A model of a vehicle must account for factors such as runtime solvers, behaviour of kinematic robotic simulation, and the validity of vehicle component dynamics in comparison to the real world [24]. Solutions to specific scenarios within the considered factors, ViL framework, as well as examples of vehicle, sensor and environment models have been presented and explained in depth in [24] and [29].

An entire vehicle is a very complex combination of software and hardware, so specific interfaces allowing for clear control and communication between components is necessary. The interfaces also depend on the method chosen for the ViL simulation testing. For ViL to take place, a real vehicle must be connected and receive inputs from a simulation. It must also act upon these inputs by adjusting acceleration or path. To do this, however, it needs to be within a safe environment that allows for the vehicle to react without creating to any hazards. such an environment can be implemented either through a safe training ground, such as a closed off training course, or through a roller test bench. The specific integration of a roller test bench into the ViL has been explained in [29], where the simulation framework for ViL consisting of SysML [30] and the Contact and Channel Model [31] for definitions of structure and behaviour have been used. In contrast, a real environment approach has been used in [24]. To interface the vehicle control and data transmission within the loop without the use of a test bench, [24] used the Joint Architecture for Unmanned Systems (JAUS) [32]. The integration of JAUS into simulation and operation systems requires implementation of a communication bridge, such as the ROS/JAUS interface [33]. Both of these approaches require specific adaptations to the VTE in which the virtual vehicle is placed.

2.1.4 Virtual Test Environments

The needs of the different aspects of AV development need to be considered during the development of a VTE. The simulations need to be able to recreate specific conditions or scenarios for AD stack or SUT testing. The adaptation of those conditions to test specific components is conducted through the adjustment of the VTE and the models within it. Models for traffic, pedestrians or weather conditions are very commonly adjusted through the use of VTE interfaces to allow for niche testing of components.

The VTEs need to be tailored to the needs of the component being tested, not just to ensure the validity of the simulation but also allocate project resources efficiently. For example, testing of software functionality of path planning during the SiL stage, does not require robust graphical representation of vehicles or pedestrians. Instead higher accuracy of object dynamics should be considered along with the presence of an obstacle [34]. A highly detailed graphical representation of the environment might however be required during testing of the full AD stack. Common examples of adjustable components which need to be controlled for testing include dynamic objects such as pedestrians, animals and vehicles; terrain, roads, buildings, intersections, and trees; and physical phenomena such as gravity, air density and pressure, or other weather conditions [34]. The dynamics and kinematics of bodies are expressed through the Vehicle Dynamics Model (VDM) which typically consists of six degrees of freedom: position, orientation, linear velocity, linear acceleration, angular velocity and angular acceleration [35] [36]. Extensions to modelling of dynamic objects such as pedestrians and vehicles are found within simulator libraries. For instance, CARLA contains an asset library which specifies building, vehicle or pedestrian details. An asset library containing vehicle models with a variety of inertial properties and powertrains, allows the dynamics of an individual vehicle to be simulated more accurately [34]. Specific pedestrians can also be modelled dynamically via considerations of age, size, or health to increase the accuracy of the simulation for sensor testing [8] [37].

To generate desired terrain, specific models such as the Digital Terrain Model [38] or the parametric geometry model used in OPENDRIVE [39] need to be implemented into the physics engine. Physical phenomena incorporated through models to allow for real-time HiL operations are also a key component of the simulation [36]. Properties such as gravity, magnetic fields or air pressure and density can be integrated through the GRACE model [40], World Magnetic Model (WMM) [41], and the 1976 U.S. Standard Atmosphere Model [42] respectively. Additionally, the VTE aims to control other parameters such as lighting or solar angles. More general conditions such as rain may also be implemented through physical models, based on reflection of light from individual raindrops, as described within

[43], or data driven models based on data from real events such as storms. Additional concepts such as collision detection, failure simulation or sensor noise may be implemented into simulation [34].

Adjusting VTEs and implementing previously stated models into simulation requires a communication bridge with the physics engine. This communication generally can be conducted via the use of a graphical user interface (GUI) [44] or through direct communication with the software in the form of MATLAB, Python, or C++ APIs. The GUI allows for easy and direct communication with the physics engine to make any adjustments to the VTE of modelling. An example of a GUI can be seen in the NVIDIA Opens DRIVE Constellation Platform [45]. The GUI allows direct adjustments of scenario, vehicle, and world modelling as depicted in figure 4 below. Although the GUI allows for VTE manipulation with ease, it is limited to the factors included by the developers. Any modelling not included within the GUI has to be adjusted through the use of APIs. Some engines such as Unity [17] accept inputs both from UIs and APIs. The setup of Unity allows for seamless integration of the use of APIs and GUI through a direct interface of C# scripts to objects. The scripts defining models, behaviour or interactions may be attached to objects within the editor. The GUI then provides a visual representation of the components, allowing for the building of a dynamic VTE [46].

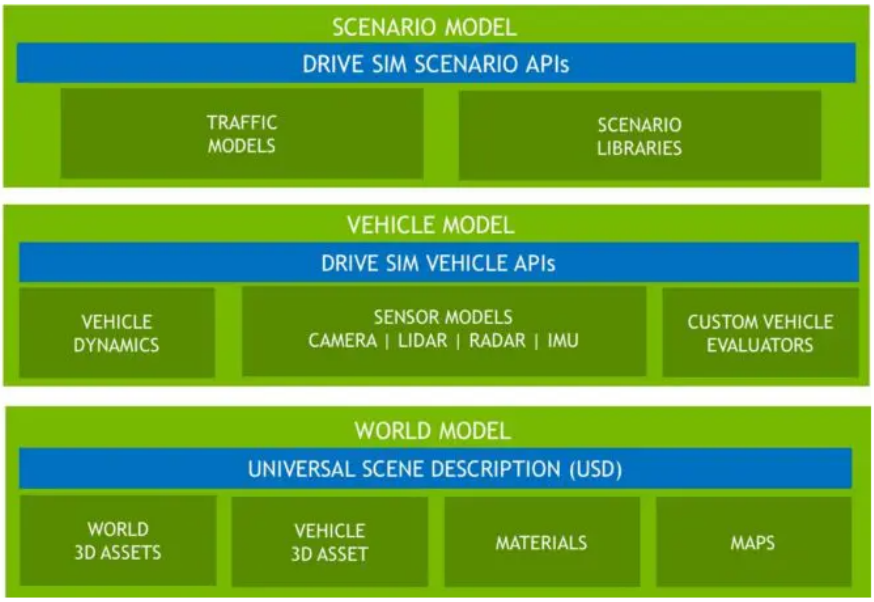


Figure 4: GUI in the NVIDIA Opens DRIVE Constellation Platform. [45]

2.2 Intelligent Transportation System

Development of CAVs theoretically enables a safe and reliable automated mode of transport. The perception and field of view (FOV) of the vehicle may be limited due to obstruction of the Visual Line-of-Sight (VLOS) of the sensors by vehicles, buildings or terrain. One approach to overcome this sensor limitation is to integrate data from numerous vehicles. The combination of data would generate an extensive dynamic map of the environment, allowing vehicles to not be limited by obstructions or the range of sensors. This approach can be further extended to all aspects of transportation. Communication of vehicles with other vehicles, infrastructure, or pedestrians would create an Internet of Things (IoT) and allow for its micromanagement - an Intelligent Transportation System (ITS). The data transmission requires specific communication methods such as the Dedicated Short Range Communication (DSRC) or the Cellular Vehicle to Everything (C-V2X). To ensure the system's reliability and safety, the communication channels need to meet specific latency and Quality of Service (QoS) standards set by the International Organization for Standardisation (ISO), The British Standards Institute (BSI), and The Institute of Electrical and Electronics Engineers Standards Association (IEEE-SA); later discussed in Section 3.

2.2.1 Dedicated Short Range Communication

Dedicated Short Range Communication (DSRC) is predominantly used for vehicle safety warnings and traffic management within the ITS and operates at the 5.9 GHz bandwidth [47]. It is a short range ($\sim$ 1km) and relatively simple form of communication focusing on Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I). The DSRC interface responsible for communication is based on the combination of a Physical (PHY) and a Media Access Control (MAC) layer based on the IEEE 802.11a protocols [1].

The DSRC architecture is based on the Open Systems Interconnection (OSI) model created by an ISO committee in 1979 [48]. It is a seven layer framework for multi-device communication protocols allowing for the standardisation of heterogeneous informatics networks. The architecture is composed of the Physical, Data Link, Network, Transport, Session, Presentation, and Application layers [48]. Within the DSRC, the layers of the OSI above the PHY and the MAC layers are defined by the Wireless Access for Vehicular Environments (WAVE) protocol [2] [49] [50]. The WAVE architecture is built on units installed within components one wishes to communicate with. For DSRC the communicating components include infrastructure, which contains Roadside Units (RSUs), and vehicles, which contain Onboard Units (OBUs). The units exchange information through a fixed radio channel called the Control Channel (CCH) and can create small IoT networks called WAVE Basic Server Sets (WBSSs) which can also extend and connect to a wide-area network [50]. Communication of units within a WBSS takes place through specific Service Channels (SCHs). WAVE operates using two communication protocols, the Internet Protocol Version Six (IPv6), and the WAVE Short Message Protocol (WSMP). WSMP is utilised to transmit short messages of certain parameters to ensure other components receive the signal in required time, however, is incapable of extending to networks which in turn is handled by the IPv6 protocol [50]. Implementation of the Transport, Application, and Presentation OSI layers as well as security services are specified by WAVE protocols. The overview of the DSRC structure can be found in Figure 5 below. Further details of the protocols within each layer, security or operations can be found within [50] and [51].

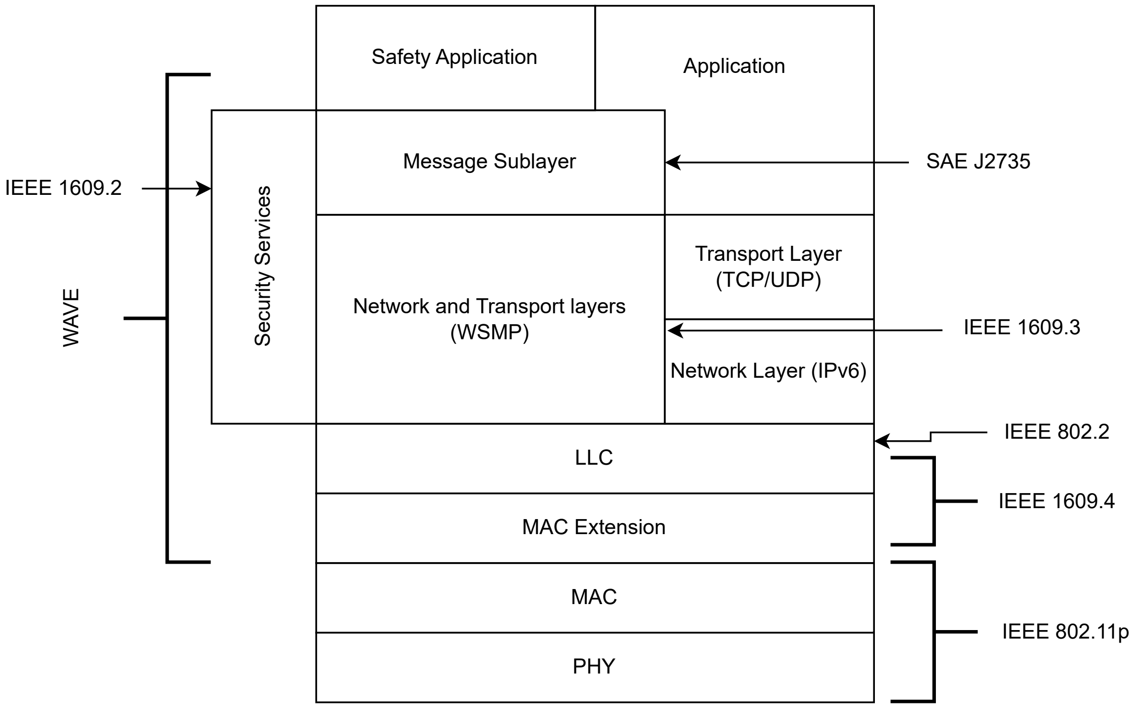


Figure 5: Diagram representing the architecture of DSRC used in the USA. The diagram builds up from the PHY layer and depicts when protocols are implemented. [52]

The starting point of the DSRC architecture consists of the PHY and MAC layers defined by 802.11a protocols [1] [50]. The PHY layer is the most fundamental component of the OSI model [48] as it is responsible for the transmission and reception of data as well as specifying the channel bandwidths or transmission power [50]. It can be further split into the Physical Medium Dependent (PMD) and the Physical Layer Convergence Procedure (PLCP) sublayers governed by clause 17 within IEEE 802.11 [52]. The PMD interfaces directly with the wireless medium through signal transmissions defined as multicarrier orthogonal frequency-division multiplex (OFDM) signals [53] based on the WAVE system [2]. The PLCP, on the other hand, defines the mapping between the MAC frame and the PHY layer data unit [52]. The management of the PHY layer is handled by a plane protocol called the Physical Layer Management Entity (PLME) [50]. Details of the PMD and the PLCP sublayers can be found within [52]. The specific PHY layer channel characteristics can be modelled through a multi-dimensional correlation function of the time-varying Channel Impulse Response (CIR) and its Fourier transform, the time-varying Channel Transfer Function (CTF) [54] [55]. The CIR and CTF are based on physical parameters of the signal including scattering models, path loss [56], Doppler attenuation, and others [57]. Further explanation of those parameters along with equations for the CIR and CTF, as well as a review of works discussing channel modelling, characteristics, and classification can be found within [55] and [58] with a range of supporting references. While the PHY layer is responsible for signal generation and reception, it requires an interface allowing it to communicate with other vehicles, i.e. the MAC layer.

The MAC layer is a part of the Data Link layer of the OSI Model, along with the Logical Link Control (LLC) sublayer based on IEEE 802.2 protocols [59]. The MAC layer is the interface responsible for distribution of OFDM signals, generated by the PHY layer, among 802.11 station devices (STAs) [52]. The architecture of the MAC layer (IEEE 802.11a) can be defined as a combination of service sets: the Basic Service Set (BSS), Independent BSS (IBSS), and the Extended Service Set (ESS) [51]. The IBSS is specifically defined as an ad-hoc network composed of STAs. The BSS utilises an Access Point (AP) which acts as a master STA, whereas, the ESS is a composition of two or more BSSs connected through a Distribution System (DS) [51]. The LLC sublayer is responsible for the correct allocation of signals between networks and devices. More specifically, it defines the Logical Protocol Data Unit (LPDU) transfer and error control on wireless links between DSRC units [60]. It controls the data flows between service APs through classification of unacknowledged connectionless modes (Type1) and acknowledged connectionless modes (Type3). Only Type3 modes support error control and sequence delivery through the use of stop-and-wait Automatic Repeat reQuests (ARQ) [61] for short messages within the DSRC [60]. The collaboration of the MAC service sets along with the allocation through the LLC sublayers form the fundamental communication interface within the DSRC. A more extensive description of the service set architecture of MAC and LLC protocols along with considerations of signal transmissions have been presented within [51] and [60]. The IEEE 802.11 [1], 802.2 [59], 1609 families [62], along with the SAE J2735 [3] are the protocols defining the WAVE system implemented in DSRC.

2.2.2 Cellular Vehicle to Everything

The Cellular Vehicle to Everything (C-V2X) communication method is a more recent approach to V2X technologies. Compared to DSRC, C-V2X is a more powerful form of data transmission, operating through recent technological advancements implemented in Long Term Evolution (LTE) 4G [63] or New Radio (NR) 5G [64] networks in the 5.9 GHz frequency band for V2X. C-V2X is better suited to communication with other components of the ITS such as V2P, V2I and V2N than DSRC, however, it struggles in terms of latency and server overloading. C-V2X operates on two interfaces based on 3GPP LTE Standards [65]; the PC5 [66] and the Uu interface [67] [68]. PC5 is the interface responsible for direct short range communications referred to as Sidelink (SL), and the Uu interface is responsible for the cellular network communications through Uplink (UL) and Downlink (DL) channels. Additionally, Uu can also be incorporated into SL transmissions as Mode 3 network-controlled modes [6].

C-V2X is a combination of PC5 and Uu interfaces, both responsible for transmissions between different components of the ITS. PC5 focuses on direct and quick communications, similarly to DSRC, by providing reliable dynamic information. It doesn't require a cellular network and focuses on transmissions between local wireless devices (V2V, V2I, and V2P) such as RSUs and OBUs, also referred to as User Equipment (UE) [66]. The Uu interface is mostly responsible for V2N communications which allow features such as network assisted driving or cloud computing [67]. With the development of 5G technologies, SL within NR V2X is capable of achieving Ultra-Reliable Low-Latency Communication (URLLC) latencies of 1 ms [69], enabling transmission of essential parameters within required time frames, hence, increasing and managing the Quality of Service (QoS) [6]. SL utilises two types of transmission mode, Mode 3 through the Uu interface and Mode 4 through the PC5 interface. Mode 3 is a network-controlled mode which assigns resources and parameters such as sub-channels through UL and DL transmissions. The allocations are assigned by LTE evolved Node B (eNB) and/or NR next Generation NodeB (gNB) base stations [70]. The parameters are then used by UEs directly to communicate between one another through the Uu interface [6]. Mode 4 is a distributed mode where UEs independently allocate and select their resources and transmission parameters based on channel conditions. The signal does not go through a base station but information of its usage may be reported to the network [6]. A visual representation of the communication links used within C-V2X has been presented in Figure 6 below. The following paragraphs will focus on elaborating on the architecture of PC5 and Uu interfaces.

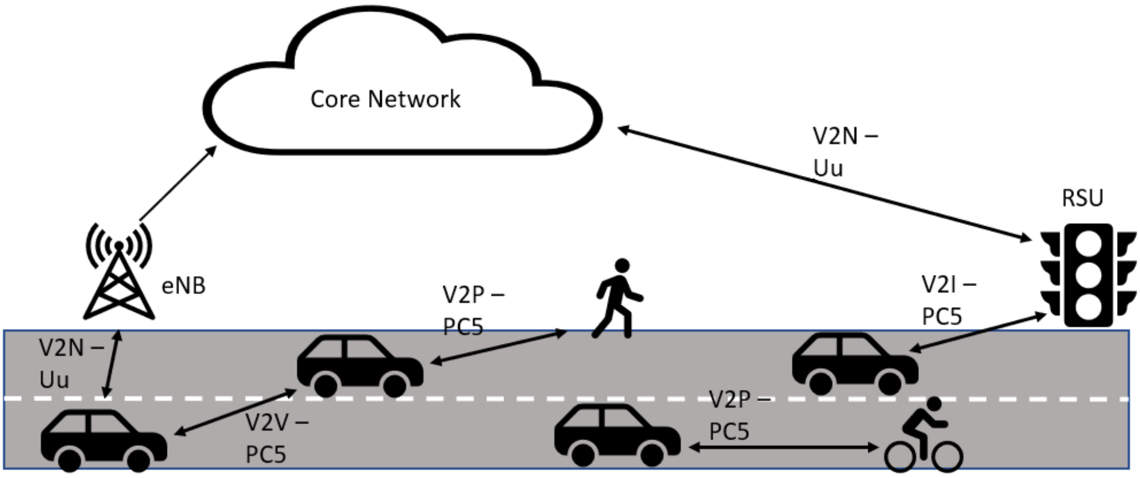


Figure 6: Diagram showing communication links used between various components of the ITS, along with associated interfaces [71].

NR V2X uses the 3GPP architecture built upon a Physical Layer operating at Frequency Range 1 (FR1) of 410 MHz - 7.125 GHz and Frequency Range 2 (FR2) of 24.24 GHz - 52.6 GHz [72] [73]. The transmissions use the OFDM waveform with a cyclic prefix (CP) and are organised within radio frames, each with a duration of 10ms [6]. It is designed to enable URLLC transmissions through utilising the PHY, MAC, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), Shared Device Access Protocol (SDAP), and Radio Resource Control (RRC) layers presented within Figure 7 below. The physical structure of NR V2X is composed of various signals and channels, more specifically the Physical Sidelink Broadcast Channel (PSBCH), the Physical Sidelink Feedback Channel (PSFCH), Physical Sidelink Shared Channel (PSSCH), and the Physical Sidelink Control Channel (PSCCH) [70]. The PSBCH transmits periodic Master Information Blocks (MIBs) which are responsible for UE-to-UE communication along with Sidelink Primary Synchronisation Signals (S-PSS) and the Sidelink Secondary Synchronisation Signals (S-SSS). The combination of the PSBCH, S-PSS and S-SSS forms the Signal Synchronisation Block (SSB) used in SL synchronisation [70]. The PSFCH was designed to transmit Hybrid Automatic Receive Request (HARQ) messages [74] between the receiver UE and the transmitter UE and occupies the last one or two slots of OFDM symbols [67] [70]. The PSSCH and the PSCCH collaborate on the transmission of Sidelink Control Information (SCI) messages. The first stage of the SCI message which includes frequency resource allocation and time resource allocation are sent through the PSCCH while the PSCCH directly linked with a corresponding PSSCH handles the second stage of the SCI message which includes HARQ process ID, Source ID, and Destination ID [70]. More detailed explanations of the communication channels, signal modelling, and 3GPP standardisation updates can be found in [6], [66], [67], [70], and [71].

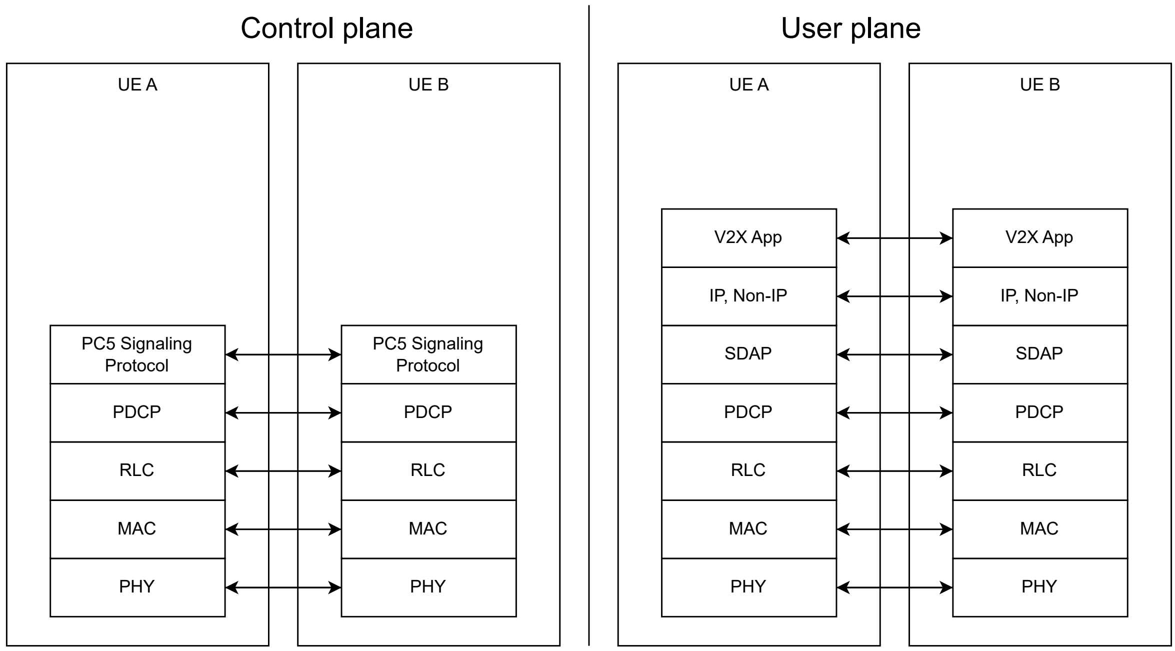


Figure 7: Structural overview of 5G NR V2X layer architecture. Inspired by [70].

Uu V2X communications are conducted through UL and DL transmissions under NR Non-Standalone (NSA) and Standalone (SA) deployments which can be sent through Multimedia Broadcast Multicast Services (MBMS) [6] [75]. They are responsible for signal transmissions between UEs and cellular base stations, gNBs for the case of NR 5G V2X. Similarly to PC5 and SL communication, specific physical channels are responsible for various forms of communications. The messages sent from UEs to gNBs transmit via the Physical Uplink Shared Channel (PUSCH) and are returned back via the Physical Downlink Shared Channel (PDSCH). This approach can be seen in the case of SL Mode 3 signals where base stations configure the periodicity of sub-channels using Downlink Control Information (DCI) through the PDCCH [6] [76]. Other parameters such as HARQ messages, scheduling assignments, and power control commands are exchanged between UEs and gNBs through the Physical Uplink Control Channel (PUCCH) and the Physical Downlink Control Channel (PDCCH). Specific signals such as channel states, system information such as cell identity or configuration parameters, or resource requests are transmitted through the Physical Downlink/Uplink Reference Signal (PDSCH-DMRS/PUSCH-DMRS), Physical Broadcast Channel (PBCH), and the Physical Random Access Channel (PRACH) respectively. Elaboration on channel functions, characteristics and modelling along with expansion on SL, UL, and DL modelling may be found within 3GPP specifications [6], [65], [70], [71], [75], [76], and [77].

2.2.3 Simulation Interfaces for Vehicle to Everything

The development of V&V for CAV consists of:

1. Model in the Loop (MiL) [78], the testing of channel or attenuation models as presented in [6];
2. SiL simulations, the development of AD stacks;
3. HiL simulations, the testing and validation of SUTs;
4. Hybrid simulations [79], the integration of simulation and real world environments and scenarios such as traffic [79] [80] or emergency procedures [81].

However, in comparison to the development of AD stacks, V2X validation may be conducted through two approaches. The signals generated within the PHY layer for both DSRC and C-V2X can be modelled and broadcasted within the simulation through the application of data transfer interfaces such as the IEEE 1609.4 protocol for DSRC and PC5 and Uu for 5G NR V2X. However, due to the complexity and advancement of technology from the 802.11 to 5G NR [82], the PHY layer of C-V2X is not commonly modelled directly. Similar models representing real world behaviour of 5G technologies may be implemented instead, depending on project resources. The complexity of directly simulating NR technologies has introduced the alternative approach to validation of V2X. Numerous connected VTEs generating output data may be run simultaneously through the use of a simulation platform based on the IEEE Standard for Modelling and Simulation - High Level Architecture (HLA) [83], through the application of APIs [81] [84]. The transmission of data outside of the simulation, removes the necessity of modelling NR technologies.

The validation process for DSRC communication methods involves transmitting simulated signal data from the 802.11p PHY layer across multiple devices under specific scenarios. The simulation primarily focuses on generating signal outputs from the PHY layer, which dictate properties such as transmission power or bandwidths as discussed in section 2.2.1, via application of a Radio Frequency (RF) model. The signal is then transmitted through the PLCP sub-layer interface into the Data Link MAC sub-layer which then distributes the data further into the LLC sub-layer in accordance with the IEEE 1609.x protocols [62], in particular the 1609.4 protocol [85]. The protocol incorporates a time-division scheme which facilitates the concurrent use of a range of server channels, allowing for simultaneous use of different applications. The data is transmitted by alternating the CCH and the SCH in Sync intervals in which either channel is utilised for 50ms, allowing for approximately ten intervals per second [86]. The simulation architecture for the implementation of DSRC technologies, specifically the IEEE 1609.4 protocol has been depicted in Figure 8 below. This architecture can then be extended to conduct V&V procedures, discussed in Section 2.2.4, at appropriate stages of development. Details of implementing various DSRC protocols and modelling within a range of simulators can be found in [85], [86], [87], and [88].

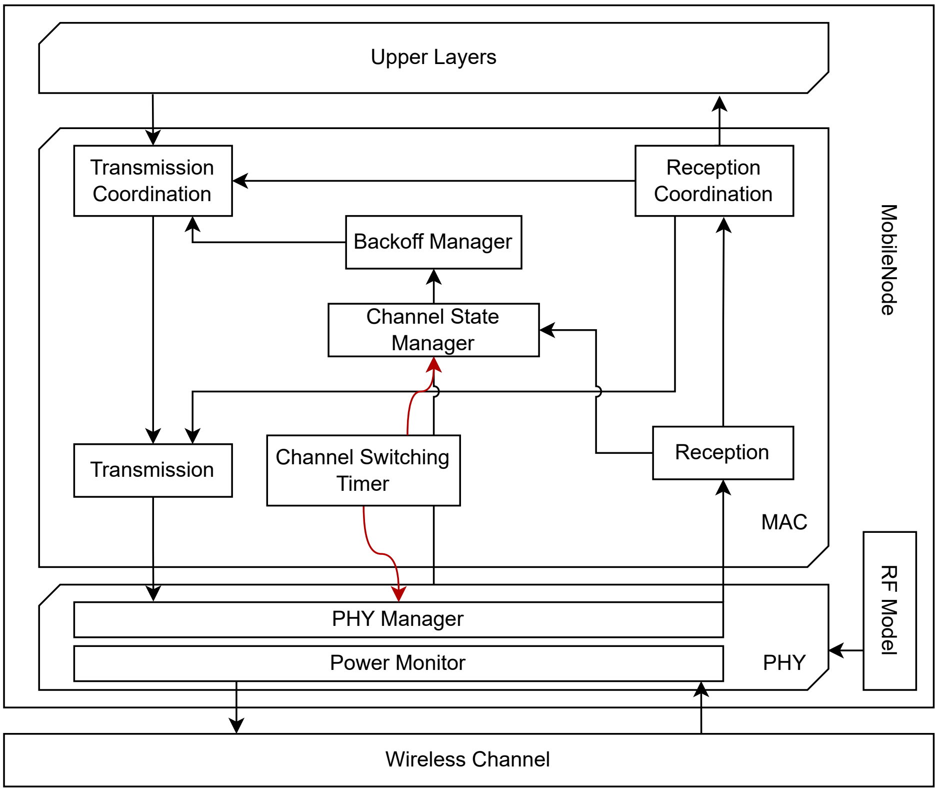


Figure 8: Diagram showing the architecture for DSRC IEEE 1609.x protocol simulations, outlining relevant interfaces and processes occurring within the MAC layer. Inspired by [86]

As stated previously due to the complexity of C-V2X technologies explicitly modelling NR communications technologies provided in the 3GPP releases [6] may prove challenging. However, depending on project time, budget, and computational power, one may want to directly simulate 5G technologies for a more robust and exact representation of the real world. Approaches to C-V2X simulation may be found in [6], [89], and [90]. An example of simulation architecture for channel signal modelling can be seen in Figure 9 below. Alternatively, simplified models closely resembling real world properties, as presented in [80], [91], [92], and [93], can be used instead, allowing for more flexibility and availability for smaller scale projects. A third approach to simulating in the development of C-V2X uses an integrated simulation platform - a comprehensive tool providing an integrated ecosystem for conducting, managing, analyzing, and visualizing multiple simulations. This allows for testing of communication methods externally between simultaneously running simulations and removes the necessity of simulating 5G technologies. Simulation platforms operate on frameworks such as the HLA, which sets the standards for cooperation between parallel running simulations [94]. Examples of utilising simulation platforms can be found in [90], [95], [96], and [97]. The application of each approach depends on project resources and may be selected accordingly.

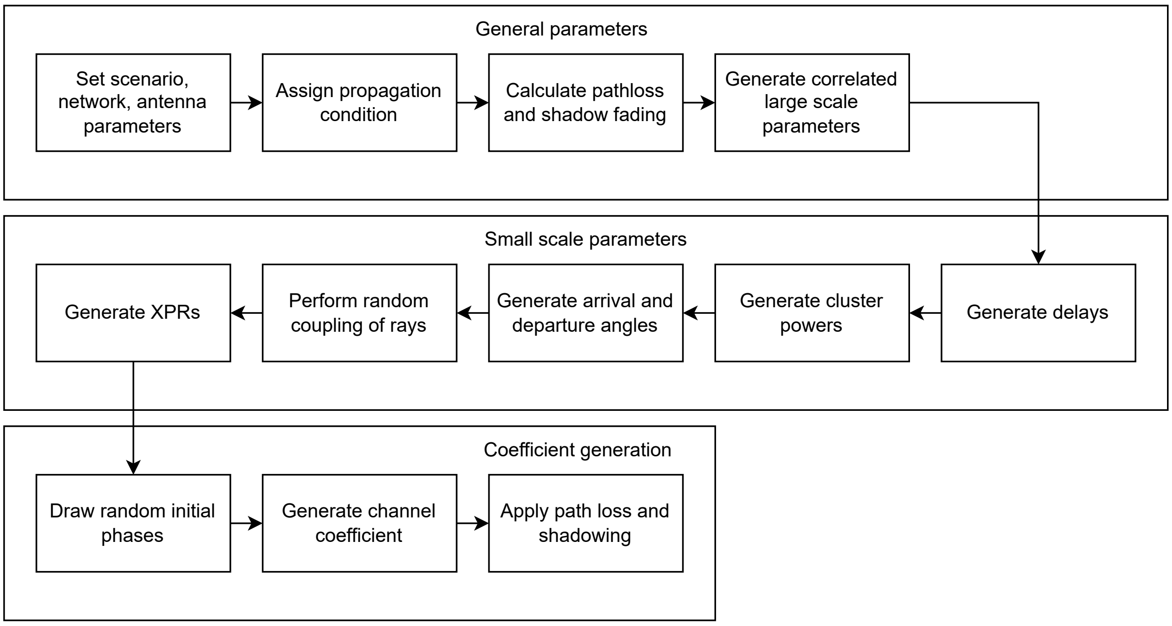


Figure 9: Flow diagram representing the architecture and process of implementing C-V2X, in particular 4G LTE and 5G NR V2X, channel modelling into a simulation. Inspired by [6].

2.2.4 Validation and Verification

Validation and Verification (V&V) is the process of warranting confidence and the integrity of an SUT or AD Stack. Generally, the validation refers to ensuring that modelling is an accurate representation of the real world, whereas, verification confirms that software and hardware align with expected models. In the context of system development, it can be considered as ensuring a component satisfies particular requirements for validation and required specifications for verification. The V&V framework can be extended further to account for uncertainty, forming the Validation, Verification, and Uncertainty Quantification (VVUQ) [98]. During the V&V procedures of a system, numerous tests are ordered such that they increase in complexity and in degree of relevance to the real world. Scenarios and requirements [99], defined by organisations such as the Association for Standardisation of Automation and Measuring Systems (ASAM), are initially tested virtually then integrated more with the real world. The V&V procedures vary depending on the component of the system being tested, for instance, software requirements may include the validation of decision making whereas hardware requirements may include sensor quality. The V&V process for CAV and V2X requires more robust procedures due to the introduction of the opaque nature of decision making by artificial intelligence (AI). A model, proposed by ISO, used for the development of V&V pyramids and procedures has been presented in Figure 10 below. More robust reviews of V&V methodologies may be found in the likes [100], [101], and [102].

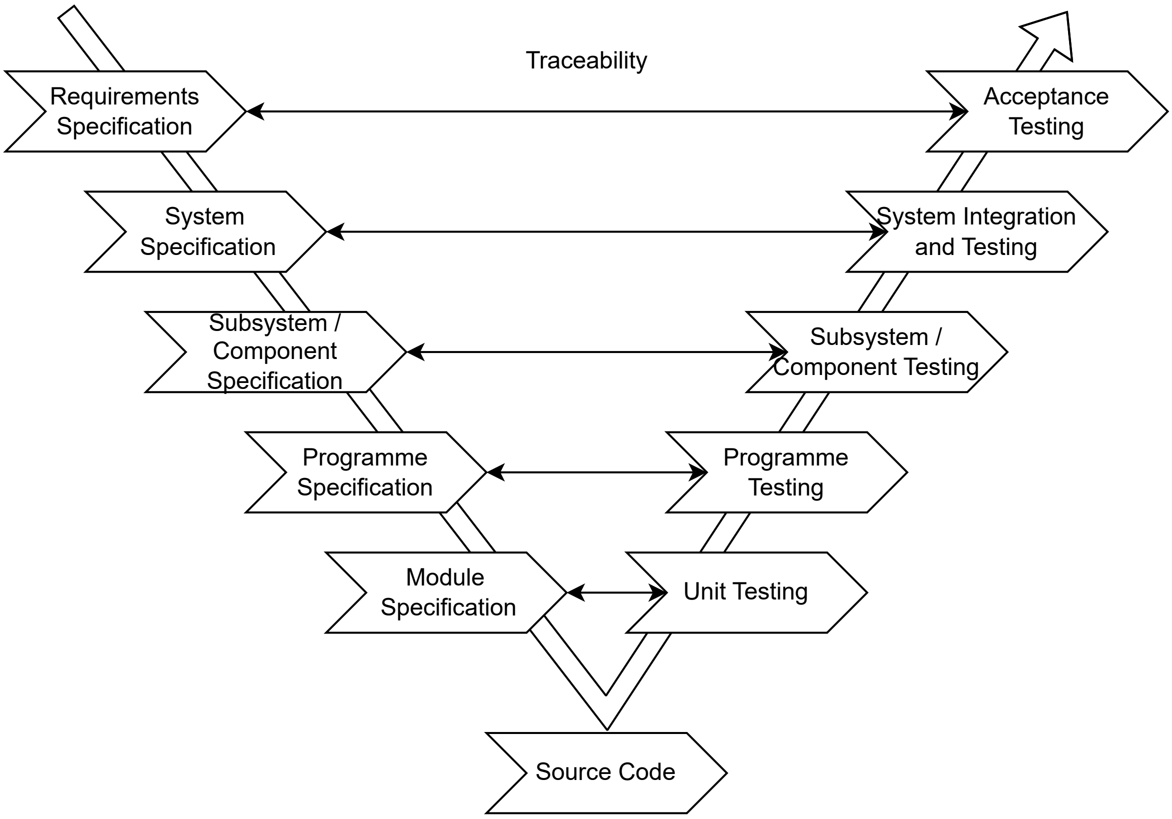
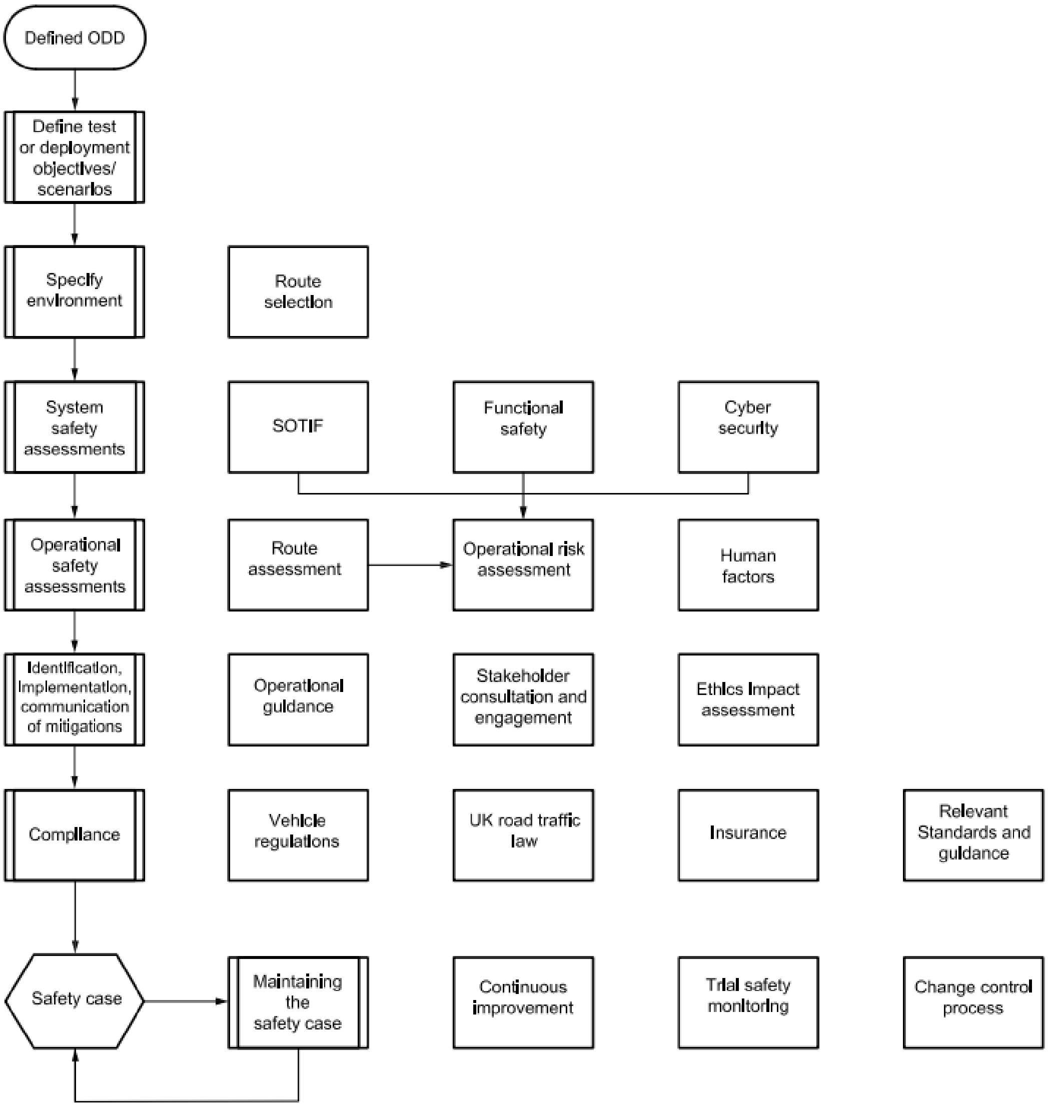


Figure 10: The V-Model in ISO 26262 used for V&V procedures [100].

3 INTRODUCTION TO STANDARDISATION

Standardisation is the implementation and development of technical standards based on the expert consensus from a range of interested parties, such as research bodies, governing bodies, or industrial firms. As seen in the previous sections, the methods used in the development of CAV and V2X technologies are flexible and vary greatly, from fundamental decisions such as choice of operating systems, simulator software or physics engines, to fine-tuned modelling and testing scenarios. Currently, many standards exist for the quality of software and hardware components, however, any research conducted via the use of simulations is unregulated and determined by the preference of project leaders or availability of project resources.

Standards relating to CAV and V2X have been issued by organisations including the British Standardisation Institute (BSI) [103], International Organisation for Standardisation (ISO) [104] [105] [106], and European Telecommunications Standards Institute (ETSI) [107] which provide clear specifications and requirements for software and hardware quality and operations. Protocols defining software and hardware, such as the IEEE 802.11x, 1609.x and 3GPP releases, are tested against standards and safety regulations provided by the above stated organisations. The BSI provides documentation on concepts such as ‘Assuring the operation safety of automated vehicles - Specification’ [108], ‘Data collection and management for automated vehicle trials for the purpose of incident investigation – Specification’ [109], and ‘Operational Design Domain (ODD) taxonomy for an Automated Driving System (ADS) - Specification’ [110]. The ODD specifies the environment for operations in which the CAV or ADS should be capable of performing dynamic tasks safely [110]. Overall, the PAS specification is commonly used in defining the ODD and provides frameworks for safety case development as portrayed in Figure 11 below. The safety cases predominantly depend on the objective of the review and generally consist of consideration of independent safety case review, process review, and self-certification. Previously defined Acts of Parliament and regulations form the framework of safety cases, a tabular representation of commonly applied acts and regulations can be seen in Figure 2 below. The specifications and frameworks provided by BSI, ISO, and ETSI define the requirements for safety, data management, system operation, and many more aspects; they are an essential part of CAV development and implementation into everyday life.



NOTE The left-hand column shows the high-level tasks in safety case development. Subsequent columns on the right are subsets of those tasks; for example, the route assessment and operational risk assessment are part of the operational safety assessment tasks.

Figure 11: Diagram representing a high-level overview of considerations and framework used for the development of safety cases as presented in the BSI PAS 1881:2022 [108].

Safety case area	Relevant legislation and good practice
Trial and testing	• Road Traffic Regulation Act 1984 • Road Vehicles (Construction and Use) Regulations 1986 • Road Traffic Act 1988 • DfT's Safety requirements for automated vehicle trials and testing, including its latest Code of practice • DfT's Highway Code • Law Commission's Automated vehicles: Joint report • Zenzic's Safety case framework • CCAV's Code of practice: Vehicle authorisations and exemptions for more complex CAV trials • BSI PAS 1882, PAS 1883, and PAS 1884
Trial Vehicles	• Road Vehicles (Construction and Use) Regulations 1986 • Road Vehicles (Approval) Regulatons 2020 • BSI PAS 1880 • Driver and Vehicle Standards Agency or Vehicle Certification Agency
Testing location	• UK GDPR • Data Protection Act 2018 • DfT's safety requirements for automated vehicle trials and testing, including its Code of practice
Data, security and connectivity	• DfT's safety requirements for automated vehicle trials and testing, including its Code of practice • DfT's Key Principles of vehicle cyber security • BSI PAS 11281 and PAS 1885

Table 2: A table presenting currently used acts and regulations in safety case frameworks [108].

There is currently a lack of widely accepted standards for the use of simulation software. The choice of simulation software, platform, robotic operation system frameworks, or physics engine is mostly dependent on project resources. Certain software, which may use more out-dated technologies which provide a less accurate representation of the real world, may have been chosen due to the computational capabilities available for the project, as seen in the use of older software such as Gazebo instead of more advanced choices such as Unity or Unreal. This choice may lead to discrepancies in research since more modern software is capable of utilising more advanced technologies such as ray tracing [111] [112] for modelling and graphical representations. Additionally, the choice of modelling methods also increases uncertainty in simulation. The use of physical or data driven models may impact the quality of the simulation, but is once again dependent on the availability of data or on the computational power at hand. In general various open-source or licensed software may be used for research purposes, most with varying modelling capabilities. More commercial organisations may also decide to keep simulation frameworks under commercially sensitive restrictions which further increase the lack of unification of simulations. The use of different interfaces, system architectures, or methods implemented in the design of VTEs create a lack of harmonisation across the field which ultimately restricts innovation and potentially leads to challenges for developers as they may have to abide to different standards set in their regions.

4 CONCLUSION

In conclusion this report has provided an introductory review of the state of the art for simulation software, hardware, and methods used in the development of vehicular communications. The report focused primarily on the interfaces used in V2X development and between simulation software, operation system frameworks, and physics engines. The report identified the different simulation stages, MiL, SiL, HiL, and Hybrid, along with their appropriate use at each stage of V2X development. The architecture, frameworks, and prior models used for development of DSRC and NR 5G V2X technologies were explained through discussion of system structure and protocols dictating the most fundamental concepts such as signal generation, data allocation, and transmission. The report focused on the interfaces used in both communication methods, in particular the IEEE 1609.x protocol family, PC5 and Uu interfaces, and the methods used to implement those protocols within simulations. Finally the report introduced the concept of validation and verification for software and hardware in the CAV sector and outlined the current state of standardisation in CAV development projects, highlighting the necessity for the standardisation for CAV and V2X technology development via simulations.

References

[1] IEEE 802.11 working group. "Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems-LAN/MAN Specific Requirements-Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band". In: *IEEE Project Authorization Request, approved on September 16 1997* (1997).

[2] The Institute of Electrical and Electronics Engineers Standards Association. *IEEE Standard for Wireless Access in Vehicular Environments–Security Services for Application and Management Messages*. https://standards.ieee.org/ieee/1609.2/10258/ [Accessed: (02/11/23)]. 2023.

[3] SAE. *Dedicated Short Range Communications (DSRC) Message Set Dictionary*. https://www.sae.org/standards/content/j2735_201603/ [Accessed: (06/11/23)]. 2016.

[4] 3GPP. *3GPP Specification Technologies Releases*. https://www.3gpp.org/specifications-technologies/releases [Accessed: (03/01/24)].

[5] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Next Generation V2X". In: *IEEE Std 802.11bd-2022 (Amendment to IEEE Std 802.11-2020 as amended by IEEE Std 802.11ax-2021, IEEE Std 802.11ay-2021, IEEE Std 802.11ba-2021, IEEE Std 802.11-2020/Cor 1-2022, and IEEE Std 802.11az-2022)* (2023), pp. 1–144.

[6] Mario H Castañeda Garcia et al. "A tutorial on 5G NR V2X communications". In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), pp. 1972–2026.

[7] Guodong Rong et al. “LGSVL simulator: A high fidelity simulator for autonomous driving”. In: *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*. IEEE. 2020, pp. 1–6.

[8] Alexey Dosovitskiy et al. “CARLA: An Open Urban Driving Simulator”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. Ed. by Sergey Levine, Vincent Vanhoucke, and Ken Goldberg. Vol. 78. Proceedings of Machine Learning Research. PMLR, Nov. 2017, pp. 1–16. url: https://proceedings.mlr.press/v78/dosovitskiy17a.html.

[9] Manel Montull Rodríguez. “Modeling and Characterization of ADAS systems into rFpro simulation environment”. B.S. thesis. Universitat Politècnica de Catalunya, 2023.

[10] NVIDIA. *NVIDIA DRIVE Sim Powered by Omniverse*. https://www.nvidia.com/en-us/self-driving-cars/simulation/ [Accessed: (24/10/23)].

[11] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5.

[12] Javier Manjón Prado. “Using Unreal Engine as an engineering tool for traffic simulation and analysis”. MA thesis. Universitat Politècnica de Catalunya, 2020.

[13] Nathan Koenig and Andrew Howard. “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In: *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*. Vol. 3. IEEE. 2004, pp. 2149–2154.

[14] Alejandro Diaz-Diaz et al. “HD maps: Exploiting opendrive potential for path planning and map monitoring”. In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1211–1217.

[15] Ahmed AbdelHamed, Girma Tewolde, and Jaerock Kwon. “Simulation framework for development and testing of autonomous vehicles”. In: *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE. 2020, pp. 1–6.

[16] Wei Qian et al. “Manipulation task simulation using ROS and Gazebo”. In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE. 2014, pp. 2594–2598.

[17] Unity Technologies. *Unity*. https://unity.com/ [Accessed: (24/10/23)].

[18] Zi Peng et al. “A first look at the integration of machine learning models in complex autonomous driving systems: a case study on Apollo”. In: *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2020, pp. 1240–1250.

[19] Shinpei Kato et al. “Autoware on board: Enabling autonomous vehicles with embedded systems”. In: *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE. 2018, pp. 287–296.

[20] Russell Smith et al. “Open dynamics engine”. In: (2005).

[21] Mark J Kilgard. *The OpenGL utility toolkit (GLUT) programming interface API version 3*. 1996.

[22] Emir Alagić, Jasmin Velagić, and Adnan Osmanović. “Design of Mobile Robot Motion Framework based on Modified Vector Field Histogram”. In: *2019 International Symposium ELMAR*. IEEE. 2019, pp. 135–138.

[23] Lei Zhang, Rene Zapata, and Pascal Lepinay. “Self-adaptive Monte Carlo localization for mobile robots using range finders”. In: *Robotica* 30.2 (2012), pp. 229–244.

[24] Rahul Kumar Bhadani, Jonathan Sprinkle, and Matthew Bunting. “The CAT vehicle testbed: A simulator with Hardware in the Loop for autonomous vehicle applications”. In: *arXiv preprint arXiv:1804.04347* (2018).

[25] Martin Schlager, Wilfried Elmenreich, and Ingomar Wenzel. “Interface design for hardware-in-the-loop simulation”. In: *2006 IEEE International Symposium on Industrial Electronics*. Vol. 2. IEEE. 2006, pp. 1554–1559.

[26] Zhen Li, Michael Kyte, and Brian Johnson. “Hardware-in-the-loop real-time simulation interface software design”. In: *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)*. IEEE. 2004, pp. 1012–1017.

[27] Jean Bélanger, Philippe Venne, Jean-Nicolas Paquin, et al. “The what, where and why of real-time simulation”. In: *Planet Rt* 1.1 (2010), pp. 25–29.

[28] Larry E Owen et al. “Traffic flow simulation using CORSIM”. In: *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*. Vol. 2. IEEE. 2000, pp. 1143–1147.

[29] Albert Albers and Tobias Düser. “Implementation of a Vehicle-in-the-Loop development and validation platform”. In: *FISITA World automotive congress*. Vol. 2010. 2010.

[30] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.

[31] Patric Grauberger et al. “The contact and channel approach–20 years of application experience in product engineering”. In: *Journal of Engineering Design* 31.5 (2020), pp. 241–265.

[32] Steve Rowe and Christopher R Wagner. “An introduction to the joint architecture for unmanned systems (JAUS)”. In: *Ann Arbor* 1001 (2008), p. 48108.

[33] Patrick Morley et al. “Generating a ROS/JAUS bridge for an autonomous ground vehicle”. In: *Proceedings of the 2013 ACM workshop on Domain-specific modeling*. 2013, pp. 13–18.

[34] Joshua Fadaie. “The state of modeling, simulation, and data utilization within industry: An autonomous vehicles perspective”. In: *arXiv preprint arXiv:1910.06075* (2019).

[35] Nariman Fouladinejad et al. “Modeling virtual driving environment for a driving simulator”. In: *2011 IEEE International Conference on Control System, Computing and Engineering*. IEEE. 2011, pp. 27–32.

[36] Shital Shah et al. “Airsim: High-fidelity visual and physical simulation for autonomous vehicles”. In: *Field and Service Robotics: Results of the 11th International Conference*. Springer. 2018, pp. 621–635.

[37] Fanta Camara et al. “Pedestrian models for autonomous driving part II: high-level models of human behavior”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (2020), pp. 5453–5472.

[38] Canan Yemenicioglu, KAYA Sinasi, and Dursun Zafer Seker. "Accuracy of 3D (three-dimensional) terrain models in simulations". In: *International Journal of Engineering and Geosciences* 1.1 (2016), pp. 34–38.

[39] Benedikt Schwab and Thomas H Kolbe. "Validation of Parametric OpenDRIVE Road Space Models". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 10 (2022), pp. 257–264.

[40] Byron Tapley et al. "The GGM03 mean earth gravity model from GRACE". In: *AGU Fall Meeting Abstracts*. Vol. 2007. 2007, G42A–03.

[41] Arnaud Chulliat et al. "The US/UK world magnetic model for 2015-2020". In: (2015).

[42] Ronald B Stull. *Practical meteorology: an algebra-based survey of atmospheric science*. University of British Columbia, 2015.

[43] Ralph H Rasshofer, Martin Spies, and Hans Spies. "Influences of weather phenomena on automotive laser radar systems". In: *Advances in radio science* 9 (2011), pp. 49–60.

[44] Ishan Banerjee et al. "Graphical user interface (GUI) testing: Systematic mapping and repository". In: *Information and Software Technology* 55.10 (2013), pp. 1679–1694.

[45] ZVI GREENSTEIN NVIDIA. *A Path for Safe Self-Driving: NVIDIA Opens DRIVE Constellation Platform to Simulation Partners*. https://blogs.nvidia.com/blog/2018/09/12/drive-constellation-open-simulation/ [Accessed: (31/10/23)]. 2018.

[46] Unity Technologies. *Unity User Manual 2022.3 (LTS)*. https://docs.unity3d.com/Manual/index.html [Accessed: (01/11/23)]. 2023.

[47] Jian Wang et al. "A survey of vehicle to everything (V2X) testing". In: *Sensors* 19.2 (2019), p. 334.

[48] Hubert Zimmermann. "OSI reference model-the ISO model of architecture for open systems interconnection". In: *IEEE Transactions on communications* 28.4 (1980), pp. 425–432.

[49] Daniel Jiang and Luca Delgrossi. "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments". In: *VTC Spring 2008-IEEE vehicular technology conference*. IEEE. 2008, pp. 2036–2040.

[50] Roberto A Uzcátegui, Antonio Jose De Sucre, and Guillermo Acosta-Marum. "Wave: A tutorial". In: *IEEE Communications magazine* 47.5 (2009), pp. 126–133.

[51] Yunxin Li. "An overview of the DSRC/WAVE technology". In: *Quality, Reliability, Security and Robustness in Heterogeneous Networks: 7th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine 2010, and Dedicated Short Range Communications Workshop, DSRC 2010, Houston, TX, USA, November 17-19, 2010, Revised Selected Papers 7*. Springer. 2012, pp. 544–558.

[52] John B Kenney. "Dedicated short-range communications (DSRC) standards in the United States". In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182.

[53] Lin Cheng et al. "A measurement study of time-scaled 802.11 a waveforms over the mobile-to-mobile vehicular channel at 5.9 GHz". In: *IEEE Communications Magazine* 46.5 (2008), pp. 84–91.

[54] Gordon L Stüber. *Principles of mobile communication.* 1996.

[55] David W Matolak. “V2V communication channels: State of knowledge, new results, and what’s next”. In: *Communication Technologies for Vehicles: 5th International Workshop, Nets4Cars/Nets4Trains 2013, Villeneuve d’Ascq, France, May 14-15, 2013. Proceedings 5*. Springer. 2013, pp. 1–21.

[56] Johan Karedal et al. “Path loss modeling for vehicle-to-vehicle communications”. In: *IEEE transactions on vehicular technology* 60.1 (2010), pp. 323–328.

[57] John David Parsons and Prof J David Parsons. *The mobile radio propagation channel*. Vol. 2. Wiley New York, 2000.

[58] DW Matolak and Q Wu. “Channel models for V2V communications: A comparison of different approaches”. In: *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*. IEEE. 2011, pp. 2891–2895.

[59] IEEE. *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control*. https://www.iso.org/obp/ui#iso:std:iso-iec:8802:-2:ed-3:v1:en [Accessed: (03/11/23)]. 1998.

[60] Chi-Hyun Park and Dong-Ho Cho. “An adaptive logical link control for wireless Internet service in ITS”. In: *Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No. 99CH36324)*. Vol. 4. IEEE. 1999, pp. 2213–2217.

[61] Stijn De Vuyst et al. “Analysis of Stop-and-Wait ARQ for a wireless channel”. In: *4or* 7 (2009), pp. 61–78.

[62] “IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture”. In: *IEEE Std 1609.0-2019 (Revision of IEEE Std 1609.0-2013)* (2019), pp. 1–106.

[63] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.

[64] Erik Dahlman, Stefan Parkvall, and Johan Skold. *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[65] ETSI. *LTE; Service requirements for V2X services (3GPP TS 22.185 version 15.0.0 Release 15)*. https://www.etsi.org/deliver/etsi_ts/122100_122199/122185/15.00.00_60/ts_122185v150000p.pdf [Accessed: (02/11/23)]. 2018.

[66] Lili Miao, John Jethro Virtusio, and Kai-Lung Hua. “PC5-based cellular-V2X evolution and deployment”. In: *Sensors* 21.3 (2021), p. 843.

[67] Karthikeyan Ganesan et al. “5G V2X architecture and radio aspects”. In: *2019 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE. 2019, pp. 1–6.

[68] Sinuk Choi, Dongyoon Kwon, and Ji-Woong Choi. “Latency Analysis for Real-Time Sensor Sharing Using 4G/5G C-V2X Uu Interfaces”. In: *IEEE Access* 11 (2023), pp. 35197–35206.

[69] 3GPP. *Study on self-evaluation towards IMT-2020 submission*. https://www.3gpp.org/ftp//Specs/archive/37_series/37.910/ [Accessed: (16/11/23)]. 2020.

[70] Mehdi Harounabadi et al. “V2X in 3GPP standardization: NR sidelink in release-16 and beyond”. In: *IEEE Communications Standards Magazine* 5.1 (2021), pp. 12–21.

[71] Alessandro Bazzi et al. "On the design of sidelink for cellular V2X: A literature review and outlook for future". In: *IEEE Access* 9 (2021), pp. 97953–97980.

[72] 3GPP. "User Equipment (UE) radio transmission and reception part 1: Range 1 Standalone (Release 16)". In: *3GPP TS 38101-1-G30, Technical Specification* (2020).

[73] 3GPP. "User Equipment (UE) Radio Transmission and Reception; Part 2: Range 2 Standalone". In: *3GPP TS 138 101-2 V16.4.0, Technical Specification* (2020).

[74] Ashfaq Ahmed et al. "Hybrid automatic repeat request (HARQ) in wireless communications systems and standards: A contemporary survey". In: *IEEE Communications Surveys & Tutorials* 23.4 (2021), pp. 2711–2752.

[75] 3GPP. *3GPP Release 16.2*. https://www.etsi.org/deliver/etsi_ts/124300_124399/124385/16.02.00_60/ts_124385v160200p.pdf [Accessed: (21/11/23)]. 2020.

[76] 3GPP. *3GPP Release 15.7*. https://www.etsi.org/deliver/etsi_ts/136200_136299/136213/15.07.00_60/ts_136213v150700p.pdf [Accessed: (21/11/23)]. 2019.

[77] 3GPP. *3GPP Release 17 Specification*. https://www.3gpp.org/specifications-technologies/releases/release-17 [Accessed: (14/11/23)]. 2022.

[78] Jonathan Nibert, Marc E Herniter, and Zachariah Chambers. "Model-based system design for MIL, SIL, and HIL". In: *World Electric Vehicle Journal* 5.4 (2012), pp. 1121–1130.

[79] Michael Klöppel-Gersdorf and Thomas Otto. "A hybrid real and virtual testing framework for V2X applications". In: *Smart Cities, Green Technologies, and Intelligent Transport Systems: 9th International Conference, SMARTGREENS 2020, and 6th International Conference, VEHITS 2020, Prague, Czech Republic, May 2-4, 2020, Revised Selected Papers 9*. Springer. 2021, pp. 190–203.

[80] Brian McCarthy et al. "OpenCV2X: Modelling of the V2X cellular sidelink and performance evaluation for aperiodic traffic". In: *arXiv preprint arXiv:2103.13212* (2021).

[81] Ethan Zhang and Neda Masoud. "V2xsim: A V2X simulator for connected and automated vehicle environment simulation". In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–6.

[82] Waqar Anwar, Norman Franchi, and Gerhard Fettweis. "Physical layer evaluation of V2X communications technologies: 5G NR-V2X, LTE-V2X, IEEE 802.11 bd, and IEEE 802.11 p". In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE. 2019, pp. 1–7.

[83] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules". In: *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000)* (2010), pp. 1–38.

[84] Lichao Xu et al. "Distributed simulation platforms and data passing tools for natural hazards engineering: Reviews, limitations, and recommendations". In: *International Journal of Disaster Risk Science* 12 (2021), pp. 617–634.

[85] Kyle Kuffermann. *An implementation of the IEEE1609. 4 wave standard for use in a vehicular networking testbed*. Florida Atlantic University, 2014.

[86] Qi Chen, Daniel Jiang, and Luca Delgrossi. “IEEE 1609.4 DSRC multi-channel operations and its implications on vehicle safety communications”. In: *2009 IEEE vehicular networking conference (VNC)*. IEEE. 2009, pp. 1–8.

[87] Tsu-Kuang Lee et al. “Building a V2X simulation framework for future autonomous driving”. In: *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE. 2019, pp. 1–6.

[88] Hai Heng Ng et al. “BESAFE: Design and implementation of a DSRC-based test-bed for connected autonomous vehicles”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2018, pp. 3742–3748.

[89] Donglin Wang et al. “Methodologies of link-level simulator and system-level simulator for C-V2X communication”. In: *2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE)*. IEEE. 2019, pp. 178–184.

[90] Suryanarayanaraju Pusapati et al. “Simulation of NR-V2X in a 5G Environment using OMNeT++”. In: *2022 IEEE Future Networks World Forum (FNWF)*. IEEE. 2022, pp. 634–638.

[91] Krasen Angelov, Stanimir Sadinov, and Panagiotis Kogias. “Modelling and Study of the Downlink Physical Layer in 5G NR Mobile Network”. In: *2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*. 2022, pp. 1–4.

[92] Sandra Lagen et al. “New radio physical layer abstraction for system-level simulations of 5G networks”. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE. 2020, pp. 1–7.

[93] Manuel Gonzalez-Martín et al. “Analytical models of the performance of C-V2X mode 4 vehicular communications”. In: *IEEE Transactions on Vehicular Technology* 68.2 (2018), pp. 1155–1166.

[94] Judith S Dahmann. “High level architecture for simulation”. In: *Proceedings First International Workshop on Distributed Interactive Simulation and Real Time Applications*. IEEE. 1997, pp. 9–14.

[95] Tobias Queck et al. “Realistic simulation of V2X communication scenarios”. In: *2008 IEEE Asia-Pacific Services Computing Conference*. IEEE. 2008, pp. 1623–1627.

[96] David Rieck et al. “Efficient traffic simulator coupling in a distributed V2X simulation environment”. In: *Proceedings of the 3rd international ICST conference on simulation tools and techniques*. 2010, pp. 1–9.

[97] Apratim Choudhury et al. “An integrated simulation environment for testing V2X protocols and applications”. In: *Procedia Computer Science* 80 (2016), pp. 2042–2052.

[98] The American Society of Mechanical Engineers. *Verification, Validation and Uncertainty Quantification (VVUQ)*. https://www.asme.org/codes-standards/publications-information/verification-validation-uncertainty [Accessed: (25/01/24)].

[99] ASAM Test Specification Study Group. “EVOLVING LANDSCAPES OF COLLABORATIVE TESTING FOR ADAS & AD”. In: *Association for Standardization of Automation and Measuring Systems* (2022).

[100] Nijat Rajabli et al. “Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review”. In: *IEEE Access* 9 (2021), pp. 4797–4819.

[101] Michal Jasinski, Pawel Skruch, and Mateusz Komorkiewicz. “Validation Framework for Generic Radar Sensor Models”. In: *IEEE Access* 10 (2022), pp. 18257–18267.

[102] Yining Ma et al. “Verification and Validation Methods for Decision-Making and Planning of Automated Vehicles: A Review”. In: *IEEE Transactions on Intelligent Vehicles* 7.3 (2022), pp. 480–498.

[103] The British Standards Institute. *CAV Resources – Standards, guidelines, research and viewpoints*. https://www.bsigroup.com/en-GB/CAV/cav-resources/#cavstandards [Accessed: (02/11/23)]. 2022.

[104] International Organization for Standardization. *ISO 17419:2018 Intelligent transport systems Cooperative systems Globally unique identification*. https://www.iso.org/standard/70077.html [Accessed: (08/01/24)].

[105] International Organization for Standardization. *ISO 17515-3:2019 Intelligent transport systems Evolved-universal terrestrial radio access network Part 3: LTE-V2X*. https://www.iso.org/standard/73238.html [Accessed: (08/01/24)].

[106] International Organization for Standardization. *ISO 15628:2013 Intelligent transport systems Dedicated short range communication (DSRC) DSRC application layer*. https://www.iso.org/standard/59288.html [Accessed: (08/01/24)].

[107] European Telecommunications Standards Institute. *Automotive Intelligent Transport System (ITS) Standards*. https://www.etsi.org/technologies/automotive-intelligent-transport [Accessed: (08/01/24)].

[108] The British Standards Institution. *PAS 1881:2022 Assuring the operational safety of automated vehicles. Specification*. https://knowledge.bsigroup.com/products/assuring-the-operational-safety-of-automated-vehicles-specification?version=standard [Accessed: (08/01/24)]. 2022.

[109] The British Standards Institution. *PAS 1882:2021 Data collection and management for automated vehicle trials for the purpose of incident investigation. Specification*. https://knowledge.bsigroup.com/products/data-collection-and-management-for-automated-vehicle-trials-for-the-purpose-of-incident-investigation-specification?version=standard [Accessed: (08/01/24)]. 2021.

[110] The British Standards Institution. *PAS 1883:2020 Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specifi cation*. https://knowledge.bsigroup.com/products/operational-design-domain-odd-taxonomy-for-an-automated-driving-system-ads-specification?version=standard [Accessed: (11/01/24)]. 2021.

[111] Danping He et al. “The Design and Applications of High-Performance Ray-Tracing Simulation Platform for 5G and Beyond Wireless Communications: A Tutorial”. In: *IEEE Communications Surveys & Tutorials* 21.1 (2019), pp. 10–27.

[112] NVIDIA. *Real-Time Ray Tracing*. https://developer.nvidia.com/rtx/ray-tracing [Accessed: (08/01/24)].