View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by CURVE/open

DigiCAV project: Exploring a test-driven approach in the development of connected and autonomous vehicles

Kyriakopoulos, I, Jaworski, P & Kanarachos, S

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:

Kyriakopoulos, I, Jaworski, P & Kanarachos, S 2020, DigiCAV project: Exploring a test-driven approach in the development of connected and autonomous vehicles. in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE).*, 8965026, Institute of Electrical and Electronics Engineers Inc., 8th IEEE International Conference on Connected Vehicles and Expo, ICCVE 2019, Graz, Austria, 4/11/19.
https://dx.doi.org/10.1109/ICCVE45908.2019.8965026

DOI 10.1109/ICCVE45908.2019.8965026

Publisher: IEEE

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

DigiCAV project: Exploring a Test-Driven Approach in the Development of Connected and Autonomous Vehicles

Ioannis Kyriakopoulos
Horizon Scanning
HORIBA MIRA Limited
Nuneaton, United Kingdom
ioannis.kyriakopoulos@horiba-mira.com

Pawel Jaworski
Horizon Scanning
HORIBA MIRA Limited
Nuneaton, United Kingdom
pawel.jaworski@horiba-mira.com

Stratis Kanarachos
School of Mechanical, Aerospace and Automotive Engineering
Coventry University
Coventry, United Kingdom
ab8522@coventry.ac.uk

***Abstract*—Testing & validation of high-level autonomy features requires large amounts of test data, which conventionally is achieved by accumulating miles on the road and dedicated proving grounds. This places an extreme burden not only on original equipment manufacturers (OEMs) of connected and autonomous vehicles (CAVs), but also on Tier 1 suppliers of CAV components, both in terms of cost and delivery time. To this end, multiple simulation platforms and techniques have emerged, such as hardware-in-the-loop testing methods, while the concept of co-simulation is gaining popularity as a more comprehensive solution for testing and validating CAVs. The aim of the DigiCAV project is to explore the feasibility of a co-simulation platform adopting a test-driven development approach for CAVs, by enabling a seamless testing and validation process across all stages of development, supporting a wide range of testing from model-in-the-loop of a CAV component all the way to vehicle-in-the-loop of a fully assembled vehicle on HORIBA MIRA's dedicated CAV proving ground test facilities. Furthermore, emphasis will be put on quality aspects such as testing accuracy, usability and protection of intellectual property rights. This paper introduces the DigiCAV project and disseminates results from its first deliverable focusing on capturing user requirements for the proposed simulation platform.**

***Keywords*—autonomous vehicles, connected vehicles, simulation, testing, system validation**

I. INTRODUCTION

The complexity of testing new vehicles and components has become a serious burden to automotive original equipment manufacturers (OEMs) and Tier 1 suppliers, as they adopt shorter product development life cycles in an effort to reduce the time to market of new vehicles and components while saving on development costs. This has resulted in adhering to a V-model development process where testing activities are incorporated in each stage of development [1]. Among the several benefits associated with such a process are time and cost savings, usually linked to a reduction in the number of vehicle prototypes needed, and the addressing of unplanned design changes earlier in the development process [2].

In development based on the V-model, testing typically starts using a model (model-in-the-loop, MiL) and it is then carried out using various types of software-in-the-loop (SiL) and hardware-in-the-loop (HiL) test bench setups, which in many cases are different based on the unit under test (UUT). This results in different testbeds for testing the vehicle's individual components all the way to the fully-assembled vehicle. Finally, fleet testing of prototypes takes place on a test track or road for the final adjustments (vehicle-in-the-loop, ViL).

When considering the development of CAV components, especially highly automated ones (SAE Level 4 and 5), there are additional complexities introduced, stemming from the exponential complexity of CAVs compared to traditional vehicles [3], leading to significantly increased testing times [4]. Furthermore, due to their different nature and operational environment, different testing and validation (T&V) processes need to be explored with a number of identified major challenge areas presented in [3]. In more detail, the importance of proving ground testing [5], testing with the driver out of the loop [3] as well as the consideration of human factors-related variables [6] have been identified as essential requirements for CAV T&V. Additionally, the increased complexity of CAVs stresses the importance of developing T&V processes where qualities such as usability, traceability and time-to-market are acknowledged.

In an effort to address the aforementioned challenges, the Digital CAV Proving Ground Feasibility Study (hereon referred to as the DigiCAV project) builds on HORIBA MIRA's vehicle testing solutions to investigate the feasibility of a novel simulation approach. The aim is to aid the development process of CAVs by enabling iterative T&V of their components at different stages of development against a complete configuration of the vehicle and the entire CAV ecosystem (e.g. traffic and communication infrastructure), allowing T&V of CAVs much earlier in the development cycle. This will be promoted by a component-in-the-loop (CiL) simulation approach where a component could range from individual vehicle components to a fully assembled vehicle. Such an approach will build upon a proposed simulation platform enabling the connection of both real and simulated CAV components (such as autonomous driving modules, simulation software, sensor data and algorithms) to entire vehicles by providing a set of appropriate interfaces, something which will allow conducting the respective X-in-the-loop simulation tests across the entire V-cycle. Furthermore, the integration of accelerated testing processes will be also explored as well as the support of component resilience tests through the application of fault injection techniques on the interface level. The main components and data flow in the CiL simulation platform proposed here can be seen in Fig. 1.

One of the key advantages of the proposed platform is its high level of modularity achieved by decoupling the functional and interface components of subsystems from each other, while also supporting the use of open interfaces. In this

---

This project is funded by Innovate UK and HORIBA MIRA Ltd.

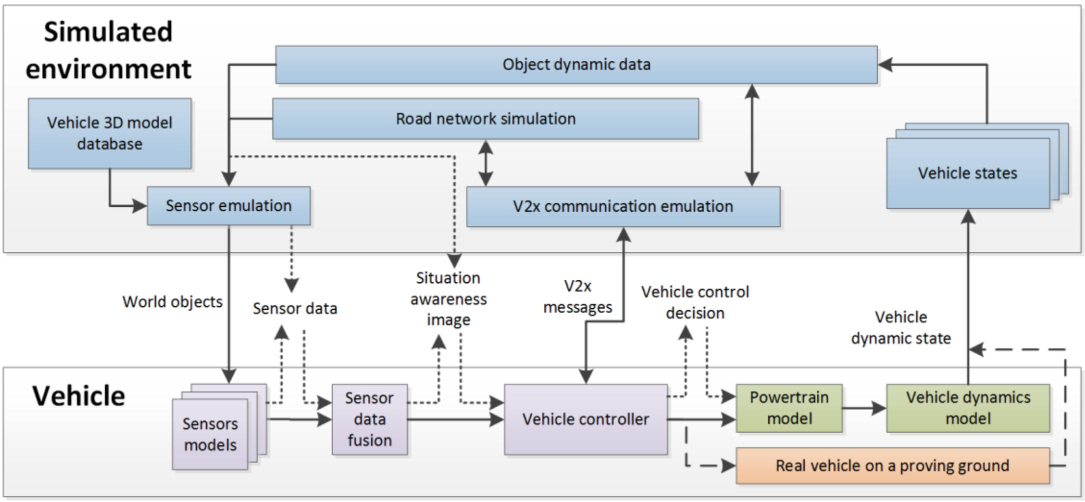

Fig. 1. Overview of the proposed component-in-the-loop simulation platform architecture.

way, the functional components will remain the same at each of the MiL/SiL/HiL/ViL stages, with only the interface components changing to suit the needs at each specific development stage. As a result of this, only one functional model will need to be maintained, and whether errors are functional or interface-based will be more explicit.

Such a modular simulation approach will also allow the support of test-driven development (TDD) of CAVs, an approach traditionally used in software development. This will provide support of relevant test cases across all stages of development, testing and validation, enabling a seamless testing process, which can lead to time and cost benefits.

Finally, the proposed platform will also promote conducting CAV tests without the need to reveal any IP information associated with the components under test, such as source code or other implementation details, to HORIBA MIRA or to other 3rd parties, while also taking into consideration the accuracy of the tests performed so that confidence in the use of the tested components can be inferred.

II. RELATED WORK

MiL and SiL testing of CAV components is usually performed fully in simulation and in complete isolation from the final vehicle.

A common approach to MiL/SiL testing is to develop a model of the UUT (e.g. a vehicle controller) in a dedicated modelling environment and then connect this (or the resulted generated software) to a vehicle, a traffic simulator, or both. Popular vehicle simulators such as IPG CarMaker [7] and PreScan [8] as well as traffic simulators, such as PTV Vissim [3] and TRANSIMS [4] allow such integration. For example, the Vissim traffic simulator [9] provides an application programming interface (API) called Vissim external driver model (VEDM), which provides full control of a vehicle's movement, hence aiding in the development and testing of CAV algorithms.

In [10], VEDM was used to code the driver model and vehicle clustering strategy for vehicles equipped with cooperative adaptive cruise control (CACC) while in [11], a decision-making CAV control algorithm was developed in Vissim using VEDM. The developed algorithm allowed a CAV to have “longitudinal control, search adjacent vehicles, identify nearby CAVs and make lateral decisions based on a ruleset associated with motorway traffic operations”. Other cases where MiL/SiL methods have been used are in [12] and [13]. Furthermore, some examples of MiL/SiL simulation platforms can be seen in [14] and [15].

A common limitation of MiL/SiL testing is the accuracy of simulation due to the increased level of abstraction of the environment models (vehicle, traffic and infrastructure) used. The DigiCAV platform attempts to increase the usefulness of these tests by conducting them in a simulated environment resembling a complete vehicle and its environment, allowing for early validation.

The following stage of HiL testing is typically the first stage of the validation phase [4], which is where most of the work around CAV component testing has been identified. A standard configuration of a HiL test bed is to externally connect the UUT (e.g. an engine or a vehicle control unit) implemented in hardware to a traffic and/or network simulator through an appropriate set of interfaces. Some examples of traffic simulators supporting this are the Vissim and TRANSIMS mentioned earlier.

In [16], a HiL simulator for developing and testing automated driving algorithms, such as CACC and coordinated lane keeping controllers, has been proposed. This consists of three main elements: a real time traffic simulator (dSPACE Scalexio), an electronic control unit (dSPACE Microautobox) and two dedicated short-range communication (DSRC) modems.

Another HiL platform for testing CAV applications is presented in [17]. The platform consists of a mixture of physical and simulated components including a physical CAV controller, a traffic signal controller, communication devices, and a traffic simulator (Vissim). The aforementioned platform has a vehicle-to-infrastructure (V2I) focus, but it is suggested that it could potentially be modified to evaluate CAV applications based on vehicle-to-vehicle (V2V) technologies as well [17]. The platform’s capabilities for testing hardware vehicle components seems to be limited to the DSRC devices (e.g., On Board Units-OBUs or Road Side Units-RSUs), while no support for testing software vehicle components exists. This is in contrast to the DigiCAV project where the connection of real and simulated vehicle components to the simulation platform will be made possible through the use of open interfaces.

In other work, a HiL testbed, based on the Robot Operating System, supporting the development and testing of autonomous vehicle algorithms and applications is proposed [18]. This is a CiL simulation platform integrating simulated and real components, allowing an easy transition from software to hardware-in-the-loop simulation.

A number of proprietary HiL simulation platforms used in CAV development were also identified. Their common characteristic is their turnkey nature, requiring little configuration or customization before use. One example is National Instruments' HiL platform for testing Advanced Driver Assistance Systems (ADAS) and autonomous driving applications [19]. This consists of NI's PXI real-time targets[1], each of them simulating vehicle components affected by typical ADAS systems. The tight synchronization achieved among the PXI targets enables simulations of various driving scenarios and sensors [20].

Another proprietary simulation platform is IPG's ESC-HiL test system used to test and validate real-world electronic stability control (ESC) units, enabling all ESC functionalities to be tested with a single HiL system [21]. The main uses of IPG's platform are: (i) the verification and validation of ESC systems and (ii) simulation-based ESC homologation (certification).

HiL testing is usually followed by ViL testing aiming at overcoming the common HiL limitations, such as the low accuracy of the vehicle models used. There are generally three ways identified of implementing ViL testing: (i) offline in simulation using a vehicle and a traffic simulator, (ii) in a laboratory environment using a chassis dynamometer test bench and (iii) on a dedicated proving ground (PG) test facility. ViL testing methods (i) and (ii) help bridge the gap between conventional HiL and real road testing. Within the objectives of the DigiCAV platform is the support of all three of the aforementioned ViL testing methods through the design of appropriate set of interfaces, whereas particular emphasis will be put on the support of ViL on PG test facilities.

An example of a simulation only ViL test bed is presented in [22], where IPG CarMaker, a full-vehicle simulation engine also including driver and environment models, is connected to the Vissim traffic simulator to create a test bed for the evaluation of ADAS systems.

An example of a chassis dynamometer test bench for CAVs is AVL's DrivingCube ViL platform [23], [24]. In this platform, the mechanical inputs (torque, steering force) are simulated by a vehicle and powertrain simulator, while a range of ADAS-related sensors (camera, radar, lidar etc.) are stimulated through wireless communication (over the air).

Finally, ViL on a dedicated proving ground test facility can be considered as a final means of validation. This enables one to evaluate the behavior of the component under test with real actors in a real vehicle, which is driven on a test track, enabling the revealing of potential negative consequences, like false positives.

PG trials of CAVs are complementing public road trials addressing the more dangerous scenarios in a controlled environment. However, the development of dedicated CAV PGs is still an emerging area, presumably because of the high cost associated with their development. The interested reader is referred to [25] for specification details of a test track for driverless cars. HORIBA MIRA is one of the few companies offering dedicated CAV PG test facilities (City Circuit and TIC-IT[2] facilities) and one of the distinctive goals of the DigiCAV simulation platform is its integration with these facilities.

A more comprehensive solution for CAV simulation is the use of co-simulation platforms, also referred to as integrated testing platforms [26]. In co-simulation, different simulators are coupled together such that global simulation of a coupled system is achieved [27]. Various approaches to co-simulation exist [28], such as dynamic (where components can enter and leave simulation during run-time), distributed (enabling the de-coupling of the simulator units as well as their parallel execution as separate processes), co-simulation of discrete- and/or continuous-time models, etc.

Generally, co-simulation is performed either adhering to an integrated or to a federated approach. In the first, different simulators are integrated into a single simulation environment, while in the second, simulators are self-contained units located and executed in the same or across different platforms, enabling the so-called distributed co-simulation. Typical advantages of this approach include dynamic management, incremental design and development support as well as parallel simulation [29]. Furthermore, a federated co-simulation approach in theory allows the combination of any simulators, although this is difficult in practice [30].

When a federated approach is considered, two popular co-simulation standards used are the High-Level Architecture standard (HLA) [31] and the Functional Mock-Up Interface (FMI) [32], with the first mostly used in the aerospace industry and the second in the automotive industry. HLA is an architecture enabling distributed and parallel simulation [6], while FMI defines a standardized interface for coupling simulation tools in a co-simulation environment [33], suitable for the simulation of complex cyber-physical systems.

In [34], the ACOSAR (Advanced Co-Simulation Open System Architecture) project is introduced. ACOSAR is based on the FMI, however, although FMI addresses the integration of simulation models, it does not support the integration of real-time systems into simulation environments. To address this gap and ease the integration and coupling of real-time systems (especially of HiL testbenches) to co-simulation platforms, ACOSAR suggested the development of a non-proprietary interface named Advanced Co-simulation Interface (ACI). An overview of the simulation platform proposed by the ACOSAR project can be seen in Fig. 2.

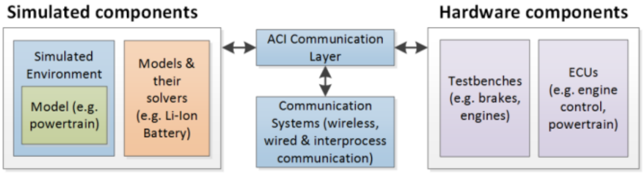

Fig. 2. An overview of the ACOSAR simulation platform architecture.

---

[1] NI PXI is a PC-based platform built around the PXI and PXI Express open standards, used for simulating a wide variety of systems.

[2] The TIC-IT facility will provide an environment for conducting controllability tests for CAVs enabling testing a wide range of CAV and traditional driving scenarios.

The objectives of ACOSAR have many similarities to those of DigiCAV, however, three distinct differences of the DigiCAV platform are: (i) full ViL capability, (ii) focus on CAV applications, especially those involving a proving ground and (iii) support of test-driven development approaches.

In [26], an integrated co-simulation platform for Connected Vehicle (CV) and Vehicle Ad-hoc Network (VANET) related studies is presented. This is based on the integration of a traffic and a driving simulator by the addition of a middleware layer (called intermediary simulation middleware) providing a set of interfaces which allow both driver models as well a human driver in-the-loop along with the traffic simulator. Furthermore, the interfaces provided by the middleware can be used to add additional vehicle and infrastructure elements to the simulation platform, such as ones suitable for testing CAVs. Although the driving simulator of the platform is implemented on hardware, no consideration towards supporting the connection of vehicle components through appropriate interfaces was given.

In [35], a co-simulation approach based on the concept of multi-agent-systems for the analysis of both autonomous and semi-assisted driving vehicles is presented. This approach allows the respective simulation tools to connect and disconnect from the simulation platform during run-time.

The focus of the simulation platform proposed is on simulation of autonomous vehicles in a mixed environment. In particular, the main focus is to perform simulations on networks of vehicular traffic, preferably in urban areas, which could be heavily crowded and traffic jams may occur, while another focus is on the V2V/V2I communications, aiming to simulate their effectiveness as well as to which degree they improve operations in CAV scenarios.

This platform allows the connection of external components called "agents" through the use of appropriate interfaces. The simulator can receive actuator values from the agents as well as various other information such as the characteristics of an agent's sensors, actuators and physical properties. Overall, there are three types of agents supported; (i) game type, (ii) real-vehicle type and (iii) infrastructure agents. Game type agents allow the user to directly control a vehicle within the simulator, a real vehicle type agent allows connection of real vehicles to the simulator and infrastructure agents allow the connection of road network infrastructure components (e.g. a smart traffic light).

Compared to the DigiCAV platform, the granularity of the simulation platform presented in [35] is at a vehicle level, whereas the one of DigiCAV at a vehicle component-level. Furthermore, the objectives of DigiCAV is to enable complete testing and validation of CAV components and not of CAV scenarios.

III. The DigiCAV Project

A. *Project Overview and Structure*

The DigiCAV project is split into eight work packages (WPs). At this stage of the project, WP1 has been completed and the project runs through WP2.

WP1 was mainly concerned with literature review and requirement definition. The key results of the literature review have been presented in Section II of this paper. Requirement definition was conducted by the use of a requirements engineering process, which included the development of a requirements capture methodology and finally a list of requirements for the DigiCAV simulation platform. Key stages of the requirements engineering process followed are in Subsections B, C and D of this section.

WP2 is concerned with the architecture design of the simulation platform and includes exploring potential simulation and interface architectures in order to select two best approaches for further investigation.

WP3 is focused on the design of necessary hardware and software interfaces to enable implementation of a proof of concept.

WP4 is focused on identifying accelerated test methods and test case scenarios which can be incorporated in the simulation platform and will include the systematic design of experiments as well as the development of fault injection methods.

WP5 is concerned with establishing a proof-of-concept purely in simulation and finalizing the architecture and interfaces for subsequent HiL demonstrations.

WP6 aims to provide a proof-of-concept demonstration of the DigiCAV platform by performing a series of HiL simulations. These will demonstrate the capability to connect software and hardware components to the DigiCAV platform.

Potential demonstrations include ones of Level 4 or 5 CAV controllers in the loop, which will be either in the form of an externally hosted web service or a black box ECU. An example of the latter can be seen in Fig. 3 where a vehicle controller (ECU) is connected to the proposed platform architecture enabling in-the-loop testing. In this example, the ECU is based on the Controller Area Network (CAN) communication protocol and is connected to the platform's universal component interface through simulated CAN interfaces provided by the platform, whereas the outputs of the controller (vehicle control decision commands) could be vehicle commands such as acceleration & heading demands.

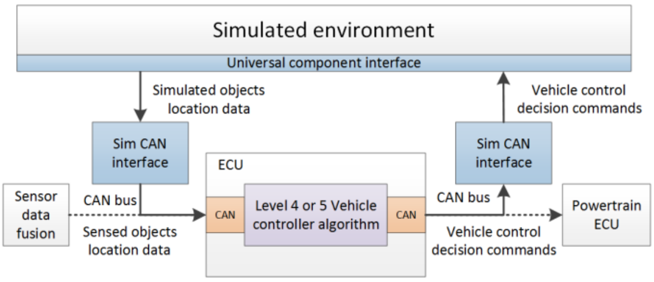

Fig. 3. An example of a vehicle controller in-the-loop in the proposed simulation platform architecture.

WP7 is concerned with evaluating the effectiveness of the proposed accelerated testing methods and identifying aspects requiring further development, as well as with developing a number of fault injection methods.

Finally, WP8 runs in parallel with the other WPs and is focused on the dissemination and exploitation of the project work, including activities such as attending conferences, disseminating project results, engaging with potential end users (OEMs and Tier 1 suppliers) and developing post-project exploitation plans.

B. *Requirements Capture Methodology*

In the DigiCAV project, a requirements engineering process using an agile methodology is followed. Unlike traditional requirements engineering processes (e.g. ones based on a waterfall life cycle model), agile requirements engineering activities are not sequential, are marked by extensive collaboration (e.g. face-to-face communication) and are performed during several short development cycles [36], [37]. Here, the commonly used agile techniques of User Stories and Use Cases are used to capture the requirements of the proposed simulation platform.

User Stories include a short description of features from a customer's point of view, which are used as a basis to create associated Use Cases. Additionally, Use Cases refer to structured descriptions of particular system functions the DigiCAV simulation platform must be able to perform in order to satisfy the corresponding User Stories. In more detail, the goals of Use Cases are: (i) to describe the set of interactions and events between the users or external systems (also known as actors) defining how different users will interact with the system and (ii) to document the simulation platform requirements needed to satisfy each Use Case.

There are several ways to specify Use Cases, but usually these are described using a Use Case template. It is worth mentioning that at this stage of the project, the interest is neither put on specific implementation details (i.e. how a Use Case will be implemented or executed) nor on specific outcomes which should be provided by the platform after a Use Case is executed, but rather on the specific simulation platform requirements required to support the Use Cases, as these will contribute in the selection of suitable simulator and interface architectures in the following WPs. On that note, the Use Case template used in this project is shown in Table 1.

TABLE I. THE DIGICAV USE CASE TEMPLATE

Field	Description
Objective	The objective of the Use Case
UUT inputs	The list of actions the Unit Under Test (UUT) should conform to, provided by the user
Simulation platform requirements	The minimum list of features the simulation platform should provide to enable support of the Use Case

The simulation platform requirements generated by each Use Case are then classified as functional and non-functional, which is a typical way to group system requirements in requirements engineering [38]. Functional requirements "capture the nature of interaction between the component and its environment", while non-functional ones constrain the solutions that might be considered, considering aspects such as cost, time and quality attributes like usability, efficiency, reliability, maintainability or reusability [39].

C. *User Stories*

As part of the requirements capture methodology followed, three User Stories have been created. In the context of this project, a user will most likely be a Tier 1 supplier of Level 4-5 CAV components or an OEM.

1) User Story 1 (US1): A user wants to test a CAV component with sufficient accuracy and without revealing any IP information associated with it, such as source code or other implementation details, while also taking into consideration time and cost constraints as well as usability attributes.

US1 is particularly suited to support start-up companies developing CAV components or technologies, or for testing new technological ideas around CAVs. For example, a company developing a battery management system for a hybrid (or electric) CAV aiming at optimizing their battery management algorithm using information provided from the vehicle controller require access to the vehicle controller and to the vehicle powertrain to enable this process. In such a scenario, the DigiCAV simulation platform will allow the connection of the UUT (battery management algorithm) to the platform where not only models, but also actual components of the aforementioned vehicle subsystems could be attached, hence also improving the accuracy of testing.

2) User Story 2 (US2): US2 expands US1, however, a user here is required to test a CAV component or a vehicle throughout all development phases. To enable US2, the DigiCAV platform supports a TDD approach, allowing a seamless testing process throughout all phases of development using a common test configuration.

US2 could be used as a basis for creating a new service provided by HORIBA MIRA, where the DigiCAV platform could be used to provide support to the user from the very early stage of idea generation (e.g. creating a novel control algorithm) all the way to the final stages where the component developed (e.g. a vehicle control unit) is integrated into a vehicle as well as in all other development stages in between.

3) User Story 3 (US3): In US3, the user has similar objectives to the one of US1, however, instead of testing a CAV component, the interest here is in testing a fully assembled vehicle. Hence, in the context of this User Story, a user will most likely be an OEM or a start-up company which has retrofitted a vehicle with autonomous technologies.

For example, an OEM intending to integrate different CAV components provided by Tier 1 suppliers into a new vehicle requires the use of accelerated testing procedures until vehicle certification is achieved. In such a scenario, the DigiCAV platform would allow the use of ViL testing of the fully assembled vehicle in conjunction with accelerated testing procedures, such as situation awareness and data fusion in-the-loop simulations.

D. *Use Cases*

The User Stories described in Section C led to the creation of seven Use Cases, summarized in Table 2.

TABLE II. THE DIGICAV USE CASES

Use Case Name	Use Case Objective
CAV control algorithm in-the-loop	A user wants to test a CAV control algorithm through MiL and SiL simulation tests.
Situation awareness and data fusion in-the-loop	A user wants to test situation awareness and data fusion algorithms through MiL and SiL simulation tests in a realistic CAV environment.
Situation awareness and data fusion on proving ground in-the-loop	A user wants to test situation awareness and data fusion algorithms through ViL simulation tests in a realistic CAV environment.
Vehicle model in-the-loop	A user wants to test a CAV component or an entire vehicle through vehicle model-in-the-loop simulations.
CAV control ECU in-the-loop	A user wants to test a developed CAV control ECU through HiL simulation tests.

Use Case Name	Use Case Objective
CAV on chassis dynamometer in-the-loop	A user wants to test a CAV component or an entire vehicle through chassis dynamometer-in-the-loop simulation tests.
CAV on proving ground in-the-loop	A user wants to test a CAV component or an entire vehicle through ViL simulation tests.

IV. CONCLUSION AND OUTLOOK

In this paper, the DigiCAV project was introduced and its unique approach in the development, testing and validation of CAVs was highlighted. This project will enable accelerated testing and validation of CAVs on a proving ground, their subsystems using HiL and their software components using SiL/MiL. Furthermore, the paper introduced the structure of the project, including projected goals, while also summarized results derived from the completion of WP1.

ACKNOWLEDGMENT

This work is supported by Innovate UK (grant number 133554), HORIBA MIRA Ltd. and Coventry University.

REFERENCES

[1] *Road vehicles – Functional safety – Part 2: Management of functional safety*, ISO 26262-2:2011, ISO, Geneva, Switzerland, 2011.

[2] F. Pfister, C. Schyr, and F. Le Rhun, "Total energy efficiency testing - The chassis dynamometer as a mechatronic development platform," *ATZ Worldw.*, vol. 111, no. 11, pp. 45–49, Nov. 2009.

[3] I. Passchier, G. v. Vugt, and M. Tideman, "An Integral Approach to Autonomous and Cooperative Vehicles Development and Testing," in *2015 IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 348–352.

[4] Ferrer Adria; Arrue Alvaro, "Virtual Reality for Automated Vehicles: Coupling virtual testing with real dynamic performance," in *2016 EARPA FORM Forum*, 2016, pp. 1–5.

[5] R. Chen, M. Arief, and D. Zhao, "An "Xcity" Optimization Approach to Designing Proving Grounds for Connected and Autonomous Vehicles," *CoRR*, vol. abs/1808.0, 2018.

[6] M. Cunningham and M. Regan, "Autonomous Vehicles: Human Factors Issues and Future Research," in *2015 Australas. Road Saf. Conf.*, 2015, pp. 14–16.

[7] IPG Automotive GmbH, "IPG CarMaker Simulator." [Online]. Available: https://ipg-automotive.com/products-services/simulation-software/carmaker/. [Accessed: 08-Apr-2019].

[8] F. Hendriks, M. Tideman, R. Pelders, R. Bours, and X. Liu, "Development tools for active safety systems: Prescan and VeHIL," in *Proc. 2010 IEEE Int. Conf. Veh. Electron. Saf.*, 2010, pp. 54–58.

[9] "PTV Vissim." [Online]. Available: http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/.

[10] A. Bibeka, "Evaluation of CACC Vehicles Clustering on Freeway Performance," M.S. thesis, Texas A & M Univ., College Station, TX, USA, 2016.

[11] A. Papadoulis, M. Quddus, and M. Imprialou, "Evaluating the safety impact of connected and autonomous vehicles on motorways," *Accid. Anal. Prev.*, vol. 124, pp. 12–22, 2019.

[12] Y. Lin and H. L. T. Nguyen, "Adaptive Neuro-Fuzzy Predictor-Based Control for Cooperative Adaptive Cruise Control System," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–10, 2019.

[13] Q. Cui, R. Ding, B. Zhou, and X. Wu, "Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering," *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.*, vol. 232, no. 9, pp. 1237–1252, 2017.

[14] P. Pfeffer, T. Stubler, M. Förster, M. Niegl, and R. Abhishek, "Model-based Calibration of the Lateral Controllers 'Steer-by-Angle' and 'Steer-by-Torque' of a Lane Keeping Assistance System," *FISITA 2018 World Automot. Congr.*, 2018, pp. 1459–1468.

[15] M. F. Syed Ahamed, G. Tewolde, and J. Kwon, "Software-in-the-Loop Modeling and Simulation Framework for Autonomous Vehicles," in *2018 IEEE Int. Conf. Electro/Information Technol.*, 2018, pp. 305–310.

[16] Ş. Y. Gelbal, S. Tamilarasan, M. R. Cantaş, L. Güvenç, and B. Aksun-Güvenç, "A connected and autonomous vehicle hardware-in-the-loop simulator for developing automated driving algorithms," in *2017 IEEE Int. Conf. Syst. Man, Cybern.*, 2017, pp. 3397–3402.

[17] J. Ma, F. Zhou, Z. Huang, C. L. Melson, R. James, and X. Zhang, "Hardware-in-the-Loop Testing of Connected and Automated Vehicle Applications: A Use Case for Queue-Aware Signalized Intersection Approach and Departure," *Transp. Res. Rec.*, vol. 2672, no. 22, pp. 36–46, 2018.

[18] R. K. Bhadani, J. Sprinkle, and M. Bunting, "The CAT Vehicle Testbed: A Simulator with Hardware in the Loop for Autonomous Vehicle Applications," *Electron. Proc. Theor. Comput. Sci.*, vol. 269, pp. 32–47, 2018.

[19] National Instruments, "Altran and NI Demonstrate ADAS HIL with Sensor Fusion," White Paper, 2017.

[20] D. A. Hall, "The challenges of HIL testing and ADAS," Design World, pp. 14–17, Aug. 2017.

[21] IPG Automotive GmbH, "ESC-HIL - turnkey solution for testing and validating real-world ESC units.", Karlsruhe, Brochure, 2016.

[22] H. Waschl, R. Schmied, D. Reischl, and M. Stolz, "A Virtual Development and Evaluation Framework for ADAS–Case Study of a P-ACC in a Connected Environment," in *Lecture Notes in Control and Information Sciences*, 2019, pp. 107–131.

[23] M. Gadringer, S. Metzner, and H. Schreiber, "Radar stimulation for a novel ADAS system," in *Graz Symp. VIRTUAL Veh.*, 2017.

[24] C. Schyr and A. Brissard, "DrivingCube – A novel concept for validation of powertrain and steering systems with automated driving," in *13th Int. Symp. Adv. Veh. Control*, 2016, pp. 79–84.

[25] Z. Szalay, T. Tettamanti, D. Esztergár-Kiss, I. Varga, and C. Bartolini, "Development of a Test Track for Driverless Cars: Vehicle Design, Track Configuration, and Liability Considerations," *Period. Polytech. Transp. Eng.*, vol. 46, 2017.

[26] Y. Hou *et al.*, "An integrated traffic-driving simulation framework: Design, implementation, and validation," *Transp. Res. Part C Emerg. Technol.*, vol. 45, pp. 138–153, 2014.

[27] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-Simulation: A Survey," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 49:1–49:33, May 2018.

[28] T. Jung, P. Shah, and M. Weyrich, "Dynamic Co-Simulation of Internet-of-Things-Components using a Multi-Agent-System," *Procedia CIRP*, vol. 72, pp. 874–879, 2018.

[29] A. Elsheikh, M. U. Awais, E. Widl, and P. Palensky, "Modelica-enabled rapid prototyping of cyber-physical energy systems via the functional mockup interface," in *2013 Work. Model. Simul. Cyber-Physical Energy Syst.*, 2013, pp. 1–6.

[30] S.-Y. Wang and C.-L. Chou, "NCTUns Simulator for Wireless Vehicular Ad Hoc Network Research," in *Ad Hoc Networks: New Research*, 2009, pp. 97–123.

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

[32] T. Blochwitz *et al.*, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models," in *8th Int. Model. Conf.*, 2011, pp. 105–114.

[33] *Functional Mock-up Interface for Model Exchange and Co-Simulation,* FMI 2.0, FMI development group, Modelica Association Project FMI, 2014

[34] M. Krammer, N. Marko, and M. Benedikt, "Interfacing Real-Time Systems for Advanced Co-Simulation – The ACOSAR Approach," in *CEUR Workshop Proc.*, 2016, pp. 32–39.

[35] M. C. Figueiredo, R. J. F. Rossetti, R. A. M. Braga, and L. P. Reis, "An Approach to Simulate Autonomous Vehicles in Urban Traffic Scenarios," in *12th Int. IEEE Conf. Intell. Transp. Syst.*, 2009, pp. 322–327.

[36] K. Beck, *Test Driven Development: By Example*, 1st ed. Reading, MA, USA: Addison-Wesley, 2002.

[37] D. Astels, *Test-Driven Development: A Practical Guide*. Englewood Cliffs, NJ, USA: Prentice-Hall (Inc.), 2003.

[38] K. Curcio, T. Navarro, A. Malucelli, and S. Reinehr, "Requirements Engineering: a systematic mapping study in agile software development," *J. Syst. Softw.*, vol. 139, no. 6, pp. 32–50, 2018.

[39] R. E. Gallardo-Valencia, V. Olivera, and S. E. Sim, "Are Use Cases Beneficial for Developers Using Agile Requirements?," in *2007 Fifth Int. Work. Comp. Eval. Requir. Eng.*, 2007, pp. 11–22.