Simulation-based Verification for Autonomous Driving

Greg Chance, Abanoub Ghobrial, Kerstin Eder
Trustworthy Systems Laboratory
Department of Computer Science
University of Bristol, UK

University of BRISTOL

Bristol Robotics Laboratory

robopilot

Capri

Visit the Trustworthy Systems Laboratory page for information on all our projects and research activities

https://www.bristol.ac.uk/engineering/research/trustworthy-systems-laboratory/

The Challenge

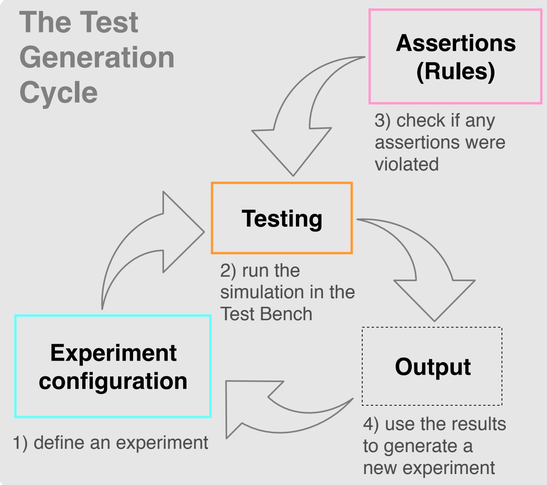
We are investigating techniques for simulation-based verification of Autonomous Vehicles (AV). We are not developing an AI or control system for the AV, but a framework in which they can be tested against a formal set of rules (**Assertions**) in a simulated environment. **Assertions** may be the rules of the Highway Code and social convention. Violation of **Assertions** means that the AV has taken illegal, dangerous or other improper actions. These sequences of actions, or traces, can later be used to test other manufacturers' AVs. Our aim is to develop test generation techniques that generate tests that provide coverage for all driving assertions faster than by using random test generation **[2]**. Doing this in a simulation allows complete control over the way in which the AV may be tested including safely subjecting it to dangerous situations that may not be feasible in the real world. The research challenges we address are; **Test Generation**, **Coverage** and **Checking**.

Test Generation
How can we generate effective tests most efficiently?

Coverage
How do we ensure that all the assertions have been tested in all interesting conditions?

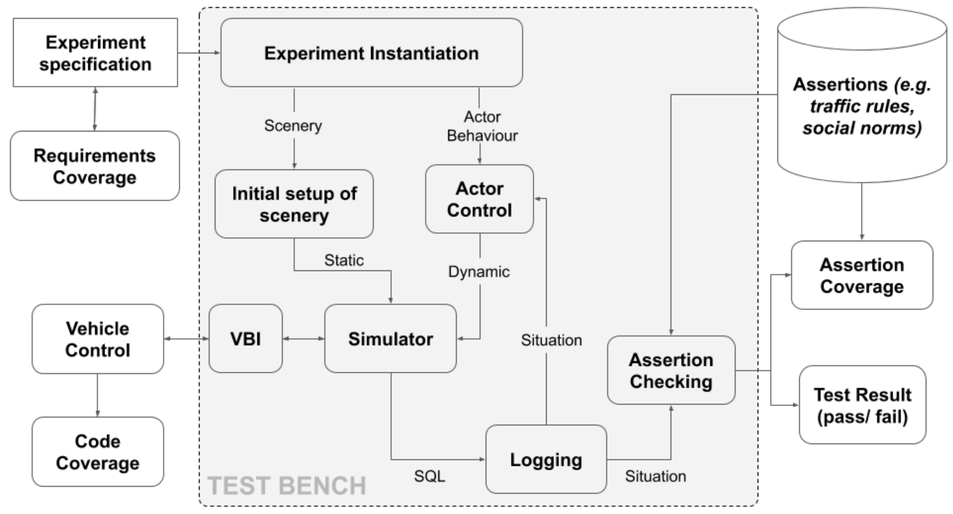
Checking
How do we know if the autonomous vehicle has violated any assertions?

The Test Generation Cycle



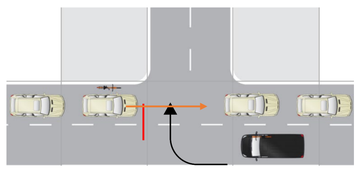
Test Bench

The Test Bench is the heart of the simulation-based verification system which is primarily driven from the **Experiment** specification to setup the **Map**, **Environment** and the **Actors** in the simulator (Carla). The action of all the dynamic agents within the simulation are logged in a relational **Database** and tested for assertion violations resulting in a pass/fail condition. It is expected that many thousands of experiments will be required over a range of different experimental conditions in order to check assertions over a variety of road layouts and environmental conditions. **Scene** refers to all static objects and a snapshot of dynamic elements. **Dynamic** elements are the elements in a scene whose actions or behaviour may change over time. The **Scenario** is defined as a temporal development between several scenes. A **situation** is defined as the subjective conditions and determinants for behaviour at a particular point in time **[6]**.



Accidentology

Accidentology is the study of the causes and effects of accidents and can be used to infer dangerous carriageways based on the accident severity and according to the type of road user. This data will be used to drive the selection of initial **Experiment** conditions including the **Scene**, **Map** and **Environment**.

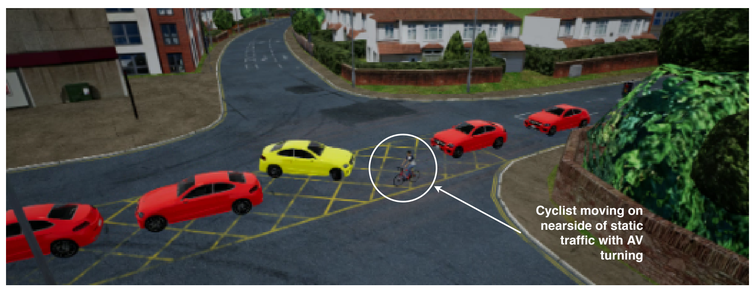


Experiment

The experiment is the highest level of specification that will drive the **Test Bench** including the **Scene**, **Map**, **Environment** and all **Actors**. Generation of successful tests will result in **Assertion Coverage**. A range of experiments over a variety of conditions (scenarios) will move the testing of the autonomous vehicle towards verification.

Scene

The scene is defined as a snapshot of a temporal development of a scenario. The Scene contains the static elements of the **Experiment** such as the **Map**, buildings, street furniture and foliage. The lighting and weather conditions also form part of the **Environment** of the scene. Dynamic components (**Actors**) may be placed within the scene and given specific missions (see **Actor State Vector**) during the simulation. Weather and lighting conditions also form part of the scene.

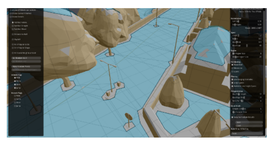


Simulation-Based Testing Pipeline

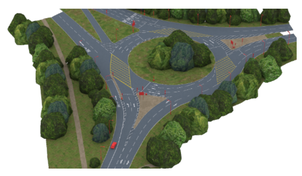
The simulation requires a representative map environment and the dynamic actors within it. Maps are generated from OSM data which are turned from network splines to rich environments containing details on the road network lanes [5] to details of road markings, lamp-posts and signs. Using the OpenDrive format a description of the road and pavement positions can be given with geo-referenced location. Actors in the simulation can be animated in a number of ways such as using the multi-agent systems paradigm or they can be taught to complete certain objectives to achieve assertion coverage for example. Simulation data is logged in a database and can be queried to observe correct AV actions and compliance with driving regulations.

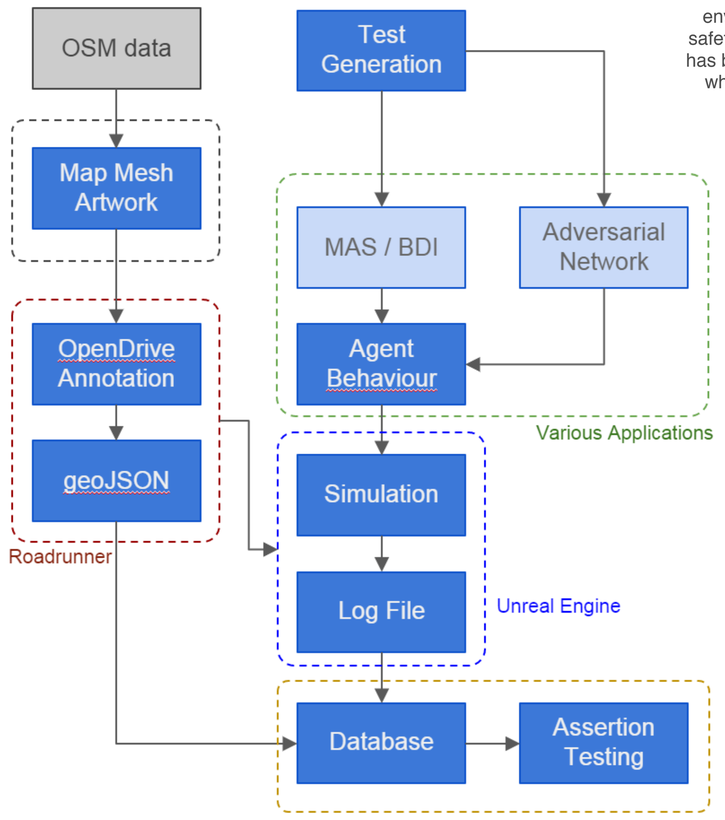
Open Street Map (**OSM**) data is an open resource for user contributed map data including road networks, buildings and topography.

Simulation maps can be supplemented with pedestrian navigation meshes to enable natural pedestrian movement.



OSM data is not detailed enough for visual simulations so artwork is generated of the road network using satellite imagery.

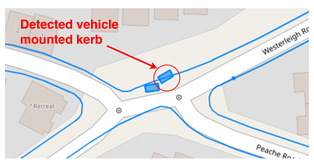




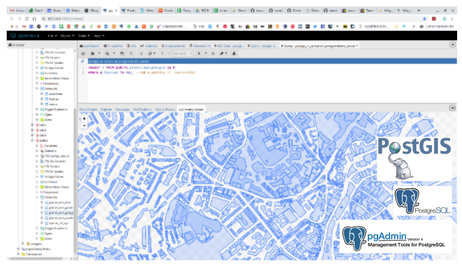
Both **Capri** and **Robopilot** projects are focused on urban environments and pedestrians are a key consideration for driving safety. Control of pedestrians (and other actors) is a challenge that has been approached from the perspective of Multi-Agent Systems where each actor is given a specific goal during the simulation to encourage natural or edge case scenario exploration.

The open source **Carla** project provides a driving research interface to the game development platform Unreal Engine. Carla provides useful tools for AV researchers, abstracting control of vehicles, pedestrians and traffic flow simulation. Lidar, radar and other perception stack tools such as object segmentation are also included.

OpenDrive and **geoJSON** is an open format for map information that includes information beyond that of OSM data, such as lane boundaries and kerb positions. This information is used during simulation for assertion checking, e.g. ensuring vehicle does not mount the pavement.



A postGIS **database** will be used to log the position and orientation of the **AV** and all other **Actors** in the **Scene** at the rate of the simulation. This includes all road network information, traffic light position and states, weather conditions, actor waypoints or mission goals.



Assertion checking is done via SQL queries to the postGIS geo-spacial database. **Assertions** can be checked for violations against a library of **driving rules**, **laws** and **social conventions**. Queries such as "did the AV pass a red traffic light" or "did the AV adhere to the local speed limit" can be monitored online or new assertions can be added and checked offline. This will be useful for assuring vehicle safety and satisfying auditors and compliance regulators.

Data Output | Explain | Messages | Notifications

	id [PK] integer	agent_id integer	sim_time double precision	asr_id integer	asr_result boolean
1	1	78	2	1	true
2	2	76	2	1	true
3	3	77	2	1	true
4	4	79	2	1	true
5	5	80	2	1	true
6	6	81	2	1	true
7	7	78	2	2	true
8	8	76	2	2	true
9	9	77	2	2	true
10	10	79	2	2	true
11	11	80	2	2	true

Actor identifier
Actor type (car, pedestrian, cyclist etc.)
Simulation time (s)
Assertion identifier
Assertion checking result logic True=Assertion passed

Environment

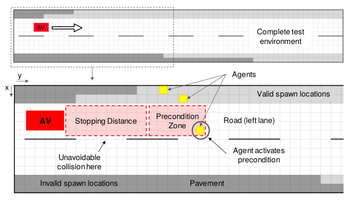
Conditions inside the simulation can be altered to replicate changes in illumination levels (day/night) including street lighting and different weather conditions such as rain or fog. These conditions can be parameterised as inputs to the **Test Bench**.

i) bright & low sun ii) raining iii) dark with lamps iv) dense fog

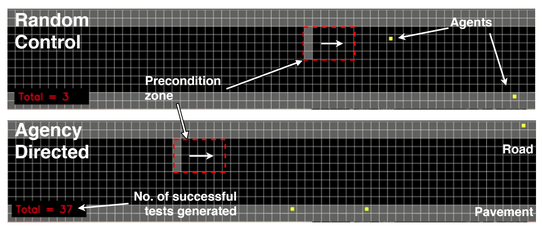
Agency Directed Test Generation [1]

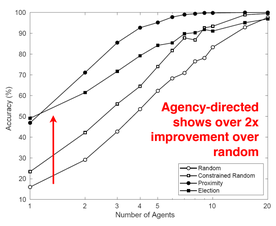
We addresses the challenge of efficiently generating effective tests for simulation-based AV verification using **software testing agents**. The multi-agent system (MAS) programming paradigm offers rational agency, causality and strategic planning between multiple agents **[3]**. We exploit these aspects for test generation, focusing in particular on the generation of tests that trigger the precondition of an assertion (collision avoidance with a pedestrian NCAP-25).

Our aim is to generate tests that exercise an assertion that requires the AV to avoid collisions with other road users, provided that such road user, a pedestrian in this case, intrudes in the path of the AV in such a way that a collision can be avoided by the AV either by braking or manoeuvring (Precondition Zone).



For the purpose of comparing the performance of agent-based test generation techniques, the test agents have different behaviours from simple **random** (i.e. no direction through agency) to more directed, strategic **planning-based agency**.





On the example of a key assertion on collision avoidance we show that, by encoding a variety of different behaviours respondent to the agent's perceptions of the test environment, the **agency-directed approach generates twice as many effective tests** than a pseudo-random approach.

Code for our agency directed test generation experiment can be found at https://github.com/ TSL-UOB/CAV-MAS

Future Work

In future work we aim to extend our existing agent models to work over multiple assertions. We also intend to generate new agent behaviour models and move towards an **agent-based coverage-directed** approach to test generation where agent goals are updated based of coverage feedback. Including personality in agents is also another avenue that could provide a tuning parameter **[4]** to generate edge cases. We are also investigating if AV research platforms (e.g. Carla) are deterministic. CAV cyber security is also the focus of upcoming projects which may be tested using the simulation and test generation approach outlined here.

Actors

Actors, in the form of cars, cyclists, HGV's, pedestrians etc., will be an **Experimental** level parameter and form the dynamic part of the simulation. Assertion violations will most likely occur from the interaction of the AV with other actors.

Autonomous Vehicle (AV)

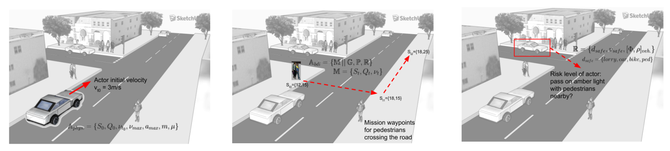
Ego Vehicle is a term used to describe the autonomous vehicle (AV) and the AI controlling it. The software for the vehicle is being provided by our project partners which will be used directly in the simulation.

As part of the **Robopilot** project, a model of an autonomous electric delivery truck by Arrival will replace a standard diesel T4 HGV.

As part of the **Capri** project an electric driverless pod will shuttle passengers in airports from car parks to their departure gates.

Actor State Vector

The Actor State Vector contains all information about any **Actor** within the simulation such as cars and cyclists but also includes items such as traffic lights, automated road signs and moving barriers as their actions can change over time. These vectors will be used to encode actions of the agents, such as specific missions and how they behave around other actors and also form the basis of how the information about the experiment is logged in the **Database** and verified for **Assertion** violations.



$\mathbf{A}_{phys.} = \{S_0, Q_0, \nu_0, \nu_{max}, a_{max}, m, \mu\}$

Initial Position and orientation
Initial velocity
Maximum velocity and acceleration
Mass and coefficient of friction

$\mathbf{A}_{bdi} = \{\mathbf{M} || \mathbf{G}, \mathbf{P}, \mathbb{R}\}$

Specific Mission
Non-Deterministic Goal
Agent actions
Agent risk factors

$\mathbf{M} = \{S_i, Q_i, \nu_i\}$
A specific mission predetermined with waypoints, orientations and velocities, this mission could work well with basic proposition language.

$\mathbf{G} = \{S_{goal}, S_{tour}\}$
A goal that the agent may complete in a non-deterministic manner appropriate to the simulation, i.e. get to location, or touring (delivery, tourist), i.e. visit these roads.

$\mathbb{P} = \{Accelerate, Turn\}$
The possible actions of the agent

$\mathbb{R} = \{d_{safe}, v_{safe}, [\Phi, \rho]_{veh.}\}$
The risk adversity of the agent including the safe distances (proxemics) and speeds when interacting with other road users, flux and density of vehicles (people may cross a road when very empty but also very full) - i.e. when is it safe to cross the road, change lanes or move from a junction, cross red/amber light

Our latest paper to be presented at IEEE AI Test 2020:
[1] Chance, G., Ghobrial, A., Lemaignan, S., Pipe, T., & Eder, K. (2019). An Agency-Directed Approach to Test Generation for Simulation-based Autonomous Vehicle Verification. http://arxiv.org/abs/1912.05434
[2] Koopman, P., & Wagner, M. (2018). Toward a Framework for Highly Automated Vehicle Safety Validation. SAE Technical Paper Series, 1, 1–13.
[3] M. Georgeff, "Communication and interaction in multi-agent planning," in Readings in distributed artificial intelligence, pp. 200–204, Elsevier, 1988.
[4] A. Zoumpoulaki, N. Avradinis, and S. Vosinakis, "A Multi-Agent Simulation Framework for Emergency Evacuations Incorporating Personality and Emotions," in Hellenic Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2010., pp. 423–428, 2010
[5] Bender, P., Ziegler, J., & Stiller, C. (2014). Lanelets: Efficient map representation for autonomous driving. IEEE Intelligent Vehicles Symposium, Proceedings, (Iv), 420–425.
[6] Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., & Maurer, M. (2015). Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2015-Octob, 982–988. https://doi.org/10.1109/ITSC.2015.164

Visit the Trustworthy Systems Laboratory page for information on all our projects and research activities

https://www.bristol.ac.uk/engineering/research/trustworthy-systems-laboratory/

University of BRISTOL

Innovate UK

We acknowledge the support of Innovate UK projects 103288 and 103703.