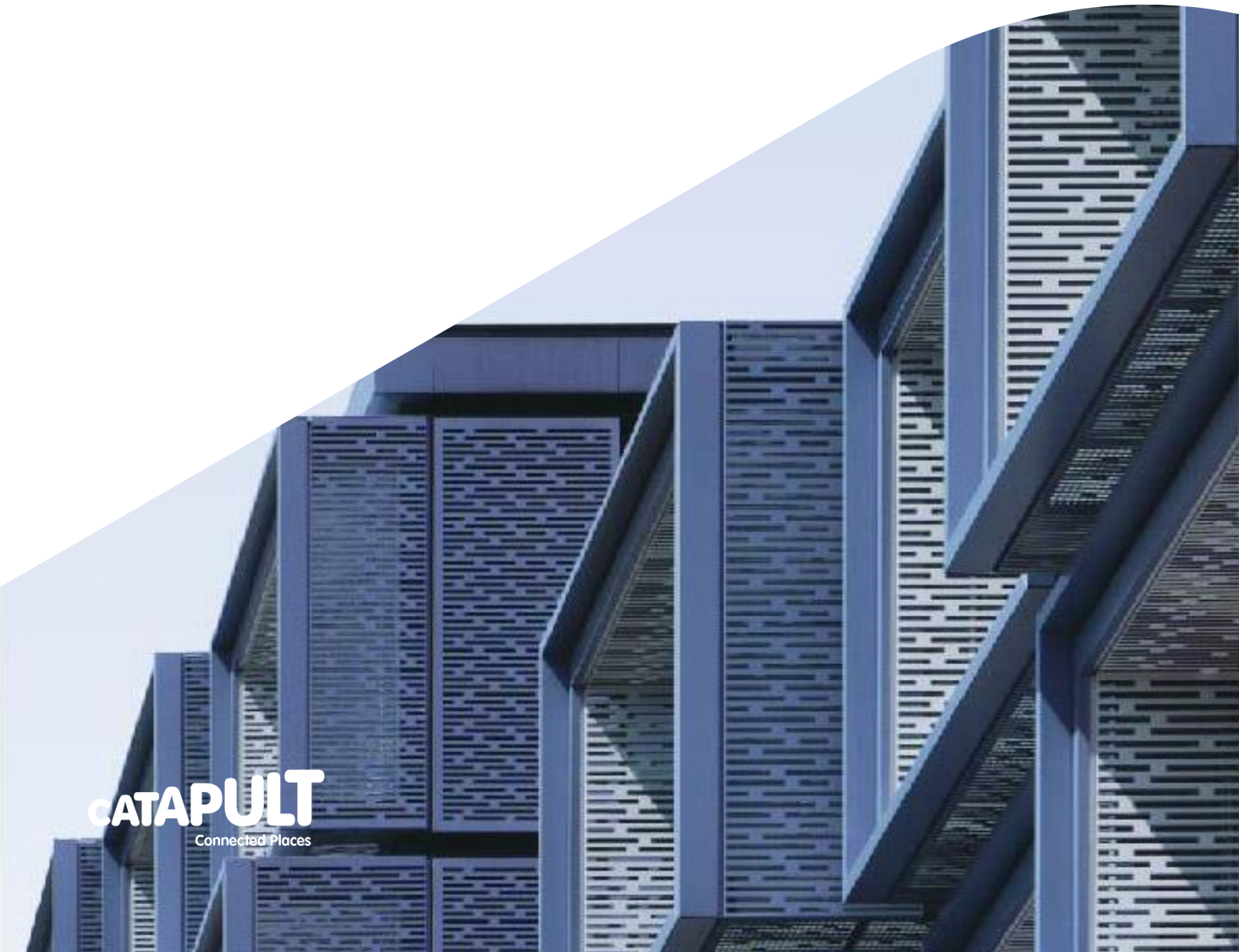
March 2021

# CertiCAV Software Framework

## Implementation and evaluation



CATAPULT Connected Places

# Executive summary

## Simulation-based and other automated test methods appear likely to play an important role in verification and validation of Highly Automated Vehicles (HAVs). This report examines some of the practical aspects of using these methods as part of an assurance process.

To gain an understanding of the process, example pass/fail tests were identified to help satisfy a plausible set of requirements. These were implemented in a simulation framework which combined an existing open-source scenario database (MUSICC) and driving simulator (CARLA). The key findings were:

- At least two categories of test are practical to include in an automated test framework. These are tests based on formal specifications (which rigorously define exactly which behaviours are acceptable) and those based on comparison of performance metrics to a reference model.
- Formal specifications are easy to measure but hard to define. Initial attempts are likely to be incomplete and imperfect but can still have an important role to play. We propose two key roles:
    - A safety responsibility specification could be used to show that an Automated Driving System (ADS) will not cause a crash by providing a formal definition of 'cause'. This sets the foundations for a 'vision zero' approach, where ADSs from different manufacturers can be assured not to crash into each other. However, the nature of this specification means it will be difficult to alter over time, as all vehicles on the road will need to use a compatible version.
    - A common cooperative driving specification, which formalises selected aspects of existing traffic rules. This could initially capture aspects of good driving which are easy to describe formally and be extended over time.
- Using metrics to compare to a reference system has some promise, but the demonstrated approach of comparing performance of an ADS to a human driver on a per-scenario basis has significant limitations. In particular, there is a risk of requiring an ADS to mimic the human driver's behaviour too closely and there is a practical limit to the number of scenarios which can be tested in this way.

- For the foreseeable future, subjective judgement will be required for some aspects of testing. For example, if a specification cannot be proven to be perfect, there may be a need to justify non-compliance based on a risk assessment.

Topics which are likely to be of interest for future work include initial definition of formal specifications, standardisation of data exchange formats and further development of the metric-based comparison concept.

# Contents

# 1 Scope

CertiCAV was a project undertaken by Connected Places Catapult and WMG, University of Warwick, on behalf of the Department for Transport's International Vehicle Standards Division[1]. It aimed to identify principles and methods relating to the type approval and safety assurance of Highly Automated Vehicles (HAVs). The project only considered the safety of the perception and tactical decision making of a nominally performing HAV (i.e. it does not attempt to address operational safety or component failures).

This report relates to the design and development of a software framework as part of the CertiCAV project. Source code for this can be found at https://gitlab.com/connected-places-catapult/CertiCAV. The software represents a proof-of-concept implementation of automated testing methods for HAV assessment, applied in a way which is compatible with the wider CertiCAV framework (which is currently unpublished). Developing it was intended to generate knowledge about how to practically apply this type of testing, to inform both the rest of the CertiCAV project and any future work based on its outputs. Implementation decisions have been based on an understanding gained from published and unpublished research, including a previous Catapult report on simulation frameworks (Saigol, Peters, Barton, & Taylor, 2018).

Development and approval of an Automated Driving System (ADS) is likely to require many types of testing involving several different stakeholders. These might include subsystem suppliers, ADS developers, independent test organisations and regulators. When developing the software, we decided to focus on a use case where test scenarios are externally defined but implemented by an ADS developer using their own systems. In this use case, the ADS developer would produce raw-data results, which could either be evaluated internally or to be made available for independent evaluation. The results would be stored in a format which makes it possible for an independent body to reproduce the evaluation if required. While this example has been used to define requirements, many of the findings in this report (and resulting software functionality) will apply to other use cases as well.

[1] This document does not necessarily represent the views of the Department for Transport or any UK government body.

# 2 Defining good driving

In order to design tests to measure good driving, a definition of what good driving means is required. We approached this in 3 steps:

- We created a qualitative description of good driving. This is the set of driving performance criteria shown in Table 1, which is derived from those published as part of a previous project (Myers & Saigol, 2020).
- We defined the concept of a performance indicator. These are metrics which can be used to quantify outcomes identified as desirable.
- Finally, we proposed that useful requirements for HAV verification will need to define conditions (apply to a subset of possible outcomes), limitations (e.g. a maximum or minimum rate of occurrence) and a measure of the level of certainty required (which may vary between requirements, e.g. requirements relating to traffic flow may require a lower level of assurance than those relating to an ADS causing collisions). They will also need to be quantified in terms of performance indicators. Setting values for these items is outside the scope of CertiCAV, but where required we have selected plausible values for the demonstration work reported here.
    - In some cases, limitations defined as part of a requirement may be trivial. For example, a requirement which must be met all the time is equivalent to setting a condition of non-conformance with an acceptable rate limit of zero.
    - Levels of certainty may be defined directly (e.g. numerically) or by reference to a methodology and outcome which will result in an acceptable level.

Requirements will specify how the vehicle should perform after deployment. Pre-deployment testing takes place under different conditions, which means it is effectively a controlled experiment to predict whether the vehicle will meet its requirements post-deployment. The conditions of a test are different to deployment and depend on the techniques used for testing, so a further step is required to define pass/fail criteria. In the case of the CertiCAV framework, features of the test technique which need to be taken into account when defining pass/fail criteria include:

- Which subsystems of an ADS are being tested (e.g. whether low-level control capability is assessed alongside tactical decision making)
- Whether the set of scenarios is statistically representative of the operating domain
- Whether some scenarios may be outside of the vehicle's operational design domain.

In Section 4, we develop an example set of pass/fail criteria which relate to a plausible set of requirements. The relationships between driving performance criteria, performance indicators, requirements, techniques used for testing and pass/fail criteria are illustrated in Figure 1.

Table 1: Driving performance criteria

| Ref | Criterion | Description |
|---|---|---|
| 1 | Do not cause harm | When conditions (e.g. other road user behaviour) are reasonable then collisions are not acceptable. The threshold of acceptance for HAV to be at fault for a collision is very low. |

| Ref | Criterion | Description |
|---|---|---|
| 2 | Avoid harm, even when not the cause | Expected to allow for possibility of unreasonable behaviour by others, or hazardous conditions. The expectation is that a HAV should be able to detect and avoid/mitigate hazards with a performance at least equal to a reasonable human driver. |
| 3 | Provide occupant safety/comfort | For an omnibus driving smoothly is an important safety factor, for vehicles with seat-belted occupants this is limited to comfort unless accelerations are severe. |
| 4 | Follow Traffic Rules | HAV is expected to comply with road traffic regulations |
| 5 | Provide reasonable safety margins | HAV is expected to maintain safety margins that are safety resilient for both operation of HAV and appear reasonable to other road users. |
| 6 | Follow recommended driving practice | Expected to exhibit good driving behaviour, following guidance in highway code and other relevant practice. |
| 7 | Facilitate established driving conventions | Expected to reasonably blend in with established driving practice, not to confuse or unreasonably hinder other road users. |
| 8 | Behave considerately to other road users | Expected to follow courteous driving practice to facilitate smooth and harmonious experience for all road users. |
| 9 | Do not unreasonably obstruct movement of traffic | Being overly hesitant/cautious and on a larger scale has environmental/economic impact on traffic logistics. |

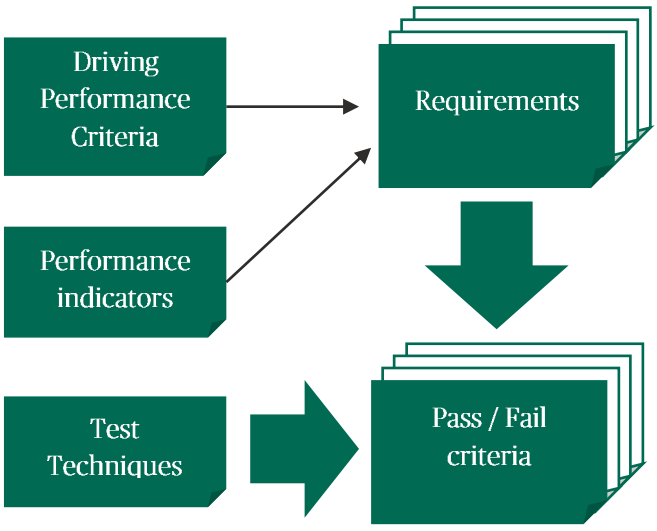



Figure 1: process for defining pass/fail criteria

# 3 Role of automated behaviour testing

For the purposes of this report, automated behaviour testing is taken to mean testing which:

- Attempts to measure the success of an ADS's tactical decision-making using a set of test scenarios.
- Takes place largely in simulation. This could include hardware in the loop testing but not the use of test tracks or real-world test drivers.
- Measures the performance of systems in a largely automated way using a well-defined methodology (i.e. is not reliant on the subjective judgement of a human assessor).
- Relates to requirements of the type defined in section 2. These contain conditions, limits and levels of certainty and are expressed in terms of performance indicators.

This type of testing has some advantages over alternative methods. It is highly repeatable (because testing takes place in simulation) and ensures that performance is measured in a consistent and unbiased way (as the evaluation itself does not include any element of human judgement). Use of simulation also allows for the ADS to be tested in virtually any scenario, even those which may be difficult or dangerous to make happen on a test track. It is reasonable to expect that the cost per test will be relatively low (in time and resource as well as money), allowing a much wider range of situations to be tested than would otherwise be possible. The cost is not zero though, which means there is still a limit to how many scenarios can be tested. Since an effectively infinite number of scenarios can occur in the real world, testing can only ever reduce the risk of an 'unknown-unsafe' event occurring after deployment, not eliminate it.

The need for a robust performance measurement is both a strength of this approach and a weakness. It ensures repeatable and predictable outcomes but also limits the type of testing which can be carried out. In CertiCAV, two categories of performance measurement approach were demonstrated: formal specifications and outcome scores.

Formal specifications provide a precisely defined boundary on behaviour, where any outcome can be mathematically shown to be either compliant or non-compliant. These can be thought of as easy to measure but hard to define. For example, requiring a HAV to maintain a minimum distance behind the vehicle in front at all times would qualify as a formal specification, but defining what that distance should be for every traffic scenario is challenging. Currently, few road traffic laws meet the level of precision needed to be considered a formal specification and defining a formal specification which captures all aspects of good driving is probably a bigger challenge than creating an acceptable quality ADS. However, this does not mean that incomplete or imperfect formal specifications cannot still be useful. A few different approaches to using them are possible:

- Formal specifications could be used to define behaviour which is certainly unacceptable. This is achieved by narrowing the scope of the situations where the specification applies (e.g. to a known set of scenarios where the correct behaviour is obvious). Compliance with these specifications would be necessary but not sufficient for approval.
- Evidence on compliance with formal specifications could be used as an input to a system based on human judgement. For example, a formal specification could be created with an

acceptance that it may not be perfect in all situations (e.g. there may be conflicting rules or following the rules may not always result in the best outcome). The ADSE could be asked to justify any deviations from the specification to the regulator in advance of testing.

- Requirements could be defined which provide a non-zero limit for the tolerable rate of non-conformance with a specification. For example, a regulator could state that rule breaking may be tolerated only if it occurs much less often than it would with a human driver. The rate would be set to allow enough margin for non-conformance with the specification in situations where applying it strictly does not lead to the most desirable outcome. However, a purely rate-based approach does not require a judgement on exactly which situations those are.

In practice a hybrid of these approaches may make sense. Different specifications could exist with different levels of justification required for non-conformance, with a maximum level of non-conformance set regardless of justification.

Outcome scores attempt to measure harms and/or benefits from a piece of driving. In themselves, they do not define whether that driving was good considering the circumstances. For example, in section 4.4 we propose an outcome scoring method which estimates the level of harm caused by any collisions in the scenario. In some situations, a minor collision might be a good outcome while in others it would be unacceptable. Our suggestion is that outcome scores are a good way to compare the system under test to a 'reference system', which could be a human driver, a formal specification of the performance characteristics which are considered achievable or even an alternative ADS.

# 4 Performance test concepts

In this section we combine the likely contents of system requirements (from section 2), with the capabilities of automated behaviour testing (outlined in section 3). Three types of test are proposed which could provide evidence on whether the requirements will be met. An example is given for each type of test. Implementation challenges are discussed both for each category of test and each example.

## 4.1 Example scenario

The examples of each test type are discussed in the context of the scenario shown in Figure 2, though they can also be applied more generally. In this scenario the vehicle under test ("Ego") is travelling on the main road past a priority junction, at which an actor vehicle ("Challenger") is waiting to pull out. When Ego reaches a certain distance from the junction (based on a value randomly selected when the concrete scenario is created), Challenger will pull out onto the main road and accelerate slowly.

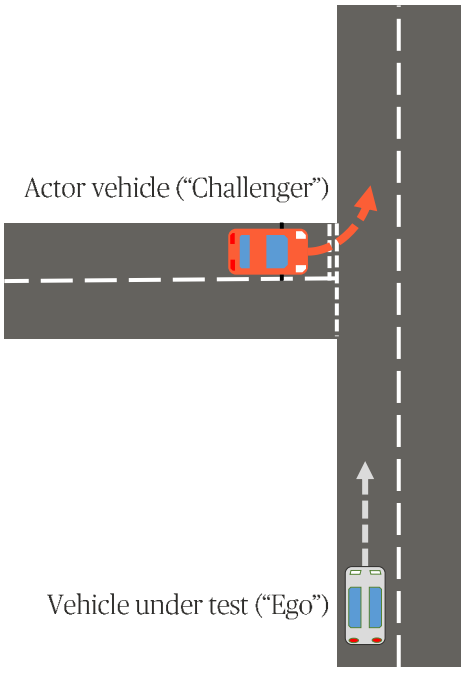



Figure 2: Example scenario

## 4.2 Safety responsibility specification tests

### 4.2.1 Summary of test purpose

Safety Responsibility Specification (SRS) tests are designed to test requirements based on the first driving criterion, "Do not cause harm". This is assumed to be based upon a formal specification common to all automated vehicles. Our implementation is related to the idea of "reasonable behaviour" proposed in Mobileye's Responsibility Sensitive Safety (RSS) concept (Shalev-Shwartz, Shammah, & Shashua, 2017). "Reasonable behaviour" is a set of assumptions about how road users

behave. If all vehicles conform to these assumptions, no collisions should occur. If one actor violates these assumptions and a crash results, it could be considered responsible for it.

### 4.2.2 Example of test application

A test based on a safety responsibility specification is applied to Challenger pulling out of the junction in the example above (section 4.1). If Challenger pulls out sufficiently close in front of Ego, Ego is not considered responsible for the resulting collision.

The CertiCAV algorithm is a little different from the RSS version for ease of explanation and implementation. It applies to only the specific case of a priority junction where vehicles may turn either left or right. It refers to the "priority vehicle" and "non-priority" vehicle rather than Ego and Challenger, as it may also be applied to scenarios where the vehicle under test does not have priority. The key steps of the process are summarised below:

1. Check whether any collisions occurred. Collisions result in a failure unless caused by another vehicle breaking a reasonable assumption.
2. Check whether both vehicles involved in a collision previously passed through the conflict area of a priority junction.
3. If the non-priority vehicle pulled out of that junction within the stopping distance[2] of the priority vehicle, mark its behaviour as unreasonable.

### 4.2.3 General implementation challenges for this test type

Implementing this example test has raised some challenges which are likely to be common to many tests of this type. While these are not specific to the example scenario or test, it is used here to explain them. A list of issues is set out below:

- Tests need to be defined in a way which does not require knowledge of the internal state of the ADS. Defining the specification in terms of reasonable assumptions is an attempt to overcome this, but these assumptions may not always be intuitive. Other vehicles can be expected to react to the behaviour of the vehicle under test, so it may be necessary to define the reasonable assumptions it can make about other vehicles in terms of its own behaviour (e.g. in the example above, it might not be reasonable to assume that Challenger will not pull out if Ego is indicating to turn into the same junction). These types of assumption add complication but may be required to create a specification which is a reasonable match with human driver expectations.
- It is surprisingly difficult to define responsibility formally, even in apparently simple situations. The algorithm outlined above would work poorly at a slip road junction, for example. Other algorithms, such as those proposed in Mobileye's RSS concept, may do better, but even these could require some changes to parameter values to work well in all different situations and road layout. IEE working group P2846[3] is working on defining industry standard assumptions in this area.
- It is difficult to formally link a collision to the unreasonable behaviour which caused it. For example: challenger pulls out close in front of Ego, which successfully avoids a collision. Some time later, Ego crashes into Challenger for no apparent reason. The CertiCAV algorithm would

[2] A conservative estimate of stopping distance is used, based on the UK standards for visibility at non-priority junctions. These mean that the priority vehicle should only ever have to brake gently to avoid a collision, even if the non-priority vehicle pulls out of the junction and then stops blocking the road.

[3] https://sagroups.ieee.org/2846/

label this as a collision caused by Challenger, as it cannot identify that a safe state has been recovered in the meantime.
    - One solution could be to adopt the concept of safe and unsafe states proposed as part of the RSS concept. This defines a state where no reasonable assumptions are currently violated as 'safe'. Once a safe state is regained, any previous unreasonable behaviour cannot be considered to have caused a crash. An implication of this is that all reasonable assumptions made by the Ego vehicle need to be known by the test script, even those relating to its own performance.
- There is a sharp transition between reasonable and unreasonable behaviour which creates an issue when measurements are imperfect. In the example above, a small error in measuring the point at which the vehicle pulled out (such as the kind of error introduced by using quantised units for position and time) could result in either, or neither, vehicle being labelled as responsible.
    - One way to mitigate this is to use asymmetrical assumptions, with the Ego required to be more cautious about the behaviour of others than it assumes others will be about its own. In this case, Ego should assume that Challenger's estimate of a reasonable distance to pull out at is slightly lower than the one it would use in the same situation.
- Similarly, different human drivers have different perceptions of reasonable, not all of which will be consistent with any formal specification. This is an issue, because HAVs need to be compatible with manually driven vehicles. A specification can help to accommodate this by making the assumptions more asymmetrical than is required to manage measurement error alone. This means a scenario would be possible where both vehicles could be considered to have behaved unreasonably, but if either is the vehicle under test it will have violated the specification.

### 4.2.4 Pullout test specific implementation challenges

Some implementation challenges are directly related to the task of assessing vehicles pulling out of junctions. They will not necessarily apply to other safety responsibility specification tests, but common themes may emerge when further examples are attempted. These are explained below:

- The current algorithm does not account for the possibility of one vehicle being unable to see the other because of another object blocking the view (occlusion). For this to be taken into account directly, the priority vehicle needs to make an assumption about the non-priority vehicle's field of view. It may not be reasonable for the priority vehicle to assume that the conflict area of a junction will remain clear if the non-priority vehicle has not had an opportunity to observe it.
    - Reasonable behaviour with restricted visibility will need to be defined carefully. For example, it may be reasonable for the non-priority vehicle to wait for another vehicle which is obscuring the view to move out of the way. However, this is unlikely to be practical if the occlusion is caused by a static object. The most problematic situations appear to be those where two vehicles which cannot see each other are moving towards the same space.
- The algorithm outlined above requires a common definition for the conflict area of a junction. This could be generally defined based on the junction layout or an accepted reference road network file. For simplicity, the example implementation specifies it directly for each junction.
- The correct behaviour at a slip road is a little different to a T junction, but these road layouts are logically quite similar (both are forms of priority junction). We tentatively suggest that a key distinction is the possibility of non-priority vehicles turning right at a T-junction. This

means that the priority vehicle must allow for the possibility that vehicles approaching the junction may turn towards it.

## 4.3 Common-cooperative driving specification tests

### 4.3.1 Summary of test purpose

The SRS tests described in section 4.2 attempt to measure whether the vehicle under test caused a crash, but current Highway Code and road traffic laws go well beyond this. Drivers are also expected to comply with requirements which help traffic to flow better and mitigate risks caused by the actions of others. The concept of a 'Common-Cooperative Driving Specification' (CCDS) is intended to cover aspects of good driving which can be developed into a formal specification and defined by a set of rules. It would operate alongside a safety responsibility specification, which only requires the ADS not to cause a crash. The CCDS relates primarily to driving criterion 4 (follow traffic rules) and 6 (follow recommended driving practice), though specifications which contribute to other criteria are also possible.

A few features of the CCDS are worth highlighting:

- Unlike the safety responsibility specification, it is not vital that all vehicles implement it identically. This means that it should be easier to modify to meet changing requirements over time.
- It does not distinguish between items which are backed by specific legislation and those which are advisory. The distinction between the two is not helpful for implementing an automated test, which simply reports whether a specification is met without judgement on the circumstances. As discussed in section 3, we assume that a separate subjective judgement will be make on whether non-conformance was acceptable for each set of test results. The level of evidence required to justify non-compliance may vary between different parts of the specification.

### 4.3.2 Example of test application

In the example scenario, we have implemented a check for whether the Ego vehicle exceeds the speed limit at any point.

Had the Ego been the non-priority vehicle at the junction, a test could have also checked that it complied with the requirement to give priority to vehicles on the main road. Current UK legislation (UK Statutory Instrument 2016 No. 362) states that a give way line must not be crossed in a way which is "likely... to cause the driver of another vehicle to change its speed or course in order to avoid an accident". A CCDS item for based on this law (in addition to the safety responsibility specification test from section 4.2) would allow the detection of situations where the ADS inconveniences another road user but no collision results. For example, it could define a maximum delay which the vehicle under test may impose on users of the main road.

### 4.3.3 General implementation challenges

Challenges associated with implementing CCDS rules include:

- Road traffic legislation and the Highway Code were written with human users in mind. Further interpretation will be required to create formal specifications for use in automated testing. For example, the legislation on give way lines (quoted above) does not define the meaning of 'likely' or 'cause'. It also sets no limit on the period for which it applies. For example, a strict interpretation could prevent a slower vehicle from ever joining a high-speed road (as if it joins,

it is likely that another vehicle will eventually catch up and be required to alter its speed or course).

- A CCDS implies a top-down approach to defining the rules which make up good driving. Unless the CCDS is extremely well designed (to a level which may not even be possible), some deployments could come across situations where one or more rules need to be contravened to ensure safe and functional operations.
    - Our assumption is that an ADS may need to break the rules in some circumstances. This would have to involve human judgement which could be based on risk assessments and justifications provided by the ADSE. There is a strong argument for requiring these to be provided before testing is started, to prevent retrospective justifications being created when an ADS does not behave as expected.
    - An alternative could be to use CCDS compliance as a metric with which to compare an ADS to a comparator system (e.g. a human driver). This avoids the need for an explicit definition of when non-compliance is tolerable, replacing it with a scoring system for rating its severity instead. However, this option would also make it harder to ensure that any non-compliance is a result of a careful decision on the part of the designers.

### 4.3.4 Speed limit and give-way specific implementation challenges

- Speed limit laws are relatively well defined, and some parts could be straightforwardly translated to a formal specification. However, there may still be edge cases where the speed limit to be enforced is ambiguous or following it precisely is undesirable.
- While we have not attempted to implement the 'give way' rule, significant assumptions would need to be made to translate this to a formal specification.

## 4.4 Reference model comparison tests

### 4.4.1 Summary of test purpose

The previous two types of test have been based on assessing the compliance of an ADS against a formal specification. As discussed in section 2, these will not be able to define all aspects of good driving for many years, if ever. Implementing the idea of an outcome score, to be compared between the ADS under test and some reference system, allows the coverage of automated testing methods to be extended. This could be used to support assessment of requirements based on driving criterion 2 (avoid harm, even if not the cause).

Comparisons to a reference model can be thought of as implementing the conditions or limitations aspects of a requirement (see section 2). For example:

- Comparing performance to a human driver on a per-scenario basis represents a condition: it identifies those events where an ADS creates a more severe outcome than the human.
- Requiring an ADS to have a lower overall rate of harm than a human driver would represent a limitation: it indirectly defines the maximum rate at which an outcome may occur.

### 4.4.2 Example test application

A comparison test was used to compare the occurrence of collisions between the vehicle under test and a reference human driver. Concrete scenarios where there were collisions involving the vehicle under test were allocated a score based on collision severity. These scenarios were then repeated but with the reference human controlling the vehicle under test, rather than an ADS.

For demonstration purposes, a simplistic method of estimating collision severity was used. The score was calculated as follows:

- A harm score for the first collision was assigned based on the change of velocity of the vehicle under test and (if present) other actor in the first collision involving the Ego. Ideally, this harm score would be based on empirical data, considering actor type (e.g. pedestrian, car) and point of collision. For simplicity, the demonstration implementation instead assigned a score based on the square of the magnitude of velocity change.
- After the first collision, the behaviour of vehicles in the scenario was not always realistic. Instead of evaluating the harm from any secondary collisions directly, a score was assigned based on the velocity immediately after the first collision (a measure of potential for further harm). This (arbitrarily) uses a score equivalent to each vehicle involved in the first collision having a second collision with a fixed object at half of the velocity it had immediately after the first.
- Primary and secondary collision harm scores for all vehicles were added together.

These scores were then compared to a human driver in the same concrete scenario, in this case one of the project team interacting directly with a desktop simulator. The framework allows for multiple participants to drive the vehicle in the same scenario, to allow for comparisons between the ADS and best, worst or average participant. A realistic implementation would need careful experimental design (e.g. ensuring that the simulation is immersive, drivers are appropriately selected and do not know the content of the scenario in advance).

### 4.4.3 General implementation challenges

Unlike the previous two tests, this example requires a comparison to be made between different systems. This comparison introduces new challenges. Some of the most important are:

- Not all variation in outcomes will be a result of intentional features of the test. Scenarios are typically designed to ensure that every vehicle driving them experiences a similar level of challenge. However, this cannot be perfectly true: the precise traffic situations experienced as part of a scenario vary pseudo-randomly as a function of Ego behaviour. For example, the length of time which a vehicle waits before moving off at the start of a scenario could alter whether it will meet oncoming traffic at a junction later. If per-scenario tests are used on a large data set, it is likely that eventually the ADS will be 'unlucky' and experience a worse outcome than the reference model for reasons unrelated to the quality of its driving.
- Use of a reference model comparison could require the ADS to approximate the behaviour of the reference model too closely in order to achieve acceptable scores. This is a problem if the reference model does something which is generally undesirable but happened to work well in a particular concrete scenario.
    - In the example scenario, the Challenger vehicle starts to move out of the junction when Ego reaches a particular (variable) point on the road. For concrete scenarios where this point is close to the junction, a faster-moving Ego vehicle may have passed the junction before the conflict occurs. This means that there are some scenarios where driving inappropriately fast could give a better score. If the human driver drives in this way, any ADS which does not would be labelled as non-compliant for that scenario.
    - One way to overcome this could be to make a statistical comparison of performance across a range of scenarios, though this raises its own issues. Some of the issues associated with statistical measures were discussed in our previous paper (Myers & Saigol, 2020).

### 4.4.4 Human VS ADS Crash severity score implementation challenges

Accounting for secondary collisions was the main challenge experienced when implementing the collision severity score method. Scoring for these will need to be thought through carefully. If the intention is to measure their severity directly (i.e. to do calculations based on all collisions the simulator reports, not just the first) then scenarios need to include realistic post-collision behaviour for all actors and the fidelity of a simulator's collision physics model becomes critical. The current approach of using residual velocity is an imperfect compromise: it will tend to overestimate the outcome severity in cases where the vehicle does not lose control (e.g. a glancing blow at high speed followed by regaining control would appear as a relatively high severity collision).

# 5 Software framework design

Examples of each test type (as described in section 4) were implemented in a software framework. The priority was to demonstrate the aspects of testing which were most likely to improve our understanding of how to design test processes. Some of the information gained from this has already been included in the implementation challenges subsections of the previous chapter. This prioritisation does mean that some aspects of a real framework have not been demonstrated. For example, we decided not to attempt to integrate a realistic ADS, instead defining the test vehicle's behaviour as part of the scenario. Implementing features like these well would have required substantial technical resources while generating little new understanding.

## 5.1 Overview of requirements

The aim was to build a test management framework with the following core functionality:

- Query and download scenarios from the MUSICC database developed as part of a previous project (Connected Places Catapult, n.d.)
- Run these in the CARLA simulator (Dosovitskiy, Codevilla, López, & Koltun, 2017) and record the raw-data results in an exchangeable format
- Apply the example tests to these results
- Combine the results of tests into an overall evaluation result
- Present these results to the user

In building this system, we aimed to make the process as repeatable and auditable as possible. For example, all results are referenced to a specific version of the framework, including test scripts. While the demonstrator system is integrated with MUSICC and CARLA, it is not tied to them. It would be reasonably straightforward to substitute these for different tools in the future, provided that the same standards are used for input and output from each.

## 5.2 Scenario assessment process

Figure 3 below shows a high-level diagram of the scenario assessment process. The key points are sumarised below:

- The process starts with the input of a scenario database query string, which can either be stored in a text file or entered directly into the framework interface.
- This is used to request and download all scenarios corresponding to that query string using the MUSICC API. Note that MUSICC can store logical scenarios (where some values are

defined by a statistical distribution rather than a single value). By default, the software will request 5 concrete scenarios for each logical.

- These scenarios are run in a version of CARLA which has been modified to save raw data results in a 'trace' file.
- Test scripts are executed based on the contents of the trace file and scenario files (e.g. the scripts may also take into account road layout information defined in OpenDRIVE).
- For scenarios where a collision occurred, the human comparison test script will initially return an inconclusive result. The test manager script can be run again with a different setting and will test the human reference driver in the logical scenarios where the ADS crashed.
- Once any human comparison trials are complete, a report is created.

Figure 4 shows the output in the test manager user interface for selecting scenarios and initialising CARLA. Figure 5 shows the same scenario running twice in CARLA: the initial run with an ADS in control (and a collision occuring) and the same concrete scenario being driven by a human.

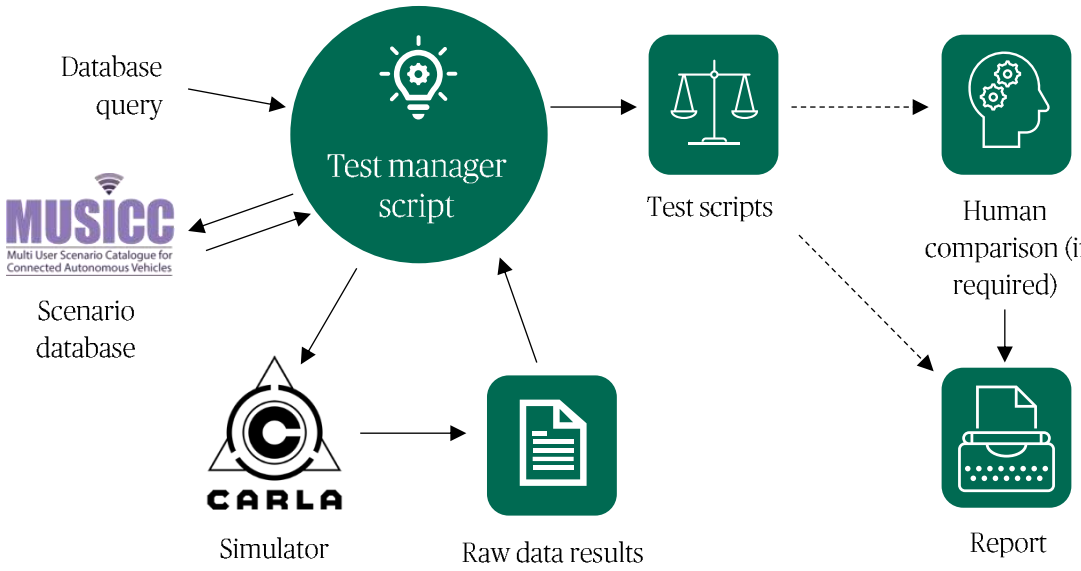



Figure 3: High level diagram of scenario assessment process

Extracting results from CARLA in an exchangeable format was a key early decision in the framework design. This is intended to allow test execution to be separated from test evaluation. For example, there are possible use cases where an independent assessor could request that an ADSE runs scenarios on their own system and provides a raw data output for further analysis. This raw data format is discussed further in section 5.3. To facilitate this type of use case, the test manager script can be set to stop as soon as the results file is complete, or start from the results file and continue with the rest of the test.

All test scripts are intended to be applied to all scenarios. If the test is not relevant to the scenario, the test script may pass trivially. For example, the pullout test described in section 4.2.2 will always return a 'pass' score if the scenario contains no priority junctions.

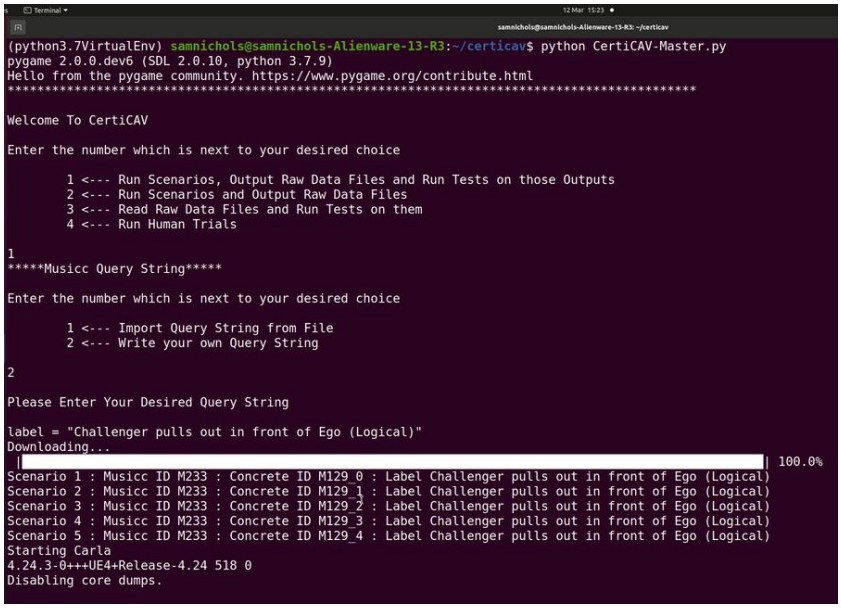



Figure 4: CertiCAV interface showing selection of operation type, query string Import and CARLA initialisation

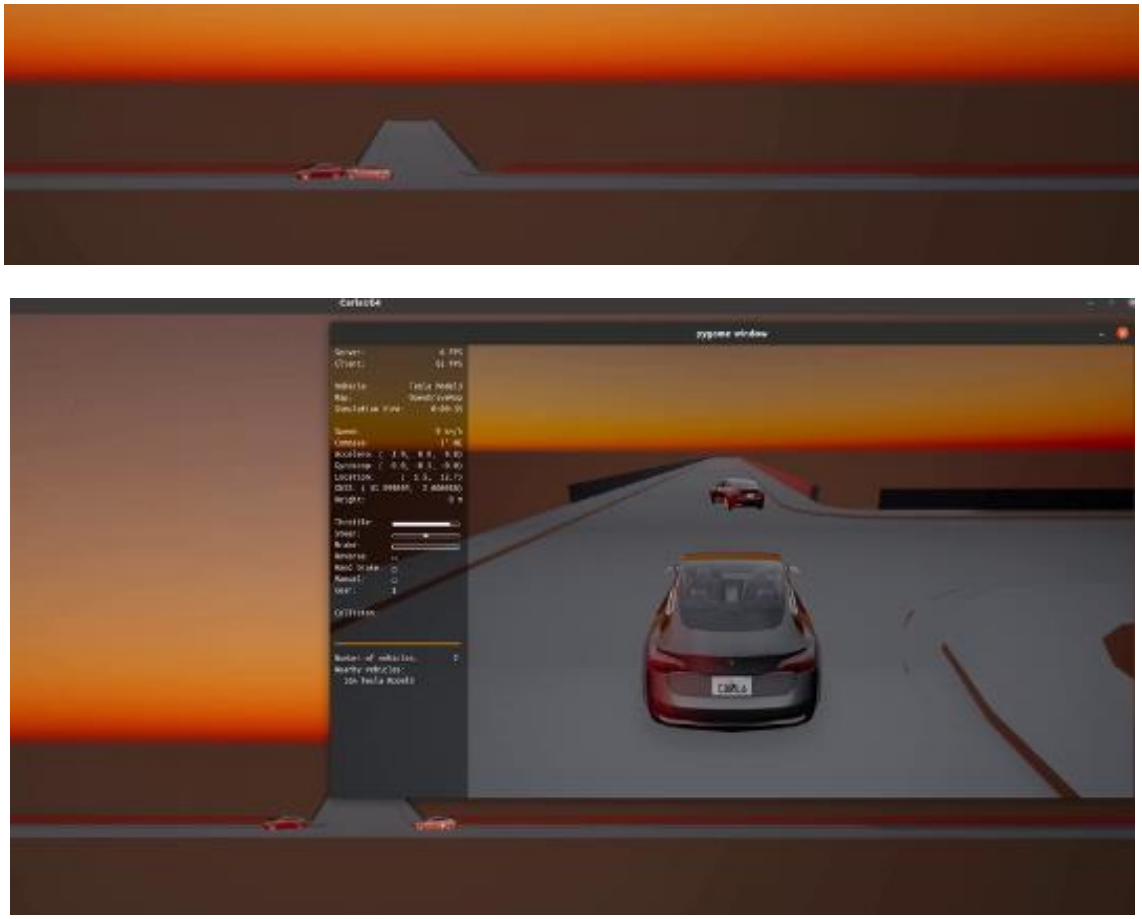


Figure 5: CARLA running initial ADS test (top) and human comparison test in same scenario (bottom)

## 5.3 Intermediate trace format

The intermediate trace format is used to store raw data output from the simulator. When combined with the information in the scenario file and road network, it should contain all information required by the test scripts. The intermediate trace format used for the CertiCAV demonstration is internally known as CertiTRACE: this is a use-case specific format largely based on the ASAM OpenSimulationInterface (OSI) standard.

It was sometimes slightly difficult to decide whether a variable should be recorded directly by the simulator or calculated from raw data as part of the test scripts. The following factors were considered in decision making:

- Factors which support including variable in trace format:
    - It is simple to extract from the simulator but hard to calculate in test scripts. For example, a variable which requires information which has already been determined by simulator but would require reconstruction of the road network and vehicle positions to implement as part of the test scripts.
    - Moving the calculation outside the simulator would result in a worthwhile improvement in transparency.
- Factors which discourage including variable in intermediate trace format:
    - Requires information which may not be readily available to the simulator
    - Straightforward to calculate from basic data (e.g. vehicle coordinates)
    - Data requirements are likely to vary depending on test script design
    - Calculating or storing it would excessively degrade simulator performance

Table 2 shows a list of items suggested for inclusion in the intermediate trace format, and which were implemented for demonstration purposes. This is based on an analysis of the example tests from section 4 and experience on previous projects. Future projects may identify more variables which should be added to this list.

Most of the items identified in Table 2 can be represented by variables available in ASAM OSI (ASAM, n.d.). CertiTRACE uses these variables but adds an explicit collisions record and some administrative data (e.g. labels to identify and group scenarios).

Table 2: Suggested variables for inclusion in trace format

| Item | Type | Reason for inclusion | Implemented in demonstration? |
|---|---|---|---|
| Weather (precipitation, atmospheric and surface conditions) | Time series | Could affect reasonable expectations of other road user's behaviour<br>Could affect Ego vehicles own reasonable expectations e.g. of surface friction | No |
| Lighting conditions | Time series | Could affect reasonable expectations of other road user's behaviour | No |
| Actor type and dimensions | Static[4] | Affects likely level of harm in the event of a collision<br>Required to calculate the positions of vehicle edges and corners<br>Could affect reasonable expectations of other road user's behaviour | Yes |
| Actor mass | Static | Could affect likely harm in the event of a collision | Yes |
| Actor positions, velocities and accelerations | Time series | Basic data required for many foreseeable tests | Yes |
| Collisions | Time series | Required for collision severity comparison | Yes |
| Status of variable signs and traffic signals | Time series | Likely to be required for CCDS tests based on existing driving rules | No |
| Visibility matrix (record of which actors could reasonably be expected to 'see' each other) | Time series | Reasonable expectations of other road users' behaviour are likely to depend on what that vehicle can 'see'. As discussed in section 4.2.4, correct behaviour may also depend on what is causing the occlusion.<br>While it is possible to calculate this based on vehicle positions, road network files and scenery, this comes relatively close to rebuilding the simulator as part of the test scripts.<br>Our initial suggestion is to include it in the trace format, but further exploration may be worthwhile. | No |

## 5.4 Reporting and audit data

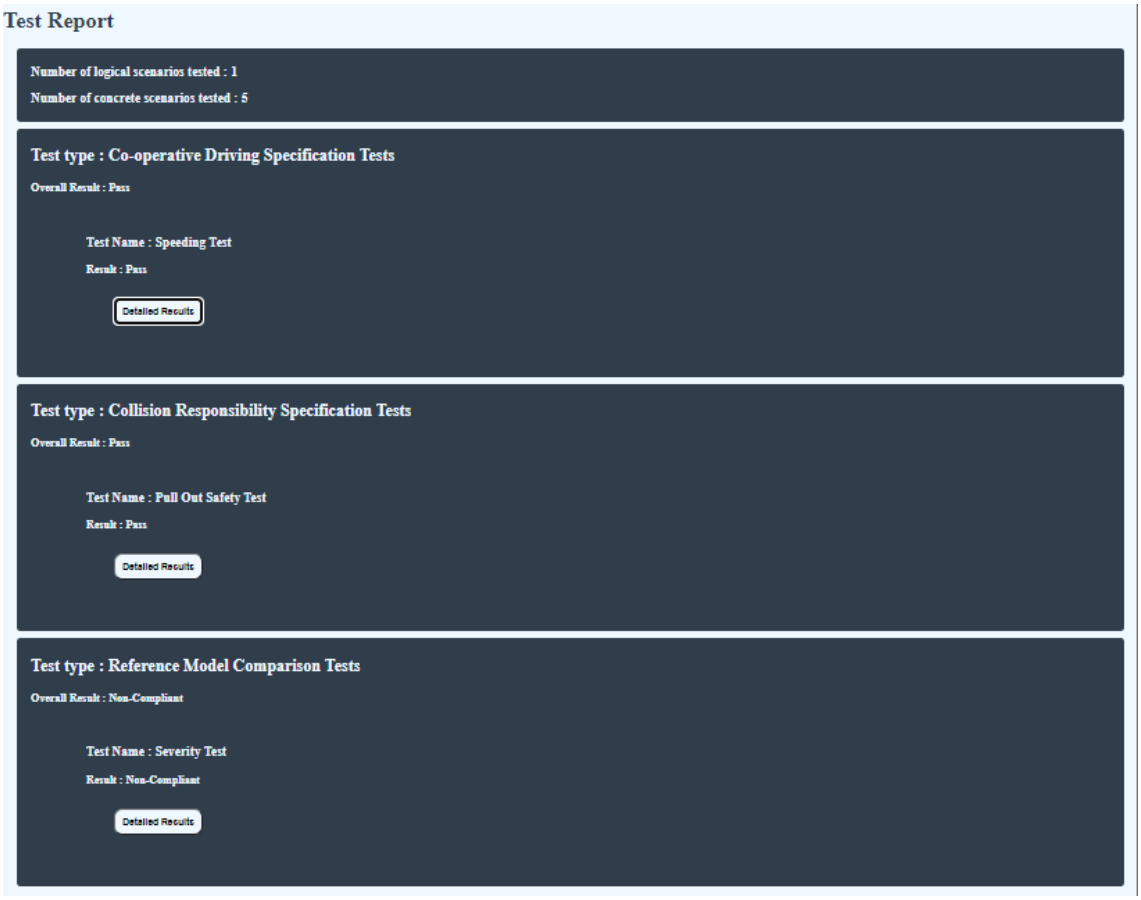



Figure 6: Example test report

The system produces a user-friendly HTML test report as shown in Figure 6 (with detailed results hidden). This shows which tests have been applied, the scenarios which they were applied to and the results from each. Section 6 explains the content of this report in more detail.

As well as a report, the system also stores the following raw data from each concrete scenario to ensure traceability:

- The CertiTRACE file containing raw data from the scenario execution. This also contains administrative data for audit purposes: a MUSICC query string, MUSICC query URL, and commit references for both CertiCAV and CARLA software.
- A video recording of the scenario being executed.

[4] This is not strictly true. Pedestrians and cyclists can vary their shape in many ways and vehicle doors can open. This appears to be difficult to represent in current formats.

# 6 Results for example scenario

This section contains an example report generated by the software framework. Ego's behaviour was evaluated according in the scenario first introduced in section 4.1. In this chapter, we explain the results shown.

## 6.1 Cooperative driving specification tests

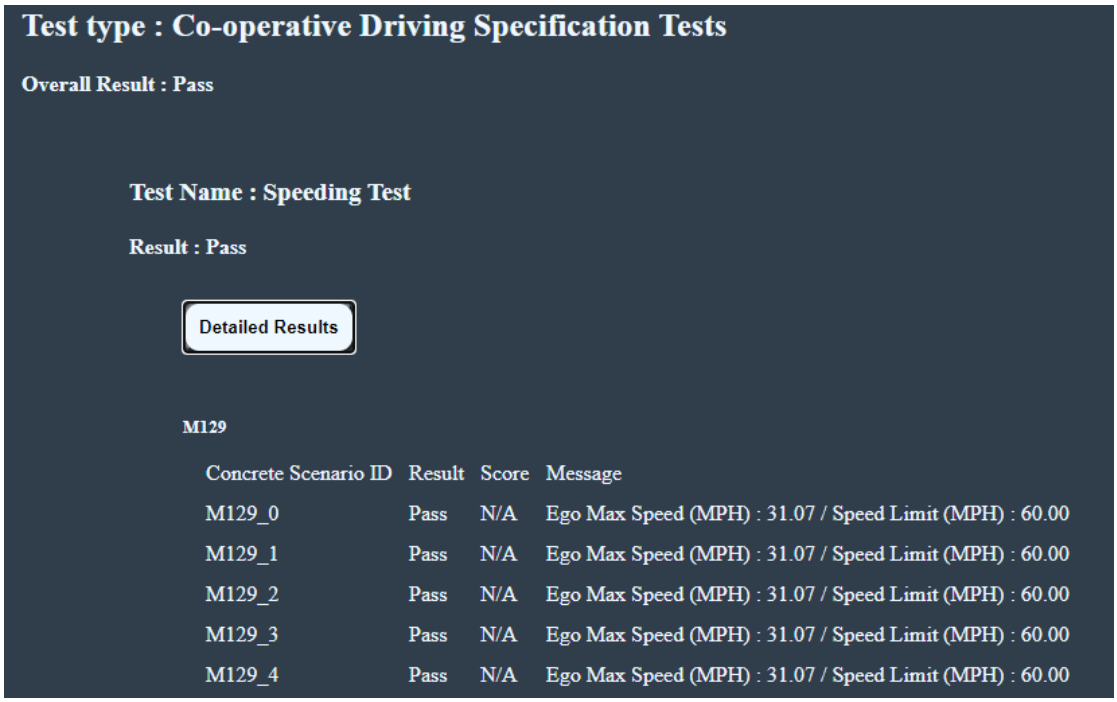



Figure 7: CCDS section of example test report

The example CCDS test was a check of speed limit compliance, with the test name "Speeding Test". As Figure 7 shows, the example 'ADS[5] drove at less than the speed limit in all scenarios, so the overall result is a pass.

[5] The demonstration used scenario-specific scripted vehicle behaviour rather than a true ADS

## 6.2 Safety responsibility specification tests

**Test type : Collision Responsibility Specification Tests**

**Overall Result : Pass**

**Test Name : Pull Out Safety Test**

**Result : Pass**

Detailed Results

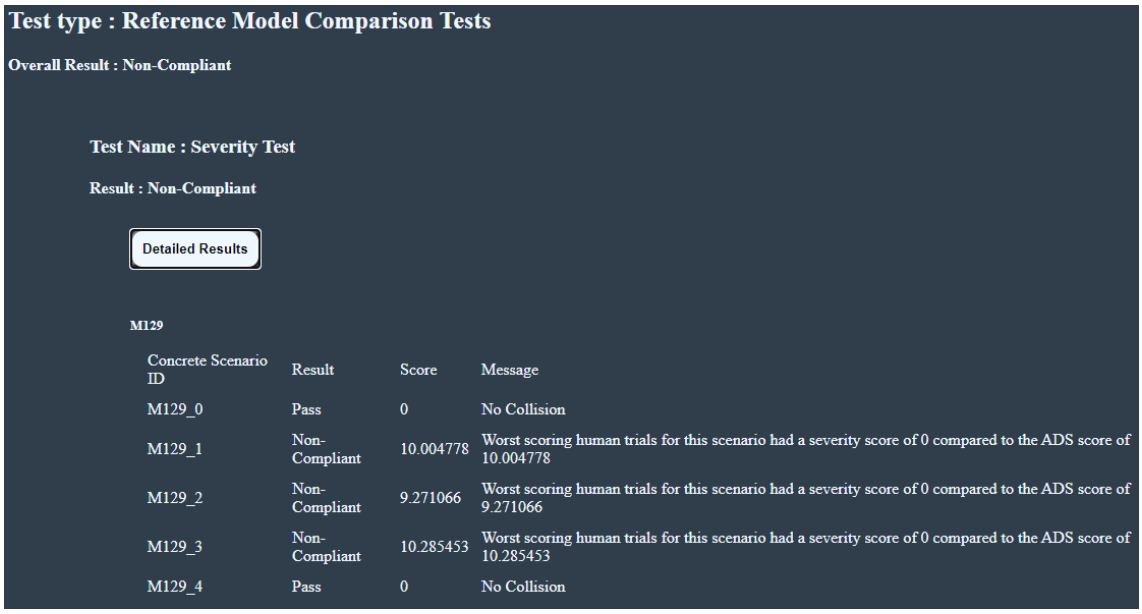
**M129**

| Concrete Scenario ID | Result | Score | Message |
|---|---|---|---|
| M129_0 | Pass | N/A | No Collision |
| M129_1 | Pass | N/A | At actor pullout, ego is 8.07m away from the conflict area when safe stopping distance is 63.60m |
| M129_2 | Pass | N/A | At actor pullout, ego is 3.40m away from the conflict area when safe stopping distance is 64.54m |
| M129_3 | Pass | N/A | At actor pullout, ego is 37.96m away from the conflict area when safe stopping distance is 63.75m |
| M129_4 | Pass | N/A | No Collision |

Figure 8: SRS section of example test report

The example safety responsibility specification test checked whether there were any collisions not explained by the unreasonable behaviour of another actor. In two concrete scenarios (M129_0 and M129_4) there were no collisions, so these trivially pass. In the other three, Challenger pulled out at a distance which the SRS defines as unreasonable. This means that, although there was a collision, Ego is not considered to have caused it for the purposes of this test. Therefore, all tests produce a pass result.

## 6.3 Reference model comparison tests



**Test type : Reference Model Comparison Tests**

**Overall Result : Non-Compliant**

**Test Name : Severity Test**

**Result : Non-Compliant**

Detailed Results

**M129**

| Concrete Scenario ID | Result | Score | Message |
|---|---|---|---|
| M129_0 | Pass | 0 | No Collision |
| M129_1 | Non-Compliant | 10.004778 | Worst scoring human trials for this scenario had a severity score of 0 compared to the ADS score of 10.004778 |
| M129_2 | Non-Compliant | 9.271066 | Worst scoring human trials for this scenario had a severity score of 0 compared to the ADS score of 9.271066 |
| M129_3 | Non-Compliant | 10.285453 | Worst scoring human trials for this scenario had a severity score of 0 compared to the ADS score of 10.285453 |
| M129_4 | Pass | 0 | No Collision |

Figure 9: Reference model comparison section of example test report

The example implemented compares the outcome severity between an ADS and a human driver. In all three scenarios where the ADS had a crash, all human drivers tested were able to avoid the crash. This is shown by the worst scoring human receiving a harm score of zero (no crash implies no harm) while the ADS received a positive score. A higher score for the ADS than the human shows that it has caused more harm and therefore the overall result is non-compliant.

# 7 Conclusions and next steps

This part of the CertiCAV project has demonstrated the principle of applying three types of test to automated vehicles as part of an automated framework. These cover tests based on formal specifications and tests based on making a comparison to a reference model. It is clear that tests based on formal specifications will be easier to apply, though tests based on reference models may be easier to define and can allow many more aspects of performance to be assessed. Key findings to consider when implementing tests include:

- Tests based on a collision responsibility specification could provide a robust way of ensuring that the ADS will only cause collisions at a very low rate, though the definition of 'cause' used may not exactly match human expectations. Using a formal specification for this can help to ensure that different automated vehicles do not violate each other's assumptions and therefore allows for a 'vision zero' approach to ADS vs ADS collisions. However, once this specification is defined, it may prove difficult to change, as all vehicles on the road need to use a compatible version to ensure safety.
- Having a separate specification (the common cooperative driving specification) for other formally defined aspects of good driving can make the approach more flexible, as compatibility between different ADS does not have to be considered to the same extent. This may also make it easier to create.

Any simple, per-scenario, comparisons with a reference driver or model are likely to require human interpretation. Care needs to be taken to avoid requiring an ADS to mimic any undesirable behaviours from the reference driver.

Topics which are likely to be of interest for future work include:

- Definition and validation of a Safety Responsibility Specification for a plausible deployment. While initial versions of this may have a limited scope, to be useful, they should give a near-complete definition of responsibility within that scope. For example, a specification could cover responsibility for all multi-vehicle collisions on motorways.
- Translation of a subset of current driving laws and advice into a Common Cooperative Driving Specification. This does not need to be complete or perfect to be useful, provided that ADS developers have an opportunity to justify any non-compliance.
- Expansion, refinement and standardisation of the scenario trace format to support the needs of the SRS and CCDS.
- Further research into methods for comparing the performance of an ADS with a human driver. A demonstration project using a fuller set of scenarios may help, as it would allow solutions to the theoretical issues identified in this report to be tested in practice.

# 8 References

ASAM. (n.d.). *ASAM OSI*. Retrieved from https://www.asam.net/standards/detail/osi/

Connected Places Catapult. (n.d.). *Multi User Scenario Catalogue for Connected and Autonomous Vehicles (MUSICC)*. Retrieved from https://cp.catapult.org.uk/project/multi-user-scenario-catalogue-for-connected-and-autonomous-vehicles/

Dosovitskiy, A., Codevilla, F., López, A., & Koltun, V. (2017). CARLA: An Open Urban Driving Simulator. *1st Conference on Robot Learning*. arXiv:1711.03938.

Myers, R., & Saigol, Z. (2020, May 26). *Pass-Fail Criteria for Scenario-Based Testing of Automated Driving Systems*. Retrieved from https://arxiv.org/abs/2005.09417

Saigol, Z., Peters, A., Barton, M., & Taylor, M. (2018, March). *Regulating and accelerating development of highly automated and autonomous vehicles through simulation and modelling*. Retrieved from http://www.zeynsaigol.com/TSC2018CAVSimulationTestingReport.pdf

Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2017). *On a Formal Model of Safe and Scalable Self-driving Cars*. Retrieved from https://arxiv.org/pdf/1708.06374.pdf

UK Statutory Instrument 2016 No. 362. (2016). *The Traffic Signs Regulations and General Directions 2016, Schedule 9*. Retrieved from https://www.legislation.gov.uk/uksi/2016/362/schedule/9/made

Robert Myers
robert.myers@cp.catapult.org.uk

Visit our website
cp.catapult.org.uk
Follow us on Twitter
@CPCatapult

Email us
info@cp.catapult.org.uk

CATAPULT
Connected Places