Connected Places Catapult

Specification for MUSICC Scenario Description Language

Version 1.0.0

April 2020

MUSICC
Multi User Scenario Catalogue for Connected Autonomous Vehicles

CATAPULT
Connected Places

Contents

Notice

CPC assumes no responsibility to any other party in respect of or arising out of or in connection with this document and/or its contents.

This document has 21 pages including the cover.

AUTHORISATION:

ACTION	SIGNATURE BLOCK	NAME AND POSITION WITHIN CPC
Written by:	Zeyn Saigol	Principal Technologist
Reviewed by:	Robert Myers	Technologist
Authorised by:	Richard Holland	Principal Technologist

RECORD OF CHANGES:

REALEASED TO	VERSION	REASON FOR CHANGE	DATE
MUSICC Industrial Advisory Group	0.0.1	Initial release	18/02/2019
MUSICC Industrial Advisory Group, Department for Transport	0.1.0	Added Description, ScenarioType, and traffic density fields to metadata	20/03/2019
MUSICC Industrial Advisory Group, MUSICC User Trials participants, Department for Transport	0.1.1	Separated out User Guide	30/04/2019
MUSICC Industrial Advisory Group, MUSICC User Trials participants, Department for Transport	0.1.3	Update to reflect project progress	11/11/2019
MUSICC Industrial Advisory Group, MUSICC User Trials participants, Department for Transport	0.1.5	Update to reflect SDL changes	20/03/2020
Public	1.0.0	Update to reflect SDL changes	02/04/2020

1. Introduction

Overview of project

The MUSICC (Multi User Scenario Catalogue for CAVs) project is developing systems to help with the challenge of ensuring that automated and highly-automated vehicles (HAVs) are safe for occupants and all other road users. The Transport Systems Catapult (TSC) is executing MUSICC on behalf of the UK's Department for Transport (DfT).

This HAV certification challenge requires coordination of validation activities between national regulators and system developers – with common (or at least aligned) approaches to validation desirable. In support of this, MUSICC is an initial implementation of a usable, living, HAV scenario catalogue. The catalogue is intended to store scenarios that specify the situations which highly-automated vehicles must handle safely to be suitable for release to market.

MUSICC's key differentiator from other similar projects is that it is focused on regulatory certification, and is led by a regulator (DfT) and an impartial, neutral mediator (CPC). The project is collaborating closely with stakeholders such as OEMs, ADS developers, organisations with expertise in HAV validation, and international regulators, to incorporate and align with ongoing and completed initiatives.

The focus of MUSICC is not on collecting scenarios, but on creating a solid foundation to enable the storage of future scenarios (however it has produced a minimum set of scenarios to prove the validity of the format and infrastructure). For a scenario repository to be adopted for production use by the HAV development community it requires all the following:

- a well-engineered scenario description language;
- a system that makes scenarios available for use, creation and update electronically; and
- use of a scenario format that is accepted by international regulatory bodies responsible for certification of HAVs.

Content of this document

This document describes the format, or Scenario Description Language (SDL), which is used to describe scenarios stored in MUSICC. We have made the decision to base the MUSICC SDL on OpenDRIVE[1] and OpenSCENARIO[2], with minor additions, as these are the most widely-used and open of the existing scenario formats. The additions that MUSICC makes to OpenSCENARIO/DRIVE are described in Section 4, and the conventions to be used when creating files for MUSICC in Section 6.

This document also contains a high-level introduction to how versions and revisions (i.e. versions of the scenario format) are managed in the MUSICC system (Section 5).

Instructions for searching and downloading scenarios are provided in a separate user guide document (`MUSICC_User_Guide.pdf`).

---

[1] Current version: http://opendrive.org/. Future: https://www.asam.net/standards/detail/opendrive/
[2] Current version: http://www.openscenario.org/index.html Future: https://www.asam.net/standards/detail/openscenario/

2. Requirements identification

MUSICC aims to facilitate easy interoperability between systems used by test experts, regulators and ADS Developers. As such, it makes sense for us to adopt any established industry best practice, including existing scenario description formats.

Input from the community has been sought through several routes:

- In September 2018, CPC hosted a workshop to explain the aims of the project to interested parties and identify high-level requirements for the system. A report from this workshop was provided to attendees (and is also available to users with access to the MUSICC SharePoint site).
- A symposium was held in June 2019 to demonstrate the system and gather input to address some of the key questions identified during development. A report from this event is available on the MUSICC webpage at https://ts.catapult.org.uk/innovation-centre/cav/cav-projects-at-the-tsc/musicc/.
- An industrial advisory group was formed. Throughout the project, members of this group have received updates on progress and have contributed their opinions on where development efforts should be focused.

3. Overview of MUSICC Representation

Functional, Logical and Concrete Scenarios

PEGASUS[3] and others have identified the need to represent scenarios at different levels of detail, and to use precise terms to describe the level of detail. We adopt their terms:

> **Functional scenario** – a textual description of the key elements of a scenario, for example “pass a broken-down car at the side of a single-carriageway road”
>
> **Logical scenario** – a more precise definition using a technical format, but leaving some aspects undefined. For example, a logical scenario might specify a range of allowed speeds for a traffic vehicle instead of an exact speed, or might leave the width of lanes unspecified.
>
> **Concrete scenario** – a fully-defined description of a scenario using a technical format. OpenSCENARIO/DRIVE is a format which allows concrete scenarios to be defined.

MUSICC stores logical scenarios. Compared to storing concrete scenarios, fewer scenarios need to be kept in the database, which has significant benefits for the curation of a regulatory scenario library. The element of randomisation in the tests also gives more confidence in the safety of systems that are certified using a process based on such a set of scenarios.

However, the system is effectively also able to store functional and concrete scenarios, while only providing a format for logical:

a) **Functional**: we store metadata for each scenario, which provides a high-level summary of the scenario, and is related to ODD specifications (see Section 4). Global tags can also be used to identify different logical scenarios with the same functional parent.
b) **Concrete**: MUSICC scenarios allow a range of possible values for scenario parameters to be supplied; however, if these ranges are not supplied, a MUSICC scenario becomes a fully-specified concrete scenario.

[3] https://www.pegasusprojekt.de/files/tmpl/PDF-Symposium/04_Scenario-Description.pdf

MUSICC scenario structure

MUSICC stores scenarios in three distinct records, corresponding to the MUSICC, OpenSCENARIO and OpenDRIVE parts of the scenario. Figure 1 shows the relationship between these records. This structure allows OpenSCENARIO and OpenDRIVE files to be re-used as part of several different MUSICC scenarios.

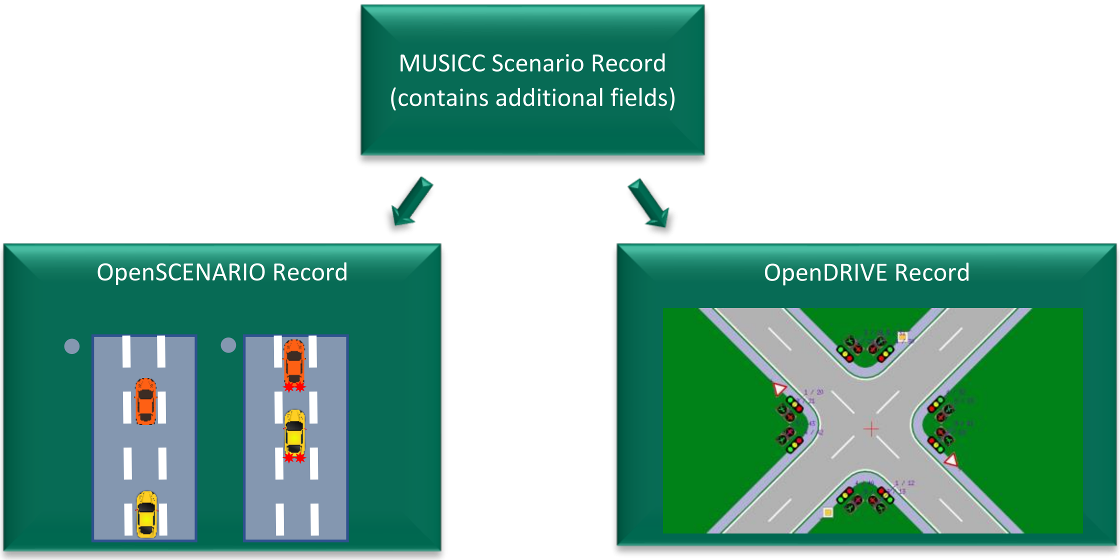

Figure 1: Relationship of MUSICC, OpenSCENARIO and OpenDRIVE records

Note we have deliberately broken the current 1:1 mapping from OpenSCENARIO to OpenDRIVE, which allows us to use the same OpenSCENARIO record with different OpenDRIVE records to capture, for example, the same basic manoeuvre with traffic signs and road standards used in different countries.

MUSICC also allows OpenSCENARIO catalogs and other resources to be stored alongside the records shown in Figure 1. These are manged independently.

MUSICC does not have specific functionality to support the use or storage of OpenCRG files. This is motivated by a strong desire to reduce complexity, and the highly detailed description of road surface provided by OpenCRG is only rarely likely to be relevant to the high-level HAV test scenarios we expect to be stored. However, if required, an OpenCRG file may be stored as an additional resource.

4. Additions to OpenSCENARIO

This section describes the data stored in a MUSICC scenario in addition to that stored as part of OpenSCENARIO and OpenDRIVE. An example XML file conforming to the MUSICC SDL is shown in Appendix A.

Metadata and scenario queries

For users of the scenario database, it is important to be able to query for the subset of scenarios which are applicable, given the capabilities of the system-under-test (SUT) – in other words, its ODD. For example, an ADS may be designed to only work on highways, in which case it shouldn't be tested against urban scenarios. Further, if ADS developers also wish to use MUSICC for internal testing, it is likely to be important to be able to query for an even smaller subset matching a specific feature or functionality.

MUSICC achieves these query capabilities through the use of metadata tags, which describe high-level properties of each scenario. These tags are conceptually separated into 4 groups, which are described in the following subsections.

General properties

- Description: free-text description of the scenario
- CountryCode
- DriveOnRightOrLeft
- UseCase: Highway, Urban, Inter-Urban or Parking
- ScenarioType: Logical or Concrete
- Exposure, which represents how frequently this scenario is expected to occur
- Test case weighting, to be used as part of automated evaluation processes
- SituationDemand, which is an indication of the complexity of a scenario, and therefore the demand it places on the driver (whether ADS or human; although SituationDemand may be slightly different between the two)
- CollisionCategory: NormalDriving, Collision or NearCollision
- InitialSpeedLimit, the speed limit at the start location for the ego vehicle (noting the speed limit may change as the scenario unfolds): SlowUrban, Urban, FastUrban, TrunkRoad, or Highway – see Appendix 0 for details.
- RealWorldMap, true if the OpenDRIVE file corresponds to a real-world location
- RealWorldCoordinate, latitude and longitude for the origin of the OpenDRIVE map
- RepresentsADASTest, true if this scenario is a tightly-scripted ADAS test procedure (c.f. NCAP testing)
- ADASFeaturesTested, a list of the ADAS feature(s) being tested (if RepresentsADASTest is true)
- SceneGraphModelDetail, which provides metadata for scenery files.
- SceneGraphModelSensorRealism, which allows labelling with intended use cases of the scenery files (testing of camera, LIDAR and RADAR perception).

For speed limits (and traffic density and average speed, defined in the following section), we have collected values into named bins, rather than using the raw numerical value (see Appendix 0 for the specific bins suggested). This ensures clarity on membership of scenario sets, makes it easier for end-users to select scenarios for a use case, and also avoids ambiguities with speed limits in mph in some territories and kmh in others.

Core ODD specification

Note: we have recently reviewed the design considerations for ODD definition languages. An informal document summarising our thoughts has been shared with the IAG.

The core ODD is summarised in seven variables:

- Features of the road (e.g. divided carriageway, 2-lanes in each direction)
- Environmental conditions (weather, time-of-day)
- Intended ego vehicle category (e.g. M1 - passenger car)
- Primary ego vehicle maneuver (e.g. turn right)
- Any other key actors (e.g. a pedestrian or an emergency vehicle)
- The action performed by the other actor (e.g. cross the road, emergency stop)
- Whether any of the key actors break rules, such as road traffic laws or advice to drivers
- The traffic density and average speed, for the busiest section of road the ego vehicle must navigate during the scenario – see Appendix 0 for details

The intention is to keep these as simple and generic as possible, while still allowing queries that correspond at a high level to an ODD (Appendix A shows all values permitted in the current SDL draft for these variables). Multiple values can be specified for each of the variables to increase the expressive power.

Note that, for the road features in particular but also for others, the intention is *not* to simply record every feature present in the scenario's OpenSCENARIO/DRIVE record. While it may be possible in the future to automate the extraction of such information, it will not capture which of these features are important to the scenario, and which are incidental background. For example, a scenario may incorporate a bicycle, but the key aspect is to test the SUT when a car in front performs an unexpected emergency stop. In that case, the KeyActorTypes field would be expected to include PassengerCar, but not PedalCycle.

Scenario admin data

- SourceOrganization, the organisation that contributed the scenario
- OwningOrganization, the organisation that owns the IP of the scenario
- Information from the FileHeader of the MUSICC record:
    - label, a short text description
    - version, a simple integer. Label+version+revision form a unique key for a MUSICC scenario
    - updateDateTime
    - updateUsername
    - revision, the MUSICC SDL revision that the scenario conforms to
- Global tags, which allow arbitrary labels (e.g. test set identity) to be assigned to scenarios
- Regulation tags, which will label any scenarios referred to by a regulation.

Custom data

A CustomMetadata element allows arbitrary name-value pairs to be included in the metadata. The hope is that, if certain custom metadata fields are found frequently in incoming scenarios, they can be migrated to a properly validated schema element in a future revision of the MUSICC SDL.

Ego vehicle objective

OpenSCENARIO does not currently provide a way to specify which of the entities in the scenario represents the system(s) under test. Also, while the expectation is that the SUT will autonomously decide its path and route, it may need to have a target destination provided. With a view to generality, the MUSICC SDL allows for any object (or objects) in the scenario to be specified as an entity under test. These may optionally be assigned a goal position. These can be defined using World, Road or Lane elements in a format which closely follows OpenSCENARIO v0.9.1 (see Appendix A for an example).

Parameterisation

Parameter stochastics means the ability to specify a range of allowed values for a given parameter, as opposed to a fixed value (in other words, to represent logical rather than concrete scenarios). This has several important benefits as noted previously, including making it harder to develop systems which will only pass a small set of fixed certification tests.

We provide two options when exporting scenarios from MUSICC, either to export in the MUSICC native format (which provides parameter stochastics), or export concrete scenarios with randomisation pre-applied.

PEGASUS use very similar approach to MUSICC, in that they store OpenSCENARIO files with additional data to specify the parameter stochastics. OpenSCENARIO/DRIVE provides the ability to define parameters at the top of the XML file, in order to avoid having to specify the same value multiple times within the file. The PEGASUS XML schema neatly leverages this mechanism by allowing ranges and distributions for OpenSCENARIO/DRIVE parameters to be defined. Therefore we have aligned the MUSICC format with the PEGASUS one, achieving partial compatibility.

The options for the distribution of a parameter are:

- Uniform distribution of values within a range
- Normal (i.e. Gaussian) distribution with an expected value and variance
- A set of discrete choices (supporting for example enumeration string types such as “Rain”/“Dry”/“Snow”)

While PEGASUS include the ability to specify dependencies between parameters (i.e. cross-correlations between the random variables), we feel this is a non-essential feature for MUSICC’s use case, so have not included it.

3D model specification

Referencing

The workshop clearly indicated the need for a well-defined method of including 3D models within a scenario, allowing visual details to be provided if desired, and making use of a standard model file format. This be achieved by referencing OpenSceneGraph objects within OpenSCENARIO.

Scenery, if provided, should be provided by referencing an OpenSceneGraph model within the `SceneGraph` element of OpenSCENARIO. This is demonstrated in the example below:

```
<RoadNetwork>
    <Logics filepath="../xodr/example.xodr"/>
    <SceneGraph filepath="../models/example.osgb"/>
```

```
</RoadNetwork>
```

Similarly, any models for actors should be referenced using a `file` element within the object definition, as demonstrated below:

```
<Object name="Ego">
    <Vehicle name="AlfaRomeo_Brera_10_BiancoSpino" category="car">
        <!-- various elements -->
            <File filepath="../models/car_blue.osgb"/>
```

All file paths should be specified relative to the file which contains them.

File types

The supported formats for 3D model files are .osgb (binary) and .osgt (ascii). For user convenience, other formats may be provided in the resources folder: these do not form part of the core MUSICC specification. 3D models used in MUSICC should be built so that they are the correct size for the scenario, with no scaling required.

5. Management of versions and revisions

Versions and revisions

MUSICC has two key requirements:

- The ability to quickly adopt new scenario formats, in response to external developments and to user feedback.
- Support regulatory use, where traceability and the preservation of previous standards are important.

Given these, we have decided to explicitly manage version numbers for both the scenarios stored in MUSICC, and the revision of the format these scenarios are defined using. A revision number for MUSICC corresponds to a fixed revision of OpenSCENARIO and of OpenDRIVE.

A *conceptual scenario* (for example, a specific motorway cut-in) will have a unique label (as described under “Scenario admin data” in Section 4 - Additions to OpenSCENARIO) – for example the label might be `M40_J12_Cut_In`. A conceptual scenario can exist in several different revisions and versions, where a new version implies a fix or minor enhancement to the scenario. Figure 2 Illustrates this pictorially.

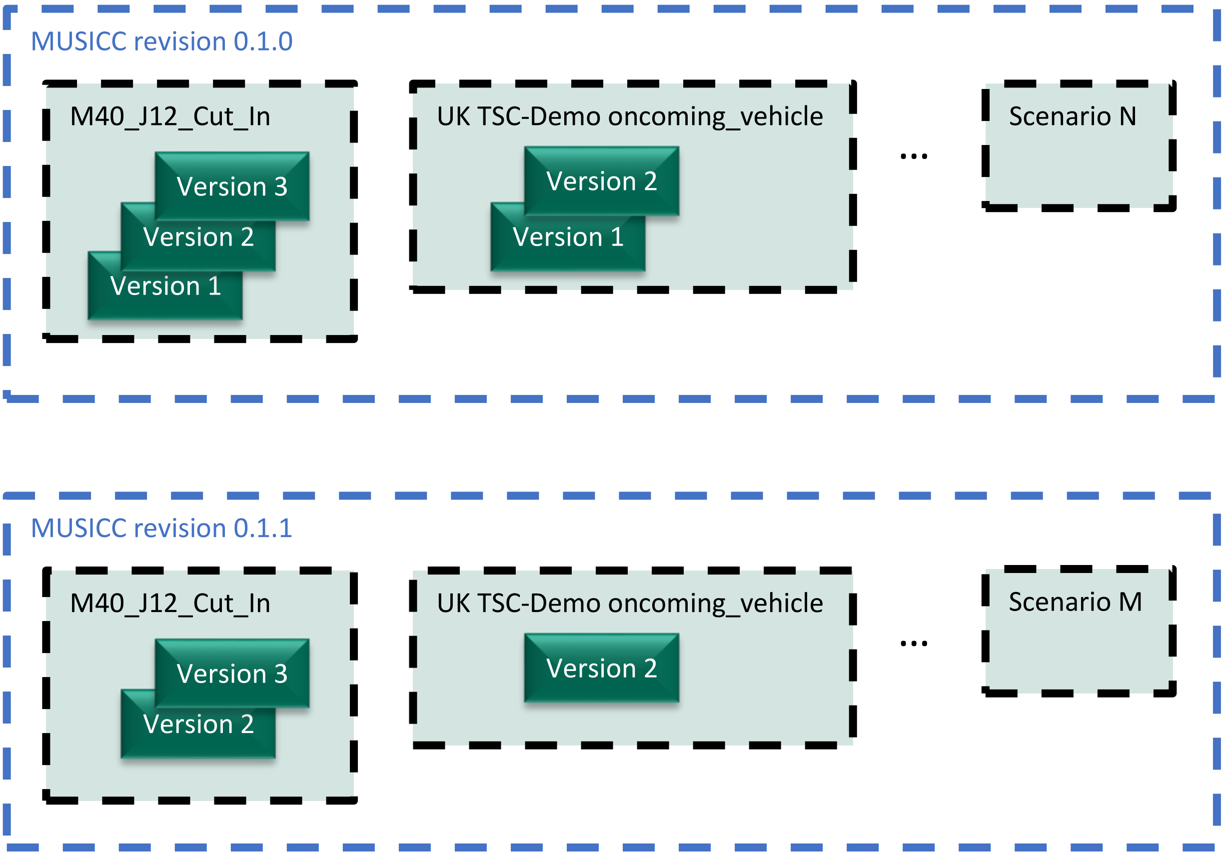

Figure 2: Relationship of revisions, conceptual scenarios (as defined by a label), and versions

Record IDs and unique keys

To help support references between records, an efficient API, and easy user-management of MUSICC files, each record has a unique ID number (which is that item's primary key in the MUSICC database). A MUSICC record ID corresponds to fixed IDs for its child OpenSCENARIO and OpenDRIVE record.

A MUSICC record also has a unique key, consisting of the revision number, label, and version number.

Revision Management

We are aware it may be useful to support multiple OpenSCENARIO and OpenDRIVE revisions, to try to align with the toolchain compatibility of different users of MUSICC. Here, “support” implies maintaining a set of scenarios at that revision, ensuring that scenarios are updated, added and removed as necessary.

We will actively maintain a maximum of 3 MUSICC revisions, which means at most 3 OpenSCENARIO and/or 3 OpenDRIVE revisions.

We currently support the two MUSICC revisions shown in the table below:

MUSICC revision	OpenSCENARIO revision	OpenDRIVE revision	MUSICC revision release date
0.1.5	0.9.1	1.4H	20/03/2020
1.0.0	1.0	1.4H	01/04/2020

6. Conventions and additional requirements

While ASAM OpenDRIVE and OpenSCENARIO have their own detailed format specifications, there are some additional requirements or conventions which should be followed when uploading scenarios to MUSICC. These are documented here.

Upload naming and structure

Uploads to MUSICC should take the form of a single ZIP file, with an internal structure as shown in Figure 3 below. Downloads will also be provided in this structure.

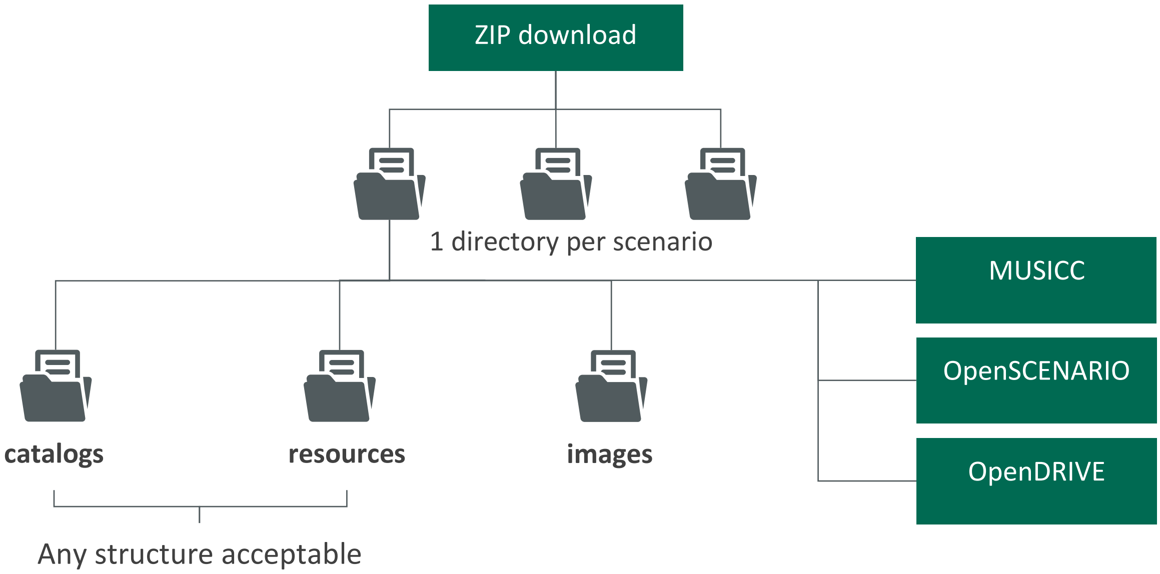

Figure 3: Structure required for MUSICC uploads

The MUSICC, OpenSCENARIO and OpenDRIVE files are required to be present: catalogs, resources and images are optional. To correctly identify and label files, MUSICC requires that:

- The MUSICC file contains a reference to the relevant OpenSCENARIO and OpenDRIVE files
- The OpenSCENARIO file contains a reference to the relevant OpenDRIVE file and any required catalog directories.
- Any files which are identical to each other have the same name, and any files which have the same name are identical.
- Images to be displayed in the user interface are stored in the 'images' folder. They should have a filename in the format NAME_n, where NAME is the same as the MUSICC file's name, and n is an integer. Images will be displayed in alphabetical order. A list of image formats supported is available at https://docs.python.org/3.6/library/imghdr.html.

The resources folder may be used to store any additional files relevant to the scenario, such as 3D models.

OpenSCENARIO

Sources of existing documentation:

- XSD and MindMap for OpenSCENARIO
- VIRES User Guide (on their wiki)
- High-level presentations on OpenSCENARIO

ASAM will release improved documentation as part of the OpenSCENARIO 1.0 “Transfer Project”.

Catalogs

MUSICC requires OSCCatalogs are used according to the following rules:

1. Within a scenario file, all catalog paths specified under the top-level <Catalogs> element must be directories, not files

```
<Catalogs>
    <VehicleCatalog>
        <Directory path="this/must/be/a/directory/"/>
    </VehicleCatalog>
```

2. All files with a ".xosc" extension within the directories specified will be read in, and a catalog namespace will be created for each file (using the name attribute from the <Catalog> element)

```
<Catalog name="MyVehicleCatalog">
```

3. When a catalog is referenced, the catalogName must correspond to the name attribute from one of the catalog files in one of the directories.

```
<CatalogReference catalogName="MyVechicleCatalog"
entryName="AudiA3_blue_147kW"/>
```

4. The type of the catalog specified by the Catalog sub-element in the scenario file (e.g. `<VehicleCatalog>`, `<DriverCatalog>`, `<TrajectoryCatalog>`) is ignored semantically.

5. The name of the catalog .xosc file is ignored semantically (e.g. it is incorrect to refer to a catalog “AudiCatalog” just because the file name is “AudiCatalog.xosc”).

End conditions

EndConditions are optional in the OpenSCENARIO specification. We recommend that all scenarios should contain end conditions, to facilitate use with automated testing tools.

Actor behaviour

When designing a scenario, the author should make sure that actor vehicle movements are defined in a way which accommodates varying ego vehicle behaviour. For example, a vehicle should not be able to avoid all conflicts at a slip road by stopping for a short time.

A Example MUSICC XML File

```xml
<?xml version="1.0" encoding="utf-8"?>

<MUSICCScenario
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="MUSICC_1.0.0.xsd">

    <FileHeader label="UK CPC-Demo highway_test_double_lane_change"
                version="3"
                updateDateTime="2019-07-26T10:00:00"
                updateUsername="Zeyn Saigol"
                revision="1.0.0" />

    <Metadata>
        <Description value="Busy highway cut-in with ego in middle lane,
         braking vehicle ahead in outside lane, and inside lane blocked by an
         HGV" />
        <CountryCode value="GB"/>
        <DriveOnRightOrLeft value="Left"/>
        <UseCase value="Highway"/>
        <ScenarioType value="Logical"/>  <!-- Alternative is Concrete, which
         implies no stochastics -->

        <Exposure value="E2"/>
        <TestcaseWeighting value = "1.2"/> <!-- to be used in scenario
         selection - this scenario should be selected 20% more often than a
         scenario with a weight of 1.0-->
        <SituationDemand value="High"/>
        <CollisionCategory value="NormalDriving"/> <!-- Alternatives are
         Collision or NearCollision -->
        <InitialSpeedLimit value="Urban"/>

        <RealWorldMap value="true" mapDataValidDateTime="2017-07-
23T11:03:00Z"/>
        <RealWorldCoordinate latitude="52.038386" longitude="-0.767261"/>

        <RepresentsADASTest value="true"/>
        <ADASFeaturesTested>
            <ADASFeature value="AEB"/>
            <!-- Full list given in XSD -->
        </ADASFeaturesTested>

        <SceneGraphModelDetail value="HighPoly"/> <!-- definitions for these
         values are provided in schema -->

        <SceneGraphModelSensorRealism>
            <TargetSensor value = "CameraRealistic"/> <!-- definitions for
             these values are provided in schema -->
            <TargetSensor value = "LidarRealistic"/>
            <TargetSensor value = "RadarRealistic"/>
        </SceneGraphModelSensorRealism>

        <RoadFeatures>
            <!-- Full list given here for review -->
            <RoadFeature value="Roundabout" />
            <RoadFeature value="HighwayEntranceRamp" /> <!-- AKA slip road,
             on-ramp -->
            <RoadFeature value="HighwayExitRamp" /> <!-- AKA slip road, off-
             ramp -->
            <RoadFeature value="TrafficLightControlledJunction" />
```

```xml
    <RoadFeature value="3LegJunction" /> <!-- Often a T-Junction -->
    <RoadFeature value="4LegJunction" />
    <RoadFeature value="MoreThan4LegJunction" />
    <RoadFeature value="RailwayCrossing" />
    <RoadFeature value="PedestrianCrossing" />
    <RoadFeature value="StopSign" />
    <RoadFeature value="GiveWaySign" />  <!-- AKA yield sign -->
    <RoadFeature value="1Lane" />  <!-- AKA single-track road -->
    <RoadFeature value="2Lane" />
    <RoadFeature value="3Lane" />
    <RoadFeature value="4Lane" />
    <RoadFeature value="MoreThan4Lanes" />
    <RoadFeature value="DividedCarriageway" />  <!-- Implies a
     physical median, AKA a central reservation -->
    <RoadFeature value="OneWayRoad" />
    <RoadFeature value="BicycleLane" />
    <RoadFeature value="HardShoulderPresent" />
    <RoadFeature value="HighGradient" />
    <RoadFeature value="SharpBend" />
    <RoadFeature value="Roadworks" />
    <RoadFeature value="Tunnel" />
</RoadFeatures>

<EnvironmentalConditions>
    <!-- Full list given here for review -->
    <EnvironmentalCondition value="Rain" />
    <EnvironmentalCondition value="NoRainButRoadWet" />
    <EnvironmentalCondition value="Snow" />
    <EnvironmentalCondition value="Ice" />
    <EnvironmentalCondition value="Fog" />
    <EnvironmentalCondition value="Cloudy" />
    <!-- Time-of-day -->
    <EnvironmentalCondition value="Daylight" />
    <EnvironmentalCondition value="Night" />
    <EnvironmentalCondition value="Dawn" />
    <EnvironmentalCondition value="Dusk" />
</EnvironmentalConditions>

<IntendedEgoVehicleTypes>
    <!--Type codes from UNECE Consolidated Resolution on the
    Construction of Vehicles (R.E.3), Revision 6, plus a few MUSICC-
    specifc extensions-->
    <EgoVehicleType value="M1"/> <!--cars-->
    <EgoVehicleType value="N1"/> <!--vans and pickup trucks-->
</IntendedEgoVehicleTypes>

<EgoManeuverTypes>
    <!-- Full list given here for review -->
    <EgoManeuverType value="RightTurn" />
    <EgoManeuverType value="LeftTurn" />
    <EgoManeuverType value="GoStraightAtJunction" />
    <EgoManeuverType value="CrossTrafficTurn" /> <!-- Left turn in
     countries that drive on the right -->
    <EgoManeuverType value="WithTrafficTurn" /> <!-- Right turn in
     countries that drive on the right -->
    <EgoManeuverType value="UTurn" />
    <EgoManeuverType value="TurnOntoMinorRoad" />
    <EgoManeuverType value="TurnOntoMajorRoad" />
    <EgoManeuverType value="HighwayMerge" />
    <EgoManeuverType value="HighwayExit" />
</EgoManeuverTypes>

<KeyActorTypes>
```

```xml
        <!-- Full list given here for review -->
        <KeyActorType value="PassengerCar" />
        <KeyActorType value="Truck" />
        <KeyActorType value="Bus" />
        <KeyActorType value="Motorcycle" />
        <KeyActorType value="SmallLowSpeedVehicle" />
        <KeyActorType value="EmergencyVehicle" />
        <KeyActorType value="PedalCycle" />
        <KeyActorType value="Trailer" />
        <KeyActorType value="MotorizedOffRoadVehicle" />
        <KeyActorType value="AnimalDrawnVehicle" />
        <KeyActorType value="RailedVehicle" />
        <KeyActorType value="RailRoadVehicle" />
        <KeyActorType value="AnimalRider" />
        <KeyActorType value="Pedestrian" />
        <KeyActorType value="TrafficControlPerson" />
        <!-- Extra categories not part of WISE Drive -->
        <KeyActorType value="Animal" />
        <KeyActorType value="InanimateObstacle" />
    </KeyActorTypes>

    <KeyActorActions>
        <!-- Full list given here for review -->
        <KeyActorAction value="ActorCrossingRoad" />
        <KeyActorAction value="ActorStoppedInRoad" />
        <KeyActorAction value="EmergencyStop" />
        <KeyActorAction value="CutIn" />
        <KeyActorAction value="Overtake" />
        <KeyActorAction value="EmergencyStop" />
        <KeyActorAction value="WrongWayTravel" />
    </KeyActorActions>

    <KeyActorRulesBroken>
        <!-- current list given here for review -->
        <RuleSet value="RSSReasonableBehaviour"/>
        <RuleSet value="NationalAdvice"/> <!-- e.g. highway code for a UK
         scenario-->
        <RuleSet value="NationalLaws"/> <!-- e.g. exceeding the speed
         limit-->
    </KeyActorRulesBroken>

    <TrafficDensity value="FreeFlow"/>
    <TrafficAverageSpeed value="UrbanFlowing"/>

        <GlobalTags>
              <GlobalTag value="UNECE-Test-Set"></GlobalTag>
        </GlobalTags>

    <Regulations>
        <Regulation value="UNECE-GRVA-Res2135"/>
        <Regulation value="UK-RTA-2019-345"/>
    </Regulations>

    <CustomMetadata>
        <CustomTag name="EnvironmentalCondition" value="Hail" />
    </CustomMetadata>

    <SourceOrganization value="Connected Places Catapult"/>
    <OwningOrganization value="Connected Places Catapult"/>
</Metadata>

<EntityUnderTest> <!-- more than one may be specified-->
    <EntityRef entityRef="Ego"/>
```

```xml
        <GoalPosition>
            <WorldPosition x="0" y="10" z="9.5" h="0" p="0" r="0"/> <!-- Must
             be a location which exists within OpenDRIVE file -->
        </GoalPosition>
    </EntityUnderTest>

    <ParameterStochastics> <!-- Based on the PEGASUS schema for logical
     scenarios -->
        <RoadNetwork>
            <ParameterGroup name="">
                <Parameter name="$someODRparameter" unit="metres" type="int">
                    <Distribution>
                        <NormalDistribution expectedValue="33.4" variance="10"
                         />
                    </Distribution>
                </Parameter>
            </ParameterGroup>
        </RoadNetwork>

        <OpenScenario>
            <ParameterGroup name="">
                <Parameter name="$someOSCparameter" unit="metres" type="int">
                    <Distribution>
                        <NormalDistribution expectedValue="33.4" variance="10"
                         />
                    </Distribution>
                </Parameter>

                <Parameter name="$otherOSCparameter" unit="weather-unit"
                 type="string">
                    <Distribution>
                        <SingleValues>
                            <List>
                                <ListElement value="Rain" label=""
                                 probabilityOfOccurence="0.25" />
                                <ListElement value="Snow" label=""
                                 probabilityOfOccurence="0.05" />
                            </List>
                        </SingleValues>
                    </Distribution>
                </Parameter>
            </ParameterGroup>
        </OpenScenario>
    </ParameterStochastics>



    <OpenSCENARIO filepath="32564.xosc"/>
    <OpenDRIVE filepath="RQ31_130Limit.xodr"/>

</MUSICCScenario>
```

Definition of ranges for binned values

Speed limits

Enum Name	Range (kmh)	Range (mph)
SlowUrban	0 to 34.99	0 to 21.8
Urban	35 to 59.99	21.8 to 37.3
FastUrban	60 to 74.99	37.3 to 46.6
TrunkRoad	75 to 99.99	46.6 to 62.1
Highway	100+	62.1+

Traffic density

Density to expected-flow values mostly from **https://en.wikipedia.org/wiki/Traffic_flow#Overview**

Enum Name	Definition
EgoOnly	Only the ego vehicle
Low	One or two other vehicles in scenario
FreeFlow	2 to 11.99 vehicles/mile/lane
MaxStable	12 to 29.99 vehicles/mile/lane
Unstable	30 to 64.99 vehicles/mile/lane
StopStart	65 to 150 vehicles/mile/lane
Jam	150+ vehicles/mile/lane

Average traffic speed

Enum Name	Range (km/h)	Range (mph)
Stationary	Strictly zero	Strictly zero
Walking	0.001 to 6.99	0.001 to 4.4
FastTrafficJam	7 to 29.99	4.4 to 18.6
UrbanFlowing	30 to 59.99	18.6 to 37.3
TrunkFlowing	60 to 99.99	37.3 to 62.1+
HighwayFlowing	100+	62.1+

1 Sekforde Street
Clerkenwell
London
EC1R 0BE
Tel: 020 7952 5111

The Pinnacle
170 Midsummer Boulevard
Milton Keynes
MK9 1BP
Tel: 01908 359 999