OPENPASS
ARCHITECTURE COMMITTEE MEETING
14.12.2018
AGENDA

1. Goals of openPASS
2. Example Case AEB
3. Redesign Input Files
4. openScenario in openPASS
5. Modules and components
6. Documentation
7. Next Steps and Timeline
8. License
9. Handling of different Use-Cases
10. General Points for further development (Coding Rules, Documentation, …) (ITK)
11. Hierarchical SystemEditor, Documentation GUI, ComponentGenerator (VW)
GOALS OF OPENPASS
Standardized effectiveness assessment platform for driver assistance systems and autonomous driving

Usages of openPASS:
- Scenario-simulation with stochastic variation (e.g. NCAP)
- Closed loop traffic simulation (virtual FOT)
- Crash re-simulation (e.g. PCM)
Given is a road with 3 lanes
Scope: OpenDRIVE import, OSI ground truth generation

An ego car is placed at the center lane, surrounded by 4 scenario cars.
The rest of the scenery is filled with common cars.
Scope: OpenSCENARIO import, SpawnLogic, OSI world representation

At a given time, one of the scenario cars performs a sudden lane change, following a predefined trajectory.
Scope: TimeTriggerEventDetector, Manipulator, TrajectoryFollower

Several cars are equipped with a simplified automatic emergency brake (AEB), which takes the input of two geometric sensors. TBD! Exemplarisch nur Ego
Scope: Stochastic Distribution of Modules, AssistantSystems, GeometricSensors

Each AEB automatically merges the information from its sensors and starts braking in case of emergency.
Scope: AssistantSystem, SensorFusion, SignalPrioritizer

The simulation is repeated several times with stochastic variation to see the variability of the AEB's intervention.
Scope: Stochastic Placement of Agents, Output
REDESIGN OF INPUT FILES
EXPERIMENT SETUP
COMBINATION AND SCENARIO CONFIGURATION

- **Combination Config.xml**
- **Scenario.xosc**
- **Vehicle**
- **Sensors**
- **Vehicle Components**
- **Agents**
- **Driver**
- **Traffic**
- **Environment**
- **Scenarios**
- **Scenario (openScenario)**
- **Scenery (openDRIVE)**
- **SceneryConfiguration.xodr**
- **Vehicle ModelsCatalog.xosc**
- **Pedestrian ModelsCatalog.xosc**
- **External Catalogs**
static module and channel architecture for all agents
REDESIGN AGENTCONFIGURATION

AgentConfiguration renamed to AppConfig
Channel architecture is static
Additional information for used libraries (FrameworkModules)
No name attribute for channels
Better readability in case of complex channel structure

Slave configuration with additional framework modules

Static Components/channels for all agents (dynamically instantiated)
REDESIGN RUNCONFIGURATION REPLACED BY COMBINATION CONFIG AND SCENARIO

Renamed and restructured in CombinationConfig.xml

Relocated to VehicleModelsCatalog.xosc
REDESIGN RUNCONFIGURATION
REPLACED BY COMBINATION CONFIG AND SCENARIO

Reduced and relocated to CombinationConfig.xml
REDESIGN RUNCONFIGURATION
ADVANTAGES

✓ Usage of the openScenario standard
✓ Flexible experiment configuration using xml format
✓ Easy readability and editability of xml files
✓ Possibility to simply generate xml files automatically (e.g. GUI)
REDESIGN RUNCONFIGURATION
CONFIG-STRUKTUR

old structure

- Framework Configuration.xml
- Run Configuration.xml
- Scenery Configuration.xml
- Agent Configuration.xml

new structure

- Combination Config.xml
- Vehicle ModelsCatalog.xosc
- Scenario.xosc
- SceneryConfiguration.xodr
- AppConfig.xml

• Path and Logger Information
• Experiment Configuration
• Traffic Configuration (as SpawnPointParameters)
• Vehicle Model Parameters (as Agents)
• Observation Module Parametrization (OutputFile)
• Scenery in openDrive Format
• Module Connections for each Agent
• Channels
• Modules with Channel References for Inputs and Outputs
## REDESIGNED INPUT FILES

### SUMMARY

<table>
<thead>
<tr>
<th>Input Files</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SceneryConfiguration.xodr</td>
<td>New file ending</td>
</tr>
<tr>
<td>AppConfig.xml</td>
<td>Combines information from frameworkConfiguration.xml and agentConfiguration.xml</td>
</tr>
<tr>
<td>CombinationConfig.xml</td>
<td>Replaces runConfiguration.xml contains all information required for an experiment</td>
</tr>
<tr>
<td>Scenario.xocs</td>
<td>New file to specify a scenario (openScenario Standard)</td>
</tr>
<tr>
<td>VehicleModelsCatalog.xocs</td>
<td>New separate input file to specify vehicle parameters (from runConfiguration.xml)</td>
</tr>
<tr>
<td>Further Catalogs</td>
<td>Linked in Scenario.xosc (Pedestrians, etc.)</td>
</tr>
</tbody>
</table>
OPENSCEENARIO IN OPENPASS
Screenplay for the simulation

Description of initial setup

Link to Scenery.xodr and Catalogs (e.g. VehicleModelsCatalog.xosc)
- Spawn-conditions for Ego and Scenario-Agents: Position, Velocity
- Stochastic Variation of Position and Velocity (custom implementation!)

```xml
<Init> <!-- initial position and velocity of agents -->
  <Actions>
    <Private object="Ego">
      <Action>
        <Position>
          <Lane roadId="1" x="20.0" laneId="-1" offset="0.0"/> <!-- position -->
          <Stochastics value="2" stdDeviation="10" lowerBound="5" upperBound="100"/>
          <Stochastics value="offset" stdDeviation="0.3" lowerBound="-0.5" upperBound="0.5"/>
        </Position>
      </Action>
      <Action>
        <Longitudinal>
          <Speed>
            <Dynamics time="100" distance="1.0" rate="0.5" shape="linear"/>
          </Speed>
        </Longitudinal>
      </Action>
    </Private>
    <Private object="Scenario">
      <Action>
        <Position>
          <Lane roadId="1" x="0.0" laneId="-2" offset="-6.4566"/> <!-- position -->
        </Position>
      </Action>
    </Private>
  </Actions>
</Init>
```
- Definition of conditions and actions for the simulation
- E.g. Activation of an ADAS at a certain time during the simulation
- Usage of Detectors and Manipulators
- Standard maintained by ASAM e.V.
- Kick-off workshop in November
- Roadmap: https://www.asam.net/conferences-events/detail/kick-off-workshop-asam-openscenario/

- Proposal workshop to define specific requirements and project plan (January 2019)
- Open points:
  - How do we handle stochastics?
Proposal by BMW:

- openScenario Implementation (with custom extensions) will be committed by BMW
- Configs for openPASS will be designed to fit to the openScenario Standard as far as possible
  - Add file header attributes
  - Add catalogs for vehicles and pedestrians (move from CombinationConfig.xml)
  - Add scenery reference (move from CombinationConfig.xml)
  - Adjust storyboard to comply with the standard (add some tags, etc.)
- Some differences to standard will exist further
  - Stochastics are not available
  - Environment and Traffic profile remain in CombinationConfig.xml
- Incorporate newer versions of openScenario as they are released

Compliance with the Standard as target for the future
MODULES AND COMPONENTS
MODULES AND COMPONENTS
AGENT STRUCTURE
Screenplay according to example case with AEB:

Event triggered by TimeTriggerDetector

$t = 4.2\text{ s}$

TrajectoryFollower activated by Manipulator
– **AlgorithmLongitudinalDriver / AlgorithmLongitudinalVehicle (Components)**
  Updates the agents pedal positions and gear.

– **AlgorithmLateralDriverTasks / AlgorithmLateralVehicle (Components)**
  Updates the agents steering wheel position.

– **ActionSecondaryDriverTasks (Components)**
  Updates the agents BrakingLight, Indicator, Horn and all Lightswitches (Headlight, HighBeam, Flasher).

– **DynamicsRegularDriving (Components)**
  Updates the agents position

– **DynamicsCollision (Components)**
  Calculates the collision partners

– **DynamicsTrajectoryFollower (Components)**
  This module forces agents to drive according to a specific trajectory, defined in a separate file.
SignalPrioritizer (Components)
- Collects signals of same kind and relays the one with highest priority
- Example: Acceleration / Braking ADAS, Driver model

SensorRecordState (Components)
- This module forwards the agent parameters to the Observer in each time step.

SensorFusion (Components)
- The SensorFusion module allows unsorted aggregation of any data provided by sensors.

SensorObjectDetector (Components)
- SensorObjectDetector is the common base of all sensors.

AlgorithmAgentFollowingDriverModel (Components)
- Driver Model.
– **ParametersAgent (Components)**
  This component defines parameters for all other components of an agent. It includes constants and stochastic variables, which are divided into driver and vehicle parameters.

– **VehicleControlUnit (todo)**
  Stores the states and acceleration of vehicle components and Driver.

– **AgentUpdater (Components)**
  The AgentUpdater executes all Set-Methods of the agent dynamics after the DynamicsPrioritizer. This includes position, velocity, acceleration and rotation.

– **EventDetector, Manipulator (Core Modules)**

– **AEB (Components)**

---

All modules are documented with inline doxygen documentation.
NEXT STEPS AND TIMELINE
The future development should follow the process flow defined in 2018.

- Define Scope of the Release
- Create Epics and User Stories in Backlog
- Define Milestone / Minor Releases as necessary

**NEXT STEPS**

BMW will make a proposal in the next SC meeting
## MILESTONES

### Development Release 1.0
- Jan: Development (milestone Releases)
- Feb: Review Pull Request
- Mar: Pull Request
- Apr: Define scope of the next Release:
  - Define Epics and User Stories
  - Generate initial backlog
- May: OpenPASS
- Jun: Major Release 1.0
- Jul: Major Release 1.0
- Aug: Major Release 1.0
- Sep: Major Release 1.0
- Oct: Major Release 1.0
- Nov: Major Release 1.0
- Dec: Major Release 1.0

Note: The schedule is subject to change based on feedback and progress.
- OSI:
  Mozilla Public License
  - modified and copied source code must remain under the MPL
  - OSI is also planning to move to ASAM e.V.

- openScenario / openDrive:
  Association for Standardization of Automation and Measuring Systems - ASAM e.V.
  License Terms: https://www.asam.net/license/
FURTHER TOPICS
HANDLING OF DIFFERENT USE-CASES

Basic Use-Case

OSI Use-Case

PCM Use-Case

Building the openPASS tool for each Use-Case individually is not feasible

One Use-Case as goal for openPASS

- Merge to one Use-Case with Commit by BMW
- Integration to one openPASS Tool that handles all UseCases

openPASS