ASAM OpenSCENARIO V2.0.0

Public Release Candidate (PRC) kickoff webinar

Ben Engel
ASAM e.V.

17th January
14:00 – 17:00 CET

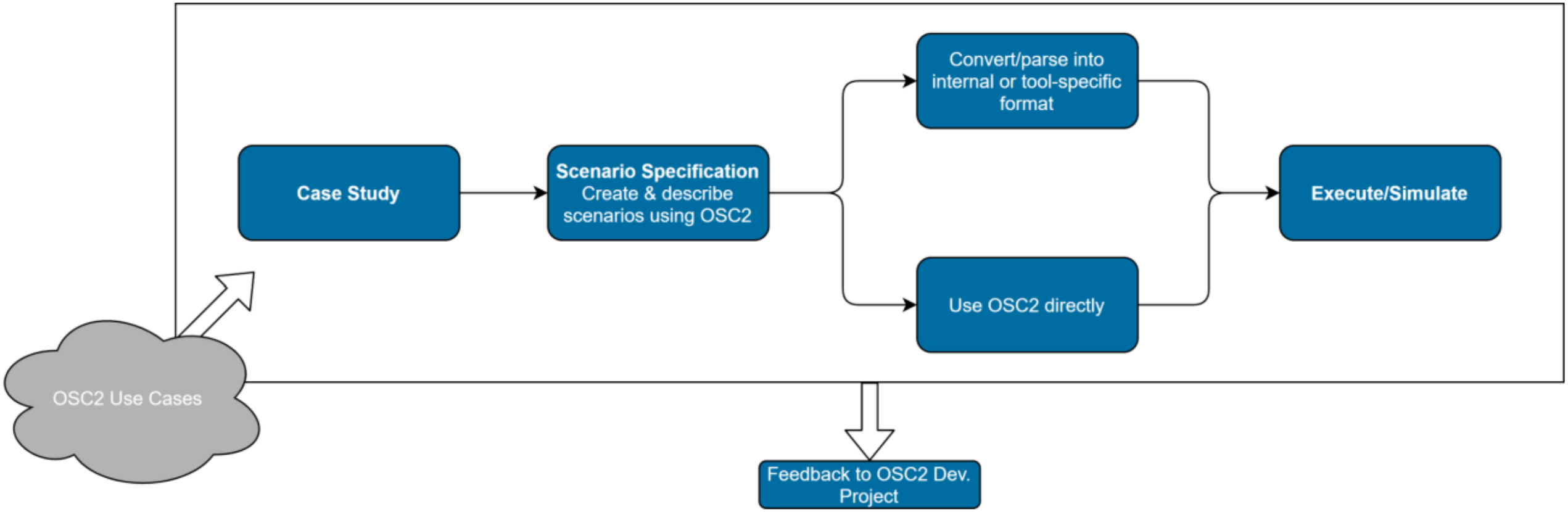
ASAM Association for Standardization of Automation and Measuring Systems

Agenda

Part 1: Intro and Overview	
14:00	Status of ASAM OpenSCENARIO V2.0.0 and the Public Release Candidate (PRC)
14:15	Overview of the standard
15:15	Submitting feedback on the standard
15:20	Mode of work for the PRC phase
15:45	**Coffee Break (10 mins.)**
Part 2: Getting started with the Standard	
15:55	Implementation Q&A “Ask us anything”/ “Beat the expert”
16:25	Introduction to writing scenarios
16:55	**Closing words**

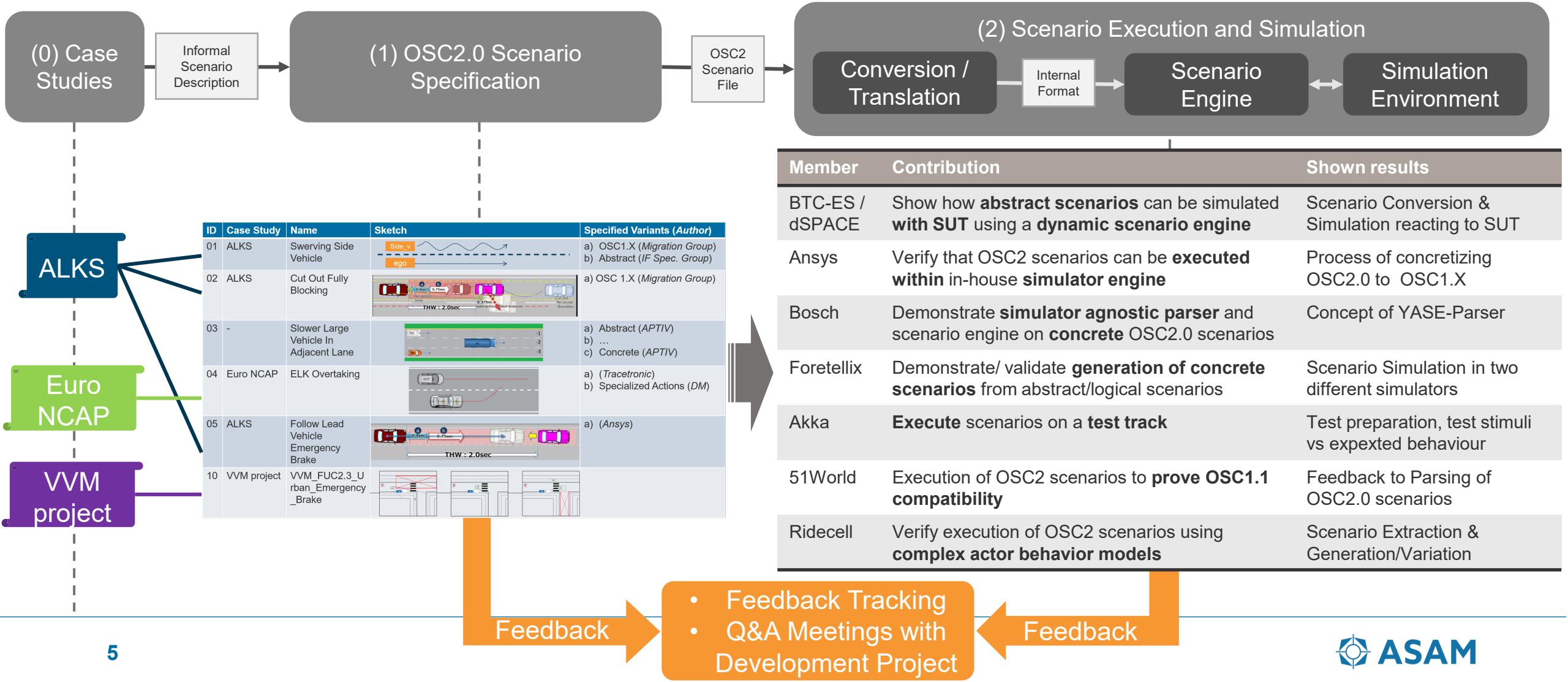
The working mode so far and looking ahead…

Approach so far...



Approach so far...

Case-study-driven early evaluation of language features regarding understandability, implementability and executability



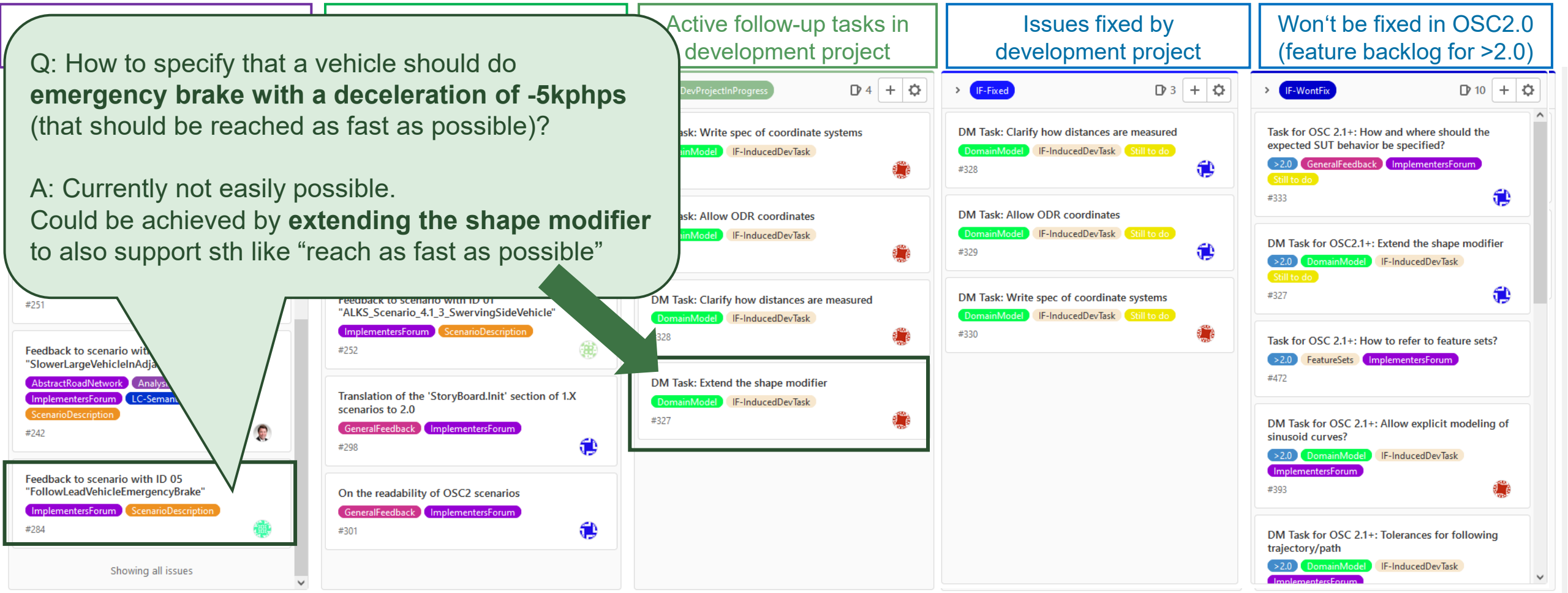
Member	Contribution	Shown results
BTC-ES / dSPACE	Show how **abstract scenarios** can be simulated **with SUT** using a **dynamic scenario engine**	Scenario Conversion & Simulation reacting to SUT
Ansys	Verify that OSC2 scenarios can be **executed within** in-house **simulator engine**	Process of concretizing OSC2.0 to OSC1.X
Bosch	Demonstrate **simulator agnostic parser** and scenario engine on **concrete** OSC2.0 scenarios	Concept of YASE-Parser
Foretellix	Demonstrate/ validate **generation of concrete scenarios** from abstract/logical scenarios	Scenario Simulation in two different simulators
Akka	**Execute** scenarios on a **test track**	Test preparation, test stimuli vs expexted behaviour
51World	Execution of OSC2 scenarios to **prove OSC1.1 compatibility**	Feedback to Parsing of OSC2.0 scenarios
Ridecell	Verify execution of OSC2 scenarios using **complex actor behavior models**	Scenario Extraction & Generation/Variation

Implementers Forum Feedback Tracking Board

Keep track of feedback, discuss with Development project, derive follow-up actions for major issues (either for OSC2.0 or 2.X)

Q: How to specify that a vehicle should do **emergency brake with a deceleration of -5kphps** (that should be reached as fast as possible)?

A: Currently not easily possible.
Could be achieved by **extending the shape modifier** to also support sth like “reach as fast as possible”



https://code.asam.net/simulation/standard/openscenario-2.0/-/boards/66

ASAM

Implementers Forum Summary and conclusions – till today.

- The Implementers Forum delivered **key feedback** that had major impact on developing the OSC2.0 standard e.g.
  - Identify gaps and missing parts in language features / domain model entities
  - Point out ambiguities, semantic unclarities and missing understandability or unintuitive usage
  - For major issues: Derived follow-up tasks to address the identified issues (see Feedback Tracking Board)

- Some **obstacles** due to the transient status of the evolving project
  - Grammar/syntax not complete and still evolving
  - Documentation not complete and fully fleshed out
  - Only possible because of very tight, interactive and constructive communication with Development Project e.g. via Q&A meetings

- With a **stable release candidate** many of these difficulties are removed
  - Parsers can be developed based on a feature-complete stable grammar that is not expected to have significant changes (except for bug-fixing etc…)
  - A full documentation is available describing all the language features and domain model entities
  - People that were hesitant and waiting for a stable version can now put their efforts into this Forum
  - Due to ‘freeze’ of development also Development project members can put their focus on Implementers Forum

Growing support for ASAM OpenSCENARIO V2.0.0…

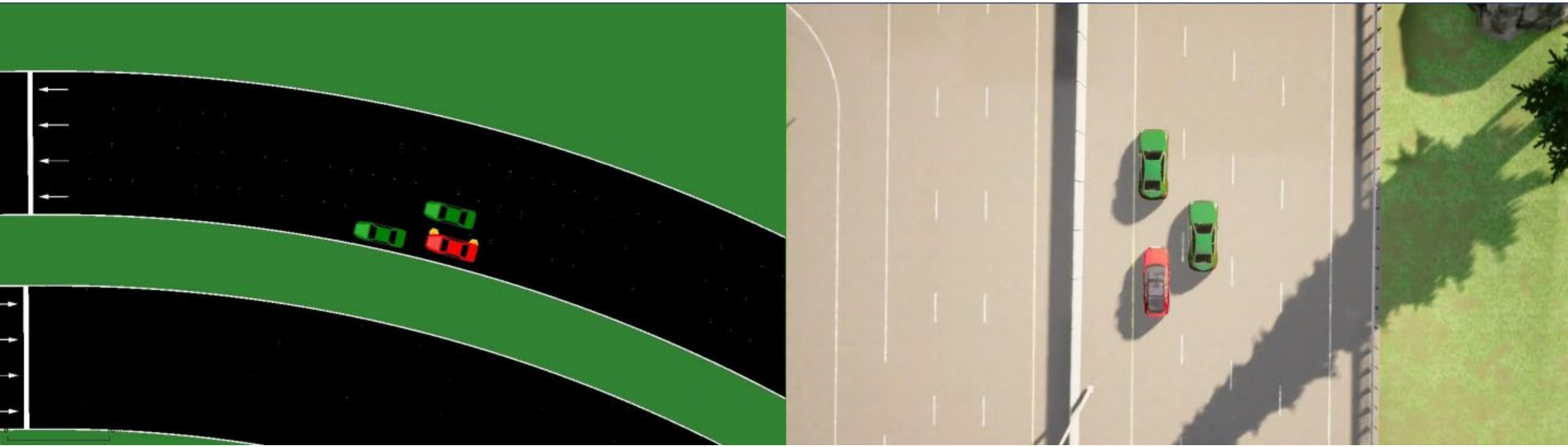
What resources are available so far?

- The standard documentation – here
- The full version of the EBNF (soon to be available on the PRC website, contact me to obtain it earlier!)
- A growing repository of scenario examples (disclaimer: not all are 100% correct syntax yet!)
  Join the implementers forum to access them before they are released publicly!
- Webinars (ASAM + non-ASAM) – reach out if you are planning any related events! – keep an eye on LinkedIn and the ASAM website!
- Growing list of tools and scripts…
  Goal: A public list of tools that support ASAM OpenSCENARIO V2.0.0 in some way!
  Developing tooling or scenarios that you want us to list? Get in touch with me!
  Both proprietary and open source solutions welcome!
- Some examples presented in the Implementers Forum so far…:
  - 51World (see the presentation at our China Regional Meeting)
  - BTC – Use of RTC for the simulation of abstract scenarios with SUT using a dynamic scenario engine

ASAM OpenSCENARIO V2.0.0 Implementers Forum Activities

ASAM

Parse OSC2 example and run in SUMO and/or Carla simulator



ALKS swerving_side_vehicle that was implemented by Scenario Specification Subgroup

Convert into Internal or Tool-Specific Format

RIDECELL

ASAM OpenSCENARIO V2.0.0 to Ridecell Scenario Language (RSL) conversion and feedback

- Simulator-agnostic
- Relative positions
- Parameter variations
- Actor behaviors

```
scenario:
  environment:
    time: <morning|afternoon|night>
    weather: <sun|cloud|fog|rain>
  road_network: <nodes, edges>
  actors:
  - name: ego
    type: vehicle
    start: <parent, s, l, heading>
    destination: <parent, s, l>
  - name: adversary
    type: vehicle
    start: <parent, s, l, heading>
    behaviors:
    - lane_follow:
        speed: Gaussian(20, 5)  # m/s
    - lane_change:
        target: <parent, s, l>
```

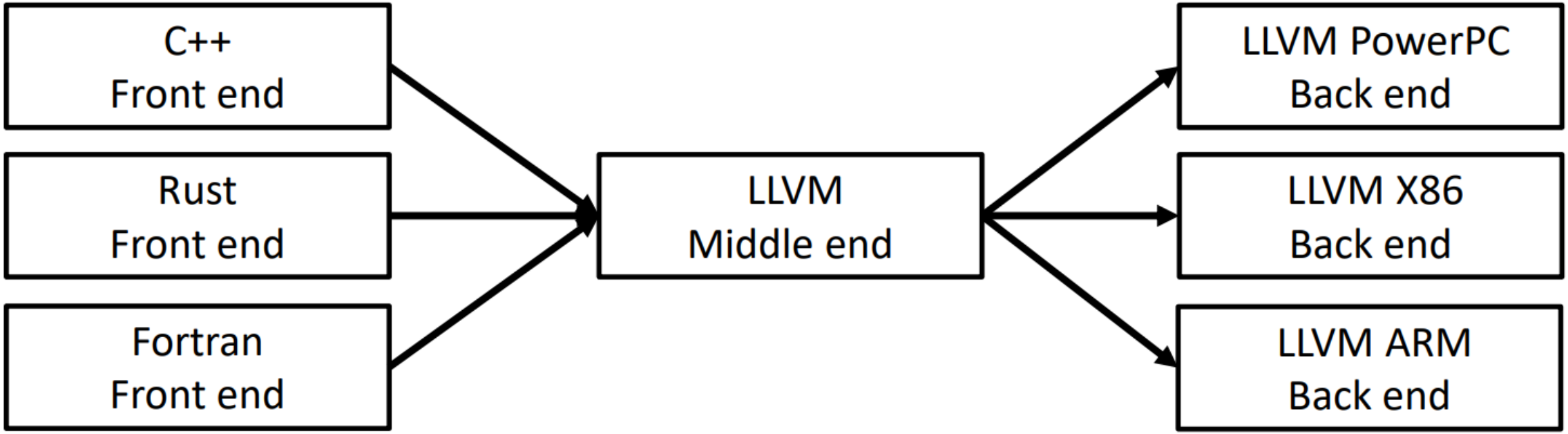
ASAM

Implementers forum activities ASAM OpenSCENARIO V2.0.0 Compiler Implementation

ASAM

Classical Compilers – Three stages

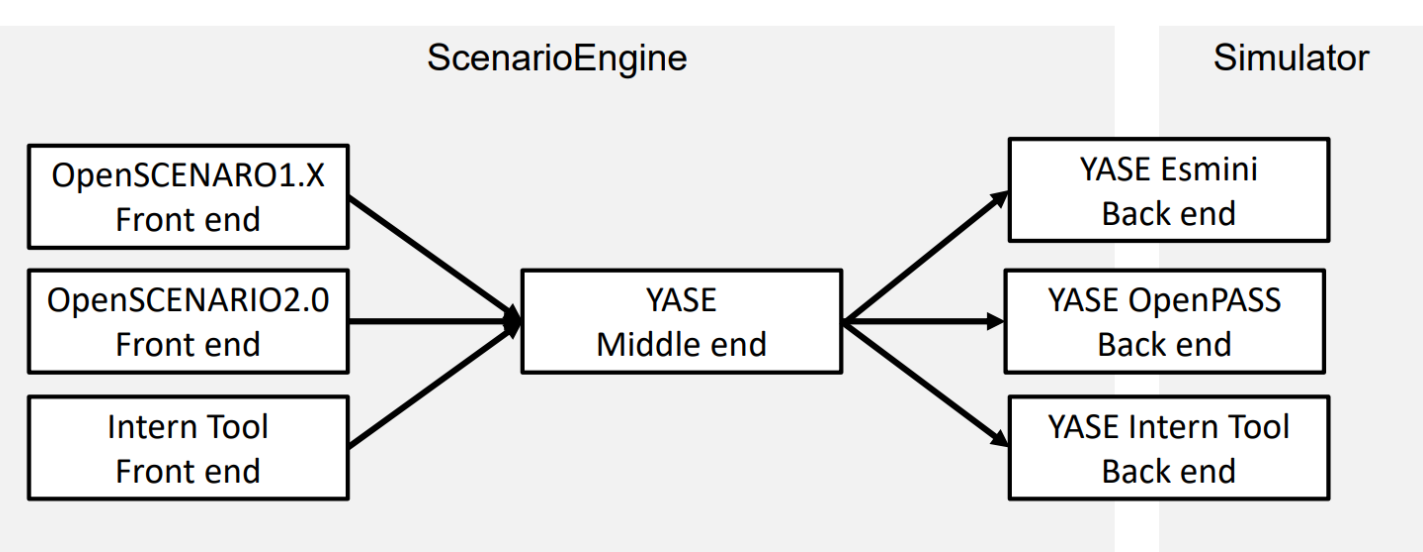
The modular approach brings flexibility:



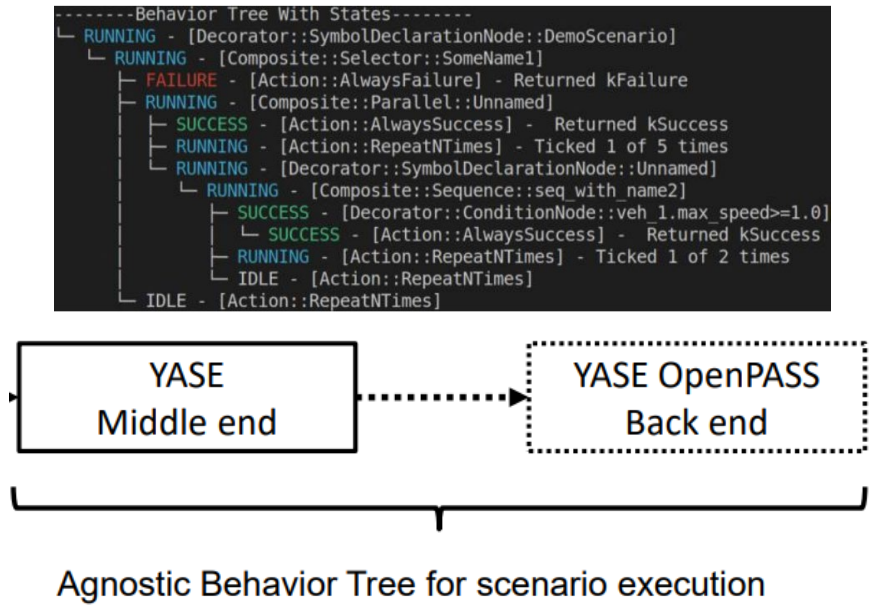
OpenSCENARIO2 requires a full compiler

YASE - Yet Agnostic Scenario Engine / Yet Another Scenario Engine

Three phases for a scenario compiler



Agnostic Behavior Tree



Agnostic Behavior Tree for scenario execution

Open Source:
https://gitlab.eclipse.org/mbauerm6f/simopenpass/-/tree/master/common

ASAM OpenSCENARIO V2.0.0 Scenario Compiler

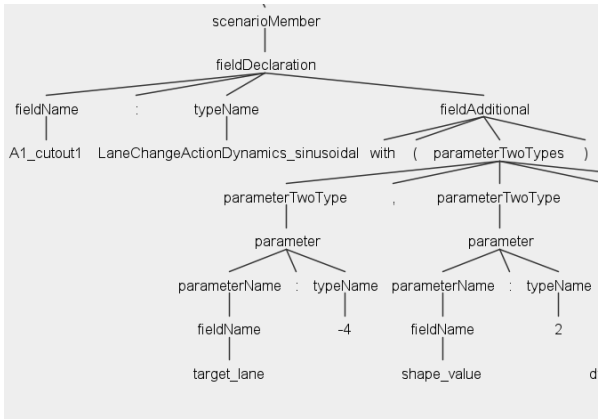
Preprocess | Compiler Frontend - Antlr4 | Compiler Backend - Scenario Engine

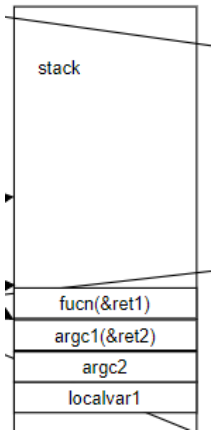
Standard Library → DSL Script → Syntax Tree → Script Engine → Semantic Model

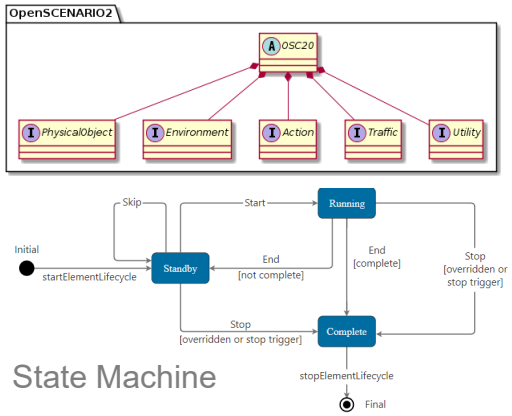
Standard Library | Grammar - EBNF | Lexical & Syntax Parser | Semantic Parser | Domain Model

```
type distance SI(m: 1)
type speed SI(m: 1, s: -1)

actor car_ego: Vehicle:
    driving_style:DrivingStyle

actor car: Vehicle:
    driving_style:DrivingStyle

def normal(mean: real, sd: real):r
    external.python("numpy.random.
```

```
grammar Scenario;
// Parser rules
osc
    :   preludeStatement* oscStatement* EOF
    ;

preludeStatement
    :   importStatement
    ;

oscStatement
    :   structDeclaration
    |   actorDeclaration
    |   actionDeclaration
    |   modifierDeclaration
    |   typeDeclaration
    |   unitDeclaration
```



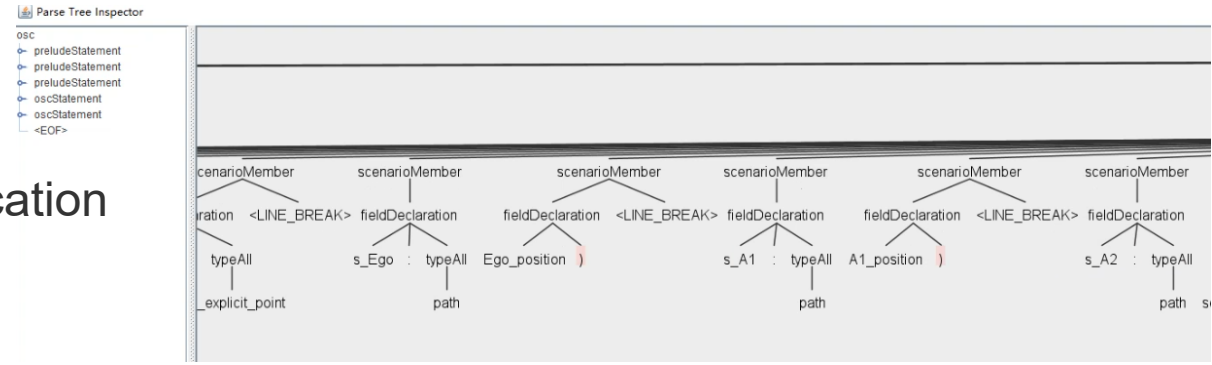


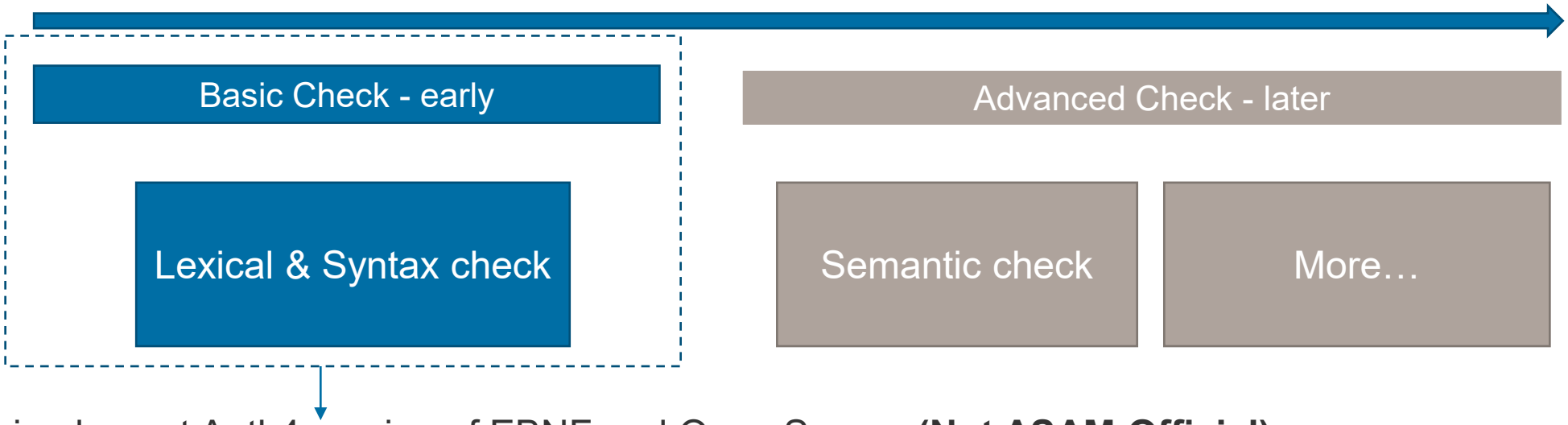


State Machine

Grammar Check Tool with EBNF of ASAM OpenSCENARIO V2.0.0

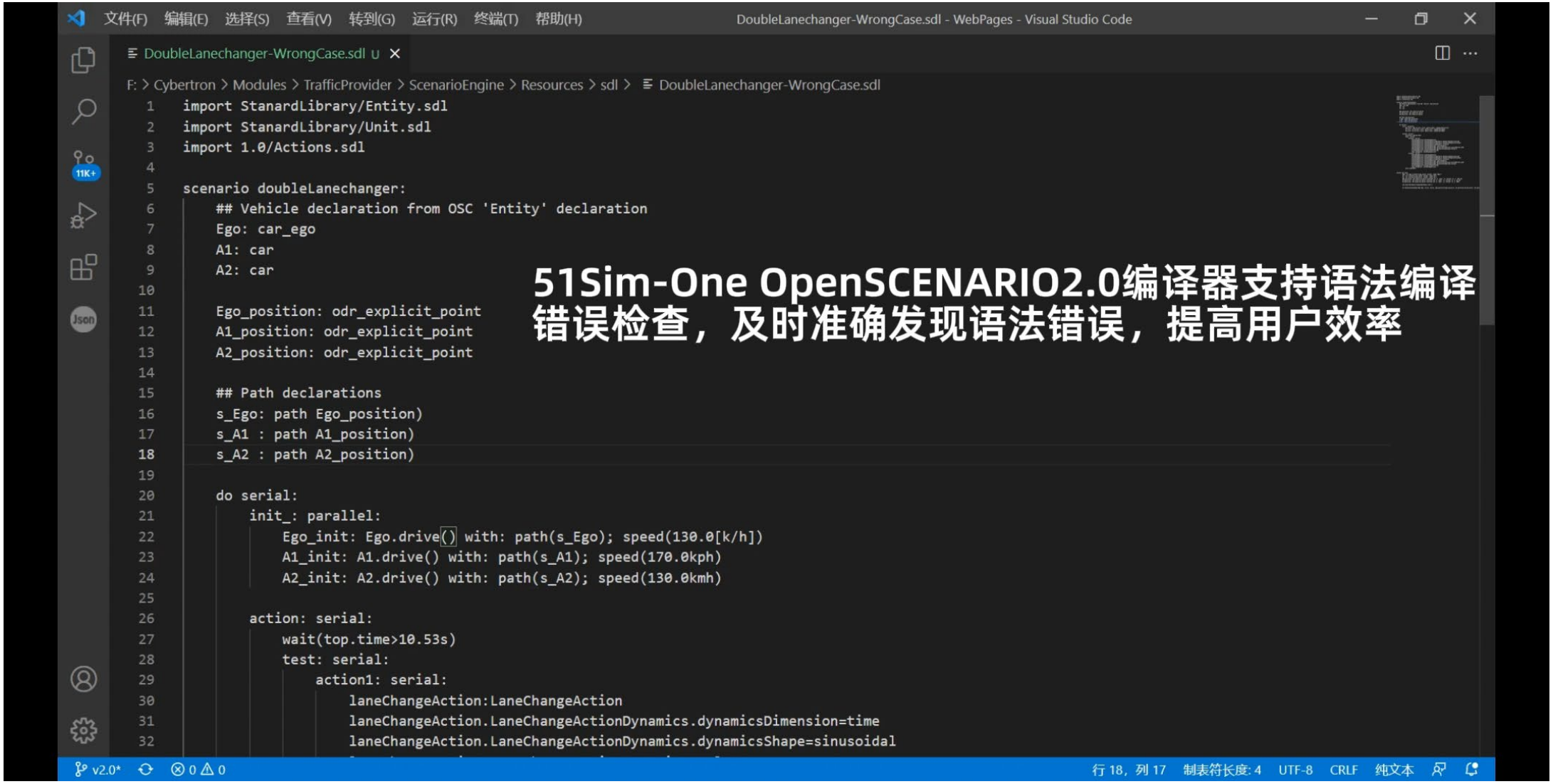
- Implementation:
  - Use Antlr4 grammar to implement EBNF of specification
- Useful
  - Useful for OpenSCENARIO 2.0 programmers
  - Useful for Simulation Implementers





Plan to implement Antlr4 version of EBNF and Open Source **(Not ASAM Official)**

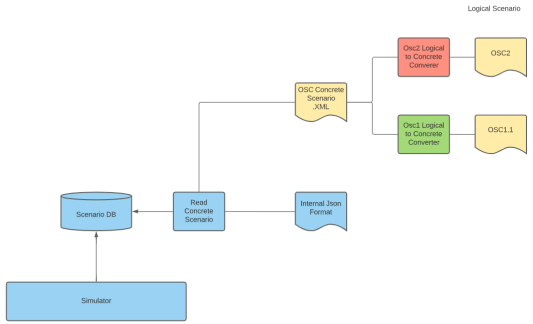
Demo – Double Lane Changer Grammar Check and Run



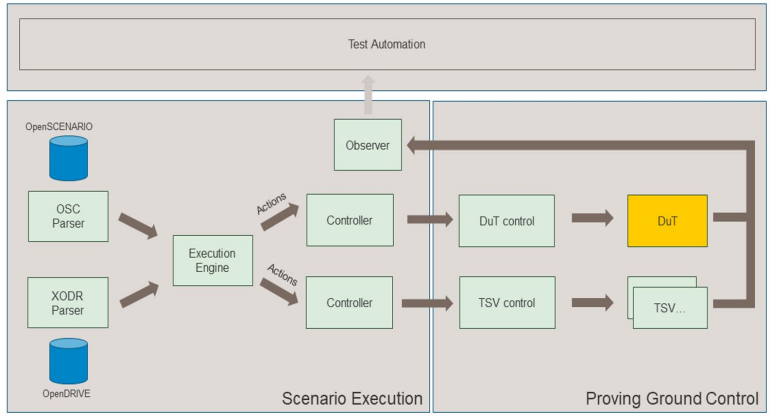
Implementers forum activities - More feedback to development project

More Implementation And Feedback

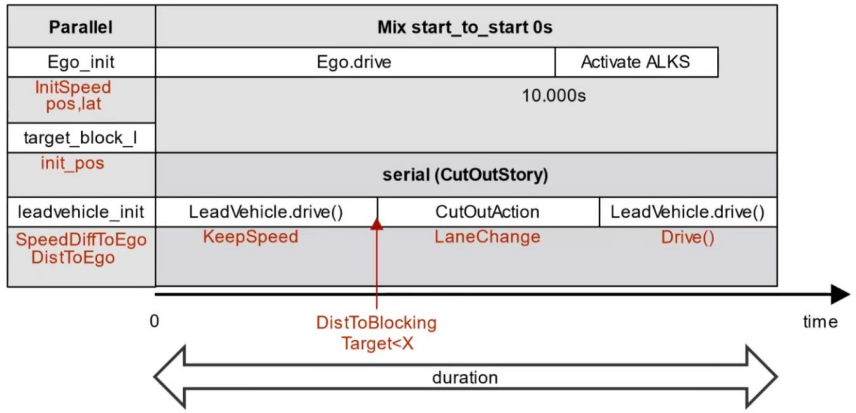
Ansys OSC2.0 logic scenario



AKKA Controller switching and junction scenario…

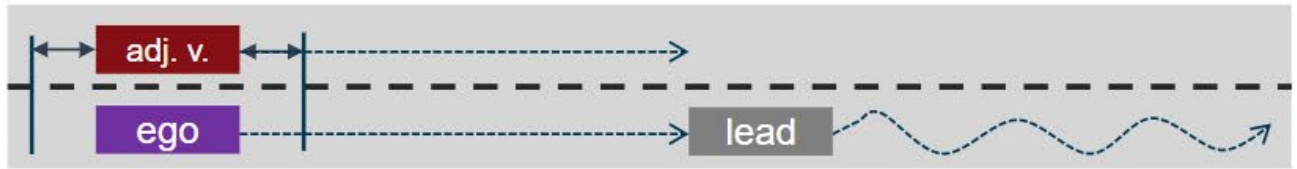


BTC embedded systems ALKS Cutout



A lot of other feedbacks, discussions and interesting presentations…

ALKS Swerving Lane – Scenario Specification




```
52  do parallel(duration: test_duration):
53
54      # 4.1.1. The test shall demonstrate that the ALKS does not leave its lane and maintains a stable position inside its ego lane
55      #        different curvatures within its system boundaries.
56      ego.drive() with:
57          # lane(left_of: adjacent_vehicle)
58          # lane(side_of: adjacent_vehicle, side: left)
59          lane(side_of: adjacent_vehicle, side: map.inner_side())
60          along(r1)
61          speed([0..60] kph)
62
63      # (c) With a lead vehicle swerving in the lane
64      lead_vehicle.drive() with:
65          lane(same_as: ego)
66          position([min_dist..max_dist]m, ahead_of: ego)
67
68          shape(sinusoid, max_amplitude=[min..max], period = [min..max]], max_deviation=...)  # pseudo syntax with intended behavior

76          # (d) With another vehicle driving close
77          do parallel()
78              adjacent_vehicle.FollowLane(0.375m)
79
80
81              adjacent_vehicle.drive() with:
82                  position([-1,1]m, ahead_of: ego)
```

ALKS Swerving Lane – Translation into internal scenario format

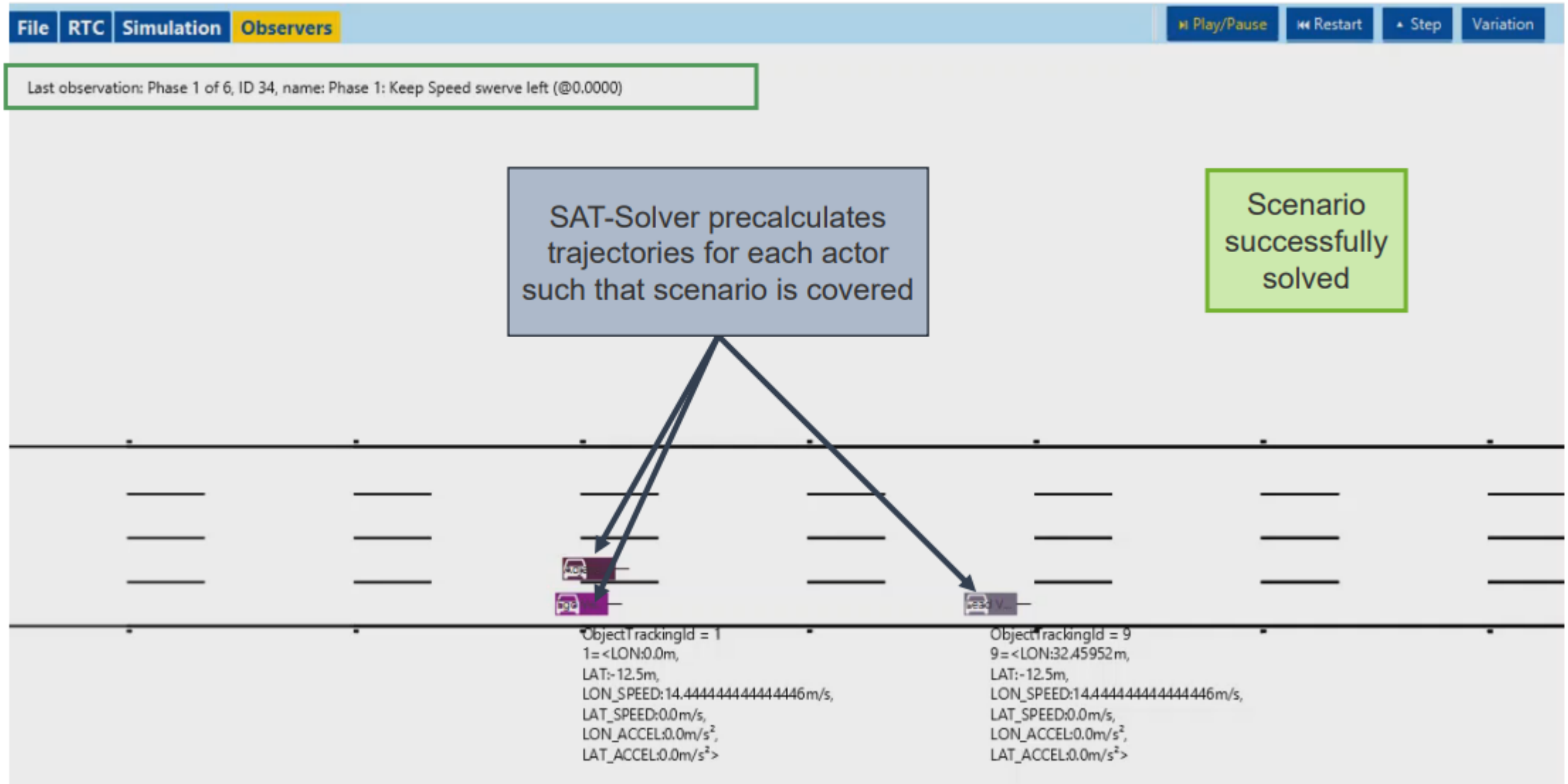
Serialization of swerving + parametrization

- Phase1
- Phase2
- Phase3
- Phase4
- Phase5
- Phase6

duration of each phase within
[minSwervePeriod, maxSwervePeriod]

```
double minEgoSpeed = 40.0/3.6;
double maxEgoSpeed = 60.0/3.6;
int adjacentVehicleInitPositionLaneId = -3;
double minSwervePeriod = 4.0;
double maxSwervePeriod = 5.0;
double swerveOffsetLeftInM = 0.7;
double swerveOffsetRightInM = 0.7;
double swerveTolerance = 0.02;
double swerve_MaxLateralAcc = 1.5;
double minLongDistEgoLead = 25.0;
double maxLongDistEgoLead = 2 * maxEgoSpeed;
double distToleranceEgoAdj = 1.0;
double maxSpeedDiffEgoLead = 3.0;
```

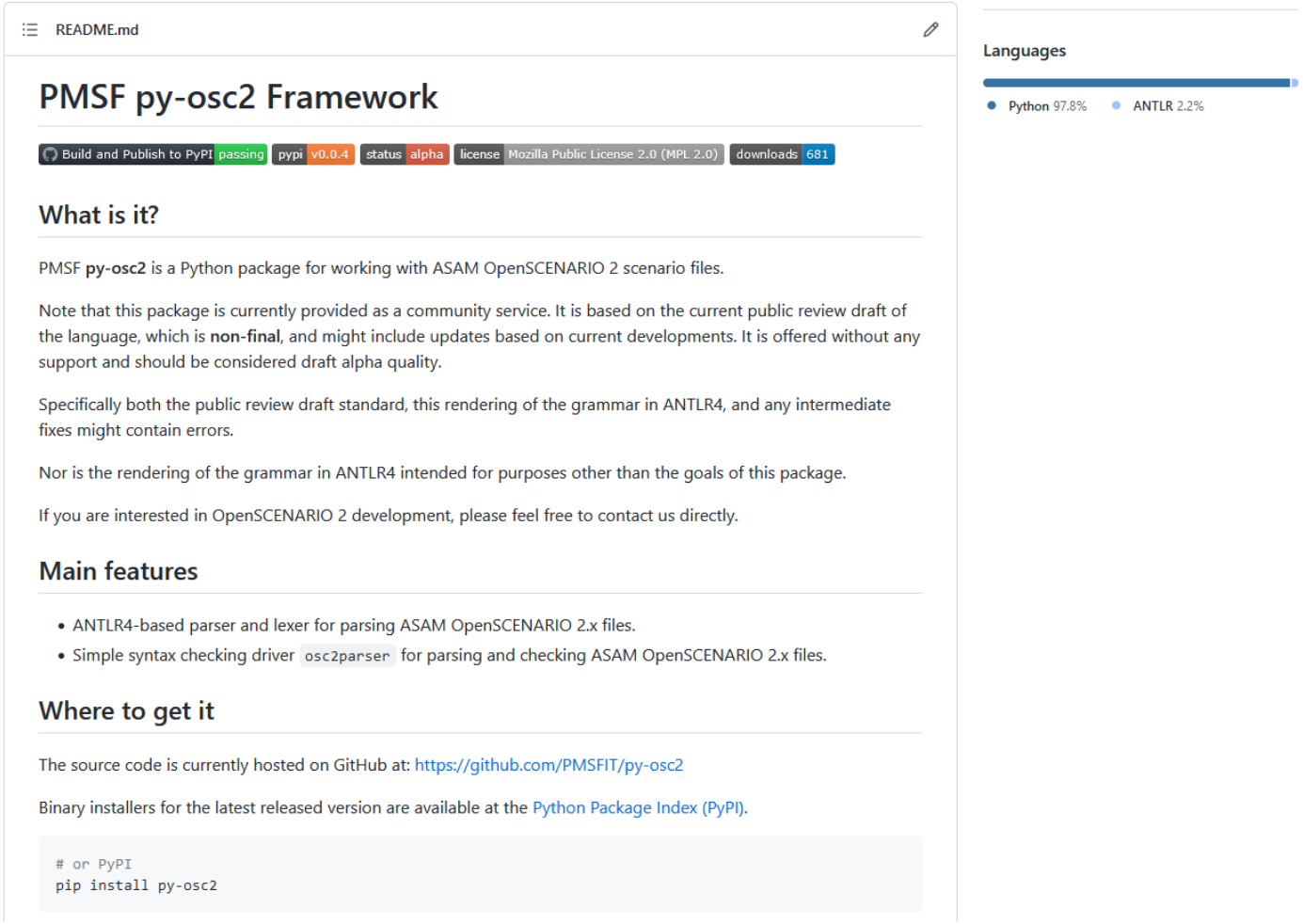
ALKS Swerving Lane – Pre-Simulation (Solver, no SUT)



PMSF py-osc2 Framework

Python-based Framework for OpenSCENARIO 2.x Files

- Currently provides:
  - ANTLR4-based parser
  - CLI tool for syntax checking
- Future enhancements:
  - More sophisticated syntax checking
  - Semantic checks (incl. Imports)
  - …
- Status:
  - Initial alpha releases, based on PRC
  - Will be updated as PRC is finalized
- Available on PyPI, GitHub (MPL 2.0):
  - https://pypi.org/project/py-osc2/
  - https://github.com/PMSFIT/py-osc2



README.md

PMSF py-osc2 Framework

Build and Publish to PyPI passing | pypi v0.0.4 | status alpha | license Mozilla Public License 2.0 (MPL 2.0) | downloads 681

What is it?

PMSF **py-osc2** is a Python package for working with ASAM OpenSCENARIO 2 scenario files.

Note that this package is currently provided as a community service. It is based on the current public review draft of the language, which is **non-final**, and might include updates based on current developments. It is offered without any support and should be considered draft alpha quality.

Specifically both the public review draft standard, this rendering of the grammar in ANTLR4, and any intermediate fixes might contain errors.

Nor is the rendering of the grammar in ANTLR4 intended for purposes other than the goals of this package.

If you are interested in OpenSCENARIO 2 development, please feel free to contact us directly.

Main features

- ANTLR4-based parser and lexer for parsing ASAM OpenSCENARIO 2.x files.
- Simple syntax checking driver `osc2parser` for parsing and checking ASAM OpenSCENARIO 2.x files.

Where to get it

The source code is currently hosted on GitHub at: https://github.com/PMSFIT/py-osc2

Binary installers for the latest released version are available at the Python Package Index (PyPI).

```
# or PyPI
pip install py-osc2
```

Languages

Python 97.8% ANTLR 2.2%

What will change looking forward…

- This forum will be the focus of the project, with WG leads coordinating
- It will be the main feedback channel on the PRC
- While we do not expect full industrial strength implementation to be done in 3 months, implementation work that will start will enable “onion peeling” of content
- Implementation work has already started during I.F. Work
- We are aware of other companies who started implementing already
- OSC 2 does not reinvent the wheel, some concepts have been implemented elsewhere, proven and can be used to accelerate the testing (see concept document)

What will change looking forward…

- Three parallel tracks:

1. Language Implementation
   What: Discussion and deep dives into the language and implementing it. Initially input based on current ToDos in the Language groups

2. Applying the Domain Model
   What: Discussion and resolution of feedback surrounding the domain model (actions, actors, road network, etc.)

3. Scenario Specification
   What: Initially, collectively increase knowledge of using the standard to write scenarios. As we progress → Q&A session with experts to resolve roadblocks in scenario specification

As we need to reevaluate interest in this forum, please **re-register** your interest to actively participate via an email to **osc2if-signup@asam.net**

Ben Engel
benjamin.engel@asam.net
Global Technology Manager
ASAM e.V.