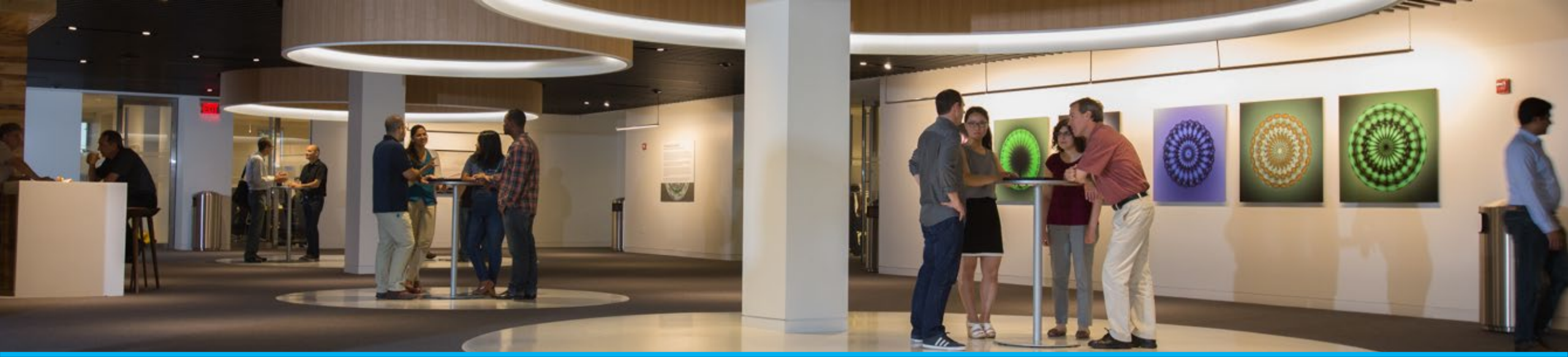
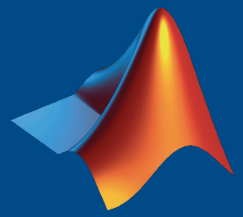
ASAM OpenSCENARIO 2.0

By
Naga Pemmaraju
Principal Product Manager
Autonomous Systems
MathWorks
npemmara@mathworks.com

© 2022 The MathWorks, Inc.





MathWorks®

Accelerating the pace of engineering and science

The leading developer of mathematical computing software for engineers and scientists.

Headquarters

Natick, MA USA

North America

United States

Europe

Finland
France
Germany
Ireland
Italy
Netherlands
Spain
Sweden
Switzerland
UK

Asia-Pacific

Australia
China
India
Japan
Korea

5 million+

users in over 190 countries

100,000+

businesses, governments, and universities

All of the top 10 automotive and aerospace companies

5000+ staff

in 34 offices around the world

$1.25+ billion

in revenues

Privately held

and profitable every year

Agenda

- ASAM OpenSCENARIO® V2.0
- OSC 1.x vs OSC 2.0
- Inter-tool Compatibility
- Recommendation
- Migration & Roadmap

Agenda

- ASAM OpenSCENARIO® V2.0
- OSC 1.x vs OSC 2.0
- Inter-tool Compatibility
- Recommendation
- Migration & Roadmap

ASAM OpenSCENARIO® V2.0

- Consistent scenario description over concrete, logical and abstract scenario levels
- Object oriented programming language - Human readable DSL
- Built in constraints, KPIs (success/fail criteria)
- Increased support for reuse and Inter-tool compatibility

DATASHEET

Title	ASAM OpenSCENARIO® V2.0.0
Domain	Simulation
Current Version	2.0.0
Release Date	20 Jul 2022
Application Areas	• Simulation • X-in-the-loop • Proving grounds
Specification Content	• Language specification • Domain model specification
File Formats	• osc

ASAM OpenSCENARIO® V2.0 - Scenario Abstraction

Abstract

A formal scenario that conceptualizes scenarios to the level of the scenario intent.

Logical

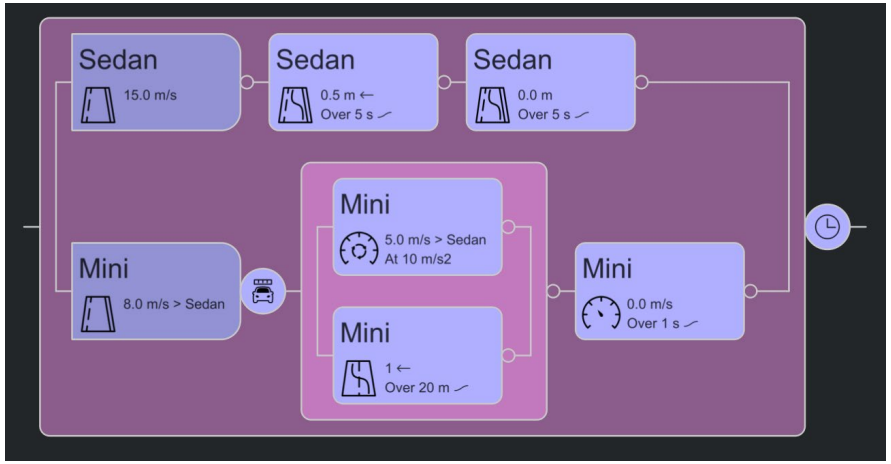
A scenario that does not specify all values for all parameters but provides a range of values that can be selected.

Concrete

A scenario for which the exact evolution of any of its parameters and variables is completely determined to a fixed value for any point in time.

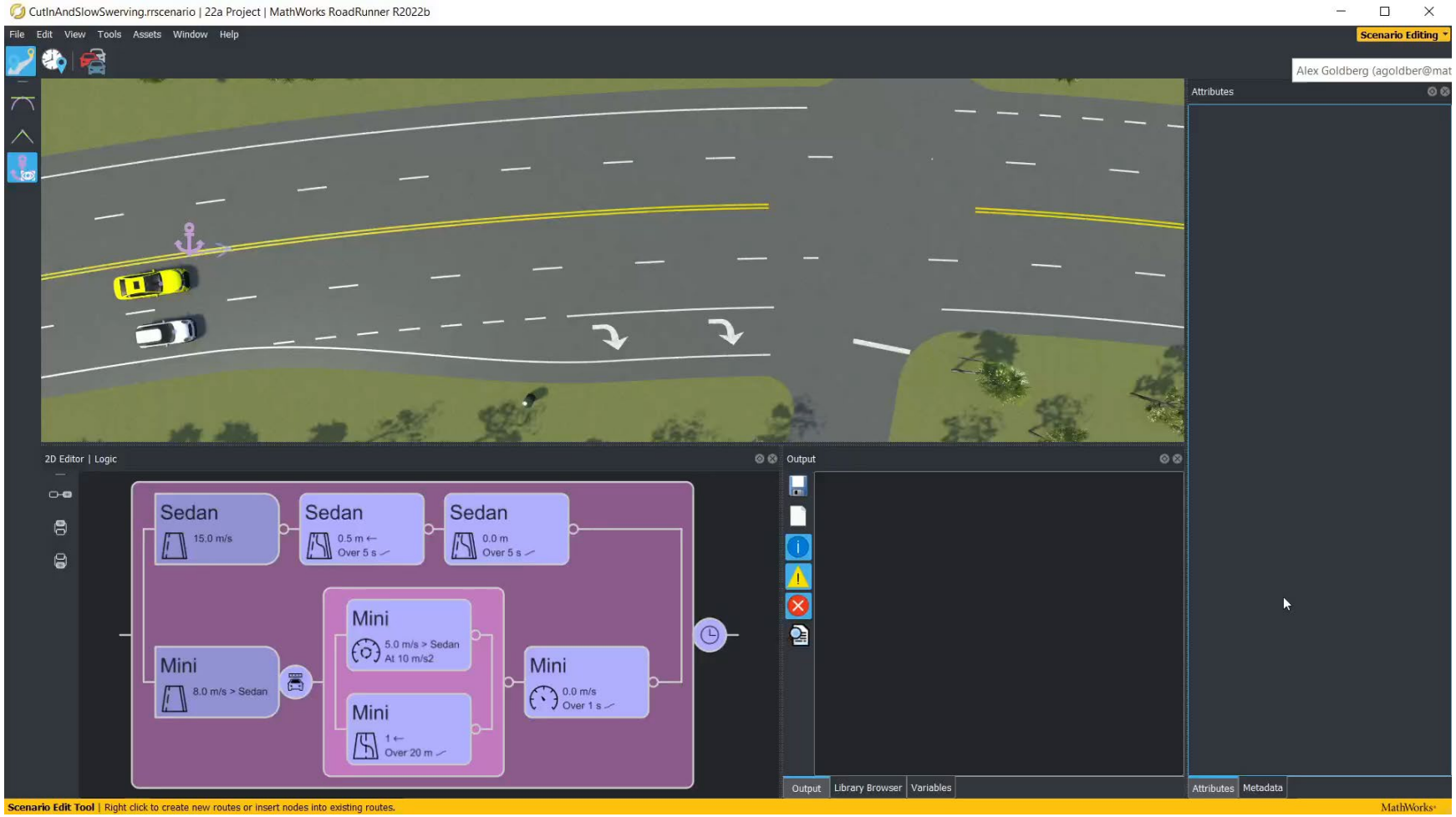
ASAM OpenSCENARIO V2.0 – Building Blocks

- The OpenSCENARIO v2.0 (“OSC 2.0”) structure is **simple** and **flexible**
  - Most scenarios are formed from a few simple building blocks:
    - *Parallel phase*: Executes members simultaneously
    - *Serial phase*: Executes members sequentially, in the defined order
    - *Action phase*: Executes a single action
    - *‘Until’ directive*: Ends a phase execution when a Boolean condition is satisfied



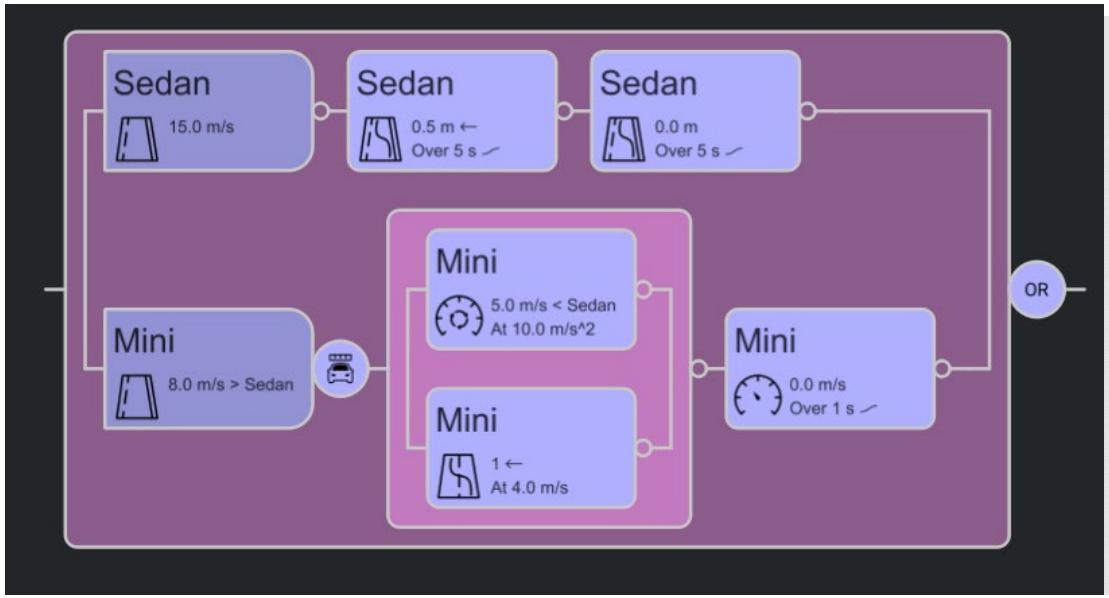
Example Scenario

- Cut-in and stop in front of swerving vehicle



ASAM OpenSCENARIO V2.0 – File Representation

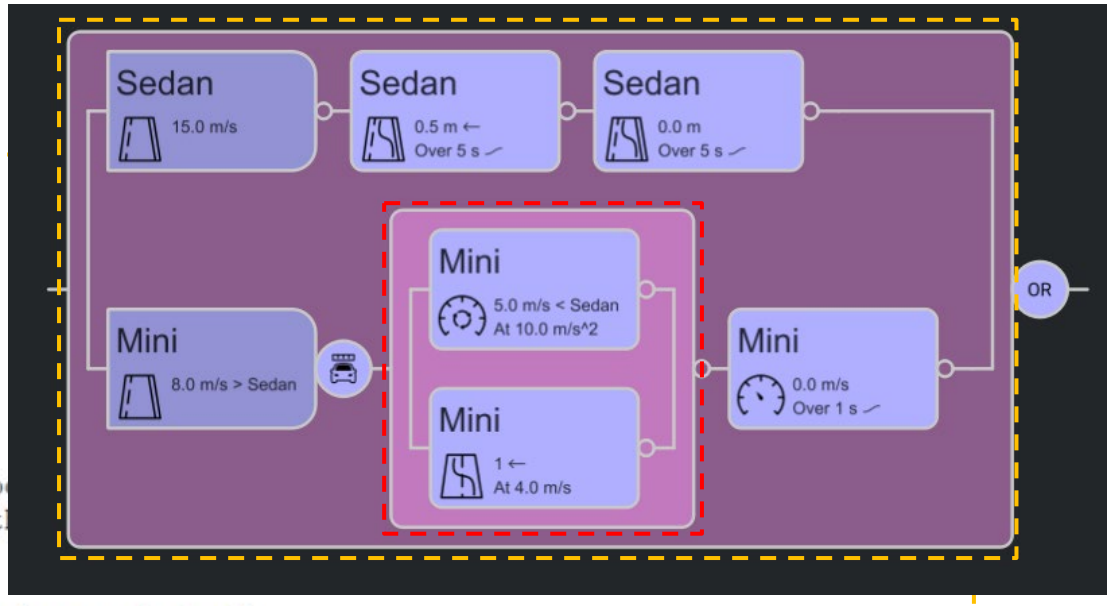
- Exported scenario has a compact, human-readable representation




```
96  do root_phase: parallel:
97      parallel:
98          mini.drive() with:
99              along(compact_car_route)
100         sedan.drive() with:
101             along(sedan_route)
102     phase_1: parallel:
103         phase_2: serial:
104             phase_3: sedan.assign_speed(15mps)
105             phase_4: sedan.follow_lane(0.5m, rate_profile: smooth, duration: 5s)
106             phase_5: sedan.follow_lane(0m, rate_profile: smooth, duration: 5s)
107         phase_6: serial:
108             phase_7: mini.drive() with:
109                 speed(8mps, faster_than: sedan, at: start, track: projected)
110                 until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
111             phase_8: parallel:
112                 phase_9: serial:
113                     mini.drive() with:
114                         speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
115                     mini.drive() with:
116                         speed(5mps, slower_than: sedan, at: all, track: actual)
117                 phase_10: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
118             phase_11: mini.change_speed(0mps, rate_profile: smooth, duration: 1s)
119     with:
120         until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_to_collisio
121
122
```

ASAM OpenSCENARIO V2.0 – Building Blocks

- *Parallel phase:* Executes members simultaneously

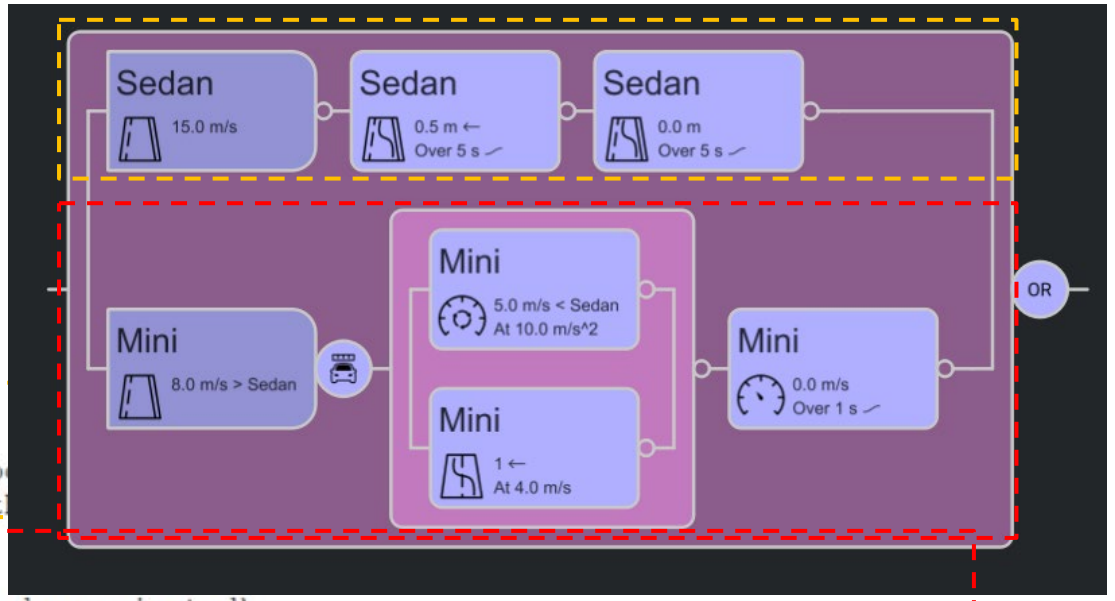



```
96   do root_phase: parallel:
97       parallel:
98           mini.drive() with:
99               along(compact_car_route)
100          sedan.drive() with:
101              along(sedan_route)
102      phase_1: parallel:
103          phase_2: serial:
104              phase_3: sedan.assign_speed(15mps)
105              phase_4: sedan.follow_lane(0.5m, rate_profile: smo
106              phase_5: sedan.follow_lane(0m, rate_profile: smoot
107          phase_6: serial:
108              phase_7: mini.drive() with:
109                  speed(8mps, faster_than: sedan, at: start, track: projected)
110                  until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
111              phase_8: parallel:
112                  phase_9: serial:
113                      mini.drive() with:
114                          speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
115                      mini.drive() with:
116                          speed(5mps, slower_than: sedan, at: all, track: actual)
117                  phase_10: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
118              phase_11: mini.change_speed(0mps, rate_profile: smooth, duration: 1s)
119      with:
120          until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_t
121
122
```

ASAM OpenSCENARIO V2.0 – Building Blocks

- *Serial phase*: Executes members sequentially, in the defined order

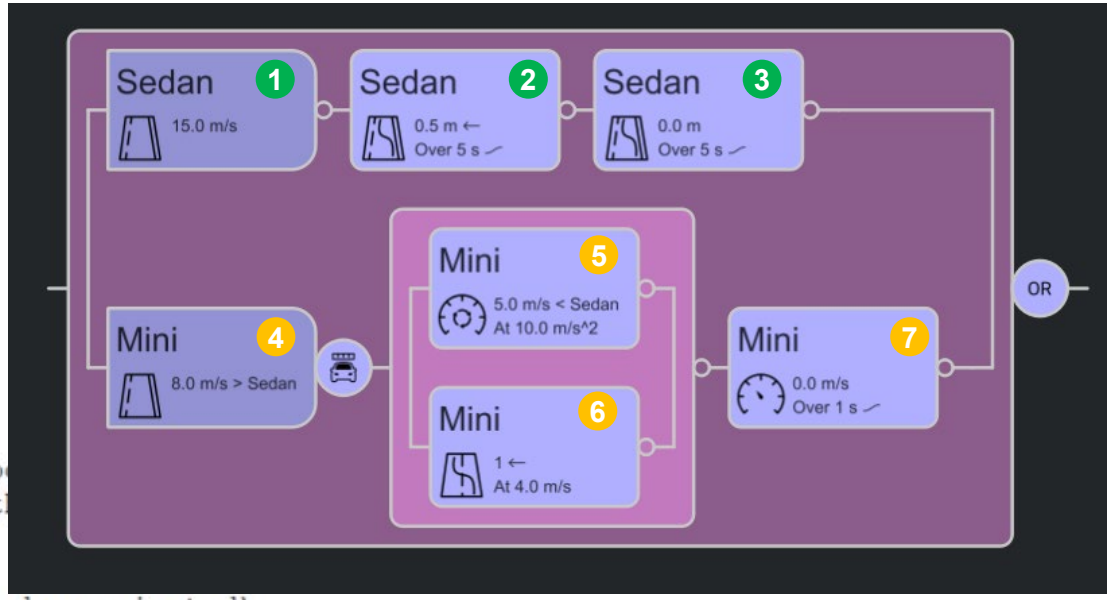
```
96    do root_phase: parallel:
97        parallel:
98            mini.drive() with:
99                along(compact_car_route)
100           sedan.drive() with:
101               along(sedan_route)
102       phase_1: parallel:
103           phase_2: serial:
104               phase_3: sedan.assign_speed(15mps)
105               phase_4: sedan.follow_lane(0.5m, rate_profile: smo
106               phase_5: sedan.follow_lane(0m, rate_profile: smoot
107           phase_6: serial:
108               phase_7: mini.drive() with:
109                   speed(8mps, faster_than: sedan, at: start, track: projected)
110                   until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
111               phase_8: parallel:
112                   phase_9: serial:
113                       mini.drive() with:
114                           speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
115                       mini.drive() with:
116                           speed(5mps, slower_than: sedan, at: all, track: actual)
117                   phase_10: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
118               phase_11: mini.change_speed(0mps, rate_profile: smooth, duration: 1s)
119       with:
120           until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_t
121
122
```



ASAM OpenSCENARIO V2.0 – Building Blocks

- *Action phase*: Executes a single action

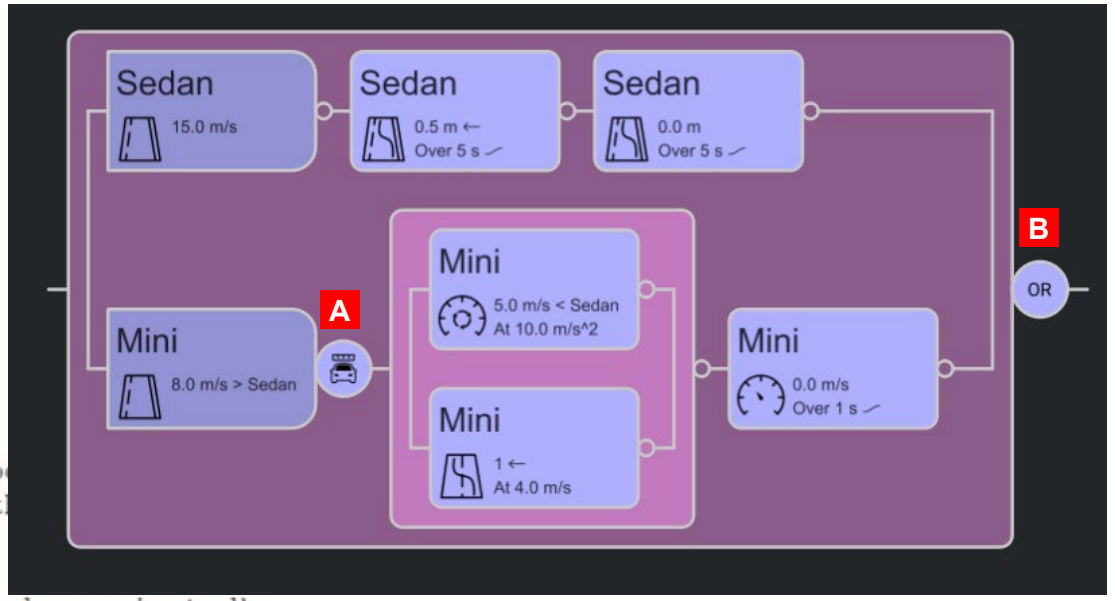
```
96  do root_phase: parallel:
97      parallel:
98          mini.drive() with:
99              along(compact_car_route)
100         sedan.drive() with:
101             along(sedan_route)
102     phase_1: parallel:
103         phase_2: serial:
104           1 phase_3: sedan.assign_speed(15mps)
105           2 phase_4: sedan.follow_lane(0.5m, rate_profile: smo
106           3 phase_5: sedan.follow_lane(0m, rate_profile: smoot
107         phase_6: serial:
108           4 phase_7: mini.drive() with:
109                 speed(8mps, faster_than: sedan, at: start, track: projected)
110                 until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
111             phase_8: parallel:
112                 phase_9: serial:
113               5     mini.drive() with:
114                         speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
115                     mini.drive() with:
116                         speed(5mps, slower_than: sedan, at: all, track: actual)
117               6 phase_10: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
118           7 phase_11: mini.change_speed(0mps, rate_profile: smooth, duration: 1s)
119     with:
120         until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_t
121
122
```



ASAM OpenSCENARIO V2.0 – Building Blocks

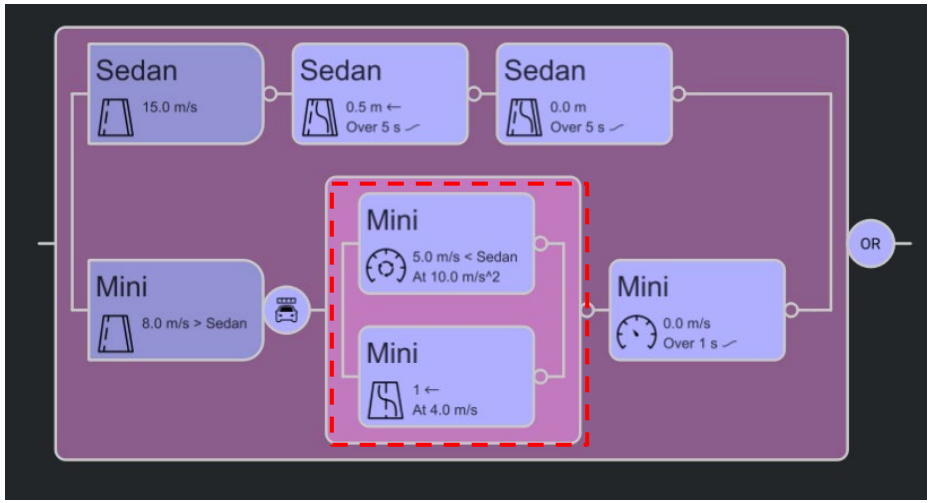
'Until' directive: Ends a phase execution when a Boolean condition is satisfied

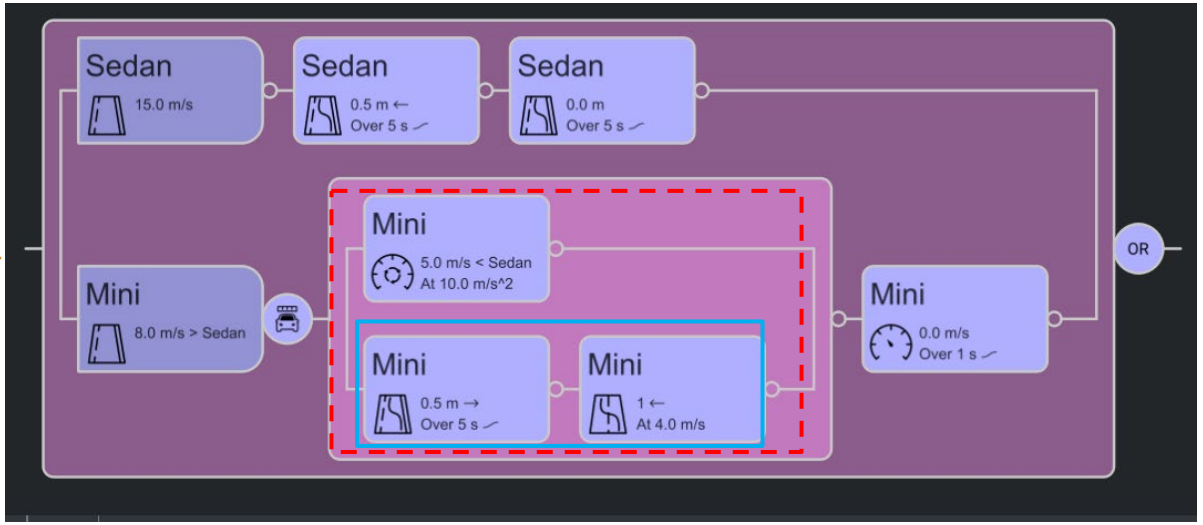
```
96      do root_phase: parallel:
97          parallel:
98              mini.drive() with:
99                  along(compact_car_route)
100             sedan.drive() with:
101                 along(sedan_route)
102         phase_1: parallel:
103             phase_2: serial:
104                 phase_3: sedan.assign_speed(15mps)
105                 phase_4: sedan.follow_lane(0.5m, rate_profile: smo
106                 phase_5: sedan.follow_lane(0m, rate_profile: smoot
107             phase_6: serial:
108                 phase_7: mini.drive() with:
109                     speed(8mps, faster_than: sedan, at: start, track: projected)
110                   A until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
111                 phase_8: parallel:
112                     phase_9: serial:
113                         mini.drive() with:
114                             speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
115                         mini.drive() with:
116                             speed(5mps, slower_than: sedan, at: all, track: actual)
117                     phase_10: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
118                 phase_11: mini.change_speed(0mps, rate_profile: smooth, duration: 1s)
119         with:
120           B until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_t
121
122
```



ASAM OpenSCENARIO V2.0 – Model Editing

- Lets say, we introduce a new serial phase within a parallel phase



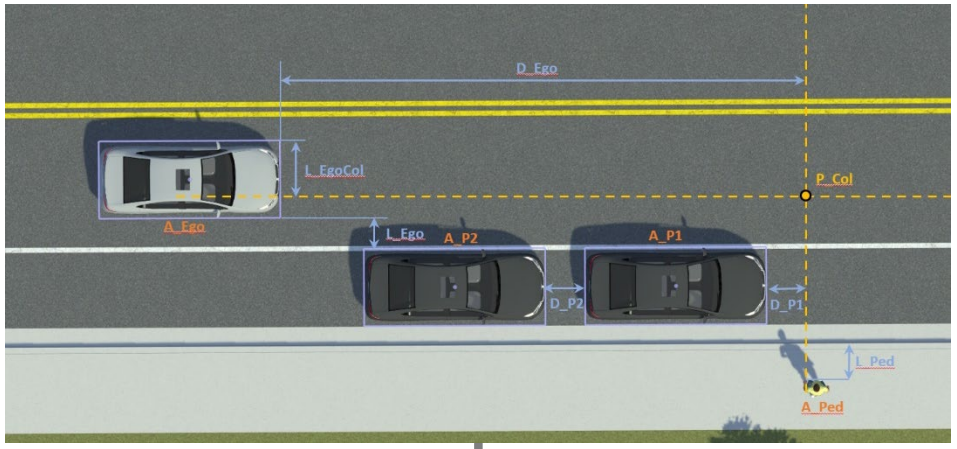



```
do root_phase: parallel:
    parallel:
        mini.drive() with:
            along(compact_car_route)
        sedan.drive() with:
            along(sedan_route)
    phase_1: parallel:
        phase_2: serial:
            phase_3: sedan.assign_speed(15mps)
            phase_4: sedan.follow_lane(0.5m, rate_profile: smooth, duration: 5s)
            phase_5: sedan.follow_lane(0m, rate_profile: smooth, duration: 5s)
        phase_6: serial:
            phase_7: mini.drive() with:
                speed(8mps, faster_than: sedan, at: start, track: projected)
                until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
            phase_8: parallel:
                phase_9: serial:
                    mini.drive() with:
                        speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
                    mini.drive() with:
                        speed(5mps, slower_than: sedan, at: all, track: actual)
                phase_10: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
    with:
        until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_to_collision(mini) == 0s))
```

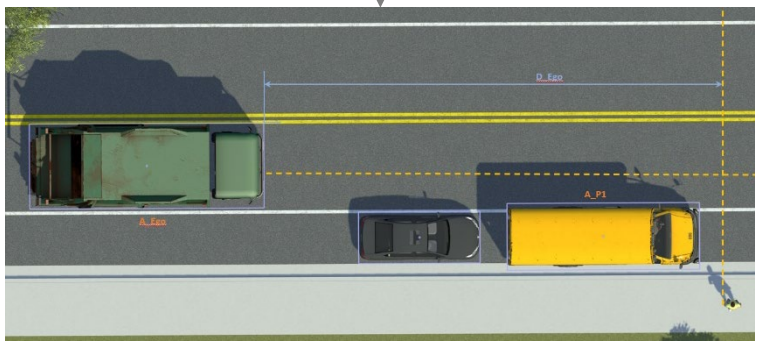
```
do root_phase: parallel:
    parallel:
        mini.drive() with:
            along(compact_car_route)
        sedan.drive() with:
            along(sedan_route)
    phase_1: parallel:
        phase_2: serial:
            phase_3: sedan.assign_speed(15mps)
            phase_4: sedan.follow_lane(0.5m, rate_profile: smooth, duration: 5s)
            phase_5: sedan.follow_lane(0m, rate_profile: smooth, duration: 5s)
        phase_6: serial:
            phase_7: mini.drive() with:
                speed(8mps, faster_than: sedan, at: start, track: projected)
                until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
            phase_8: parallel:
                phase_9: serial:
                    mini.drive() with:
                        speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
                    mini.drive() with:
                        speed(5mps, slower_than: sedan, at: all, track: actual)
                phase_10: serial:
                    phase_11: mini.follow_lane(-0.5m, rate_profile: smooth, duration: 5s)
                    phase_12: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
    with:
        until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_to_collision(mini) == 0s))
```

ASAM OpenSCENARIO V2.0 – Reusable Scenarios

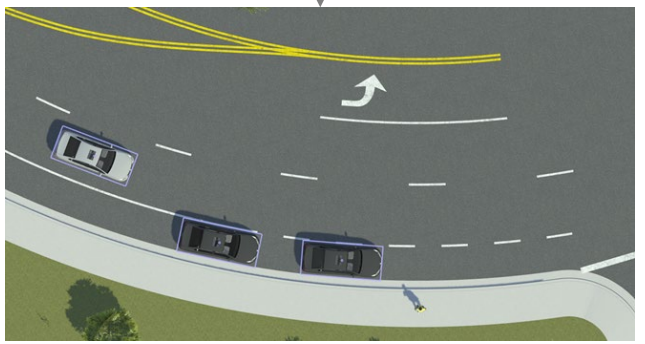
- Factoring out the commonality between scenarios and capturing them
- Uses abstract scenario and its variations
- Supports incremental development of scenarios



Change actor types



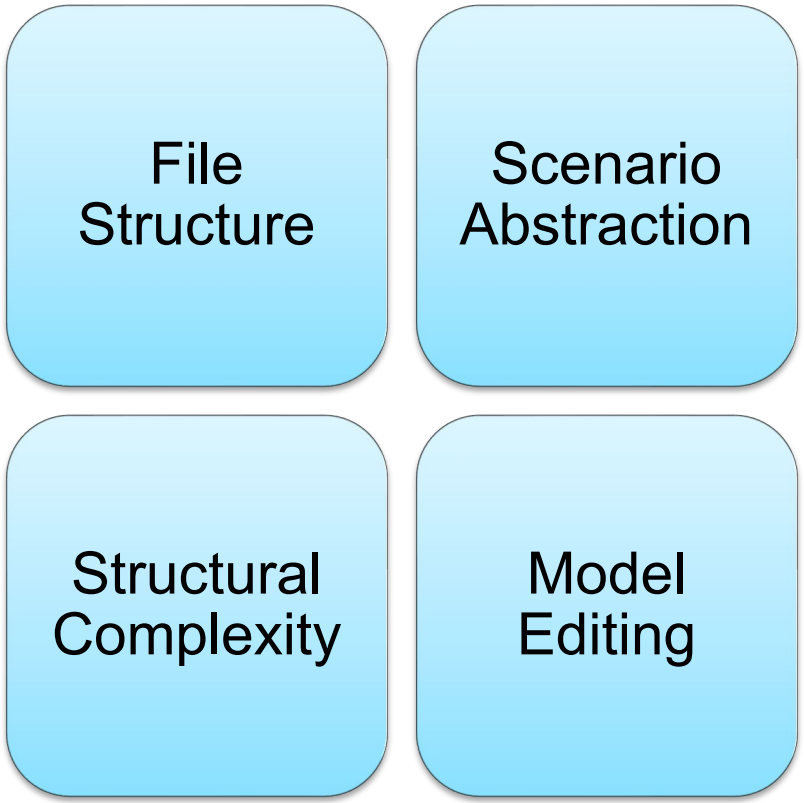
Translocate to new map



Agenda

- ASAM OpenSCENARIO® V2.0
- OSC 1.x vs OSC 2.0
- Inter-tool Compatibility
- Recommendation
- Migration & Roadmap

OSC 1.x vs OSC 2.0



OSC 1.x vs OSC 2.0 – File Structure

OSC 1.x

```
        </Actions>
    </Init>
    <Story name="Story">
        <Act name="Act">
            <ManeuverGroup name="Act_SedanGroup" maximumExecutionCount="1">
                <Actors selectTriggeringEntities="false">
                    <EntityRef entityRef="Sedan"/>
                </Actors>
                <Maneuver name="Act_SedanGroup_Maneuver">
                    <Event name="Act_SedanGroup_Event" priority="parallel">
                        <Action name="Act_SedanGroup_Change_Lateral_Offset">
                            <PrivateAction>
                                <LateralAction>
                                    <LaneOffsetAction continuous="false">
                                        <LaneOffsetActionDynamics dynamicsShape="cubic"/>
                                        <LaneOffsetTarget>
                                            <AbsoluteTargetLaneOffset value="0.5"/>
                                        </LaneOffsetTarget>
                                    </LaneOffsetAction>
                                </LateralAction>
                            </PrivateAction>
                        </Action>
                        <StartTrigger>
                            <ConditionGroup>
                                <Condition name="EnclosingActRunning" conditionEdge="none" delay="0.0000000000000000e+0">
                                    <ByValueCondition>
                                        <StoryboardElementStateCondition storyboardElementType="act" storyboardElementRef="Act" state="runningState"/>
                                    </ByValueCondition>
                                </Condition>
                            </ConditionGroup>
                        </StartTrigger>
                    </Event>
                    <Event name="Act_SedanGroup_Event2" priority="parallel">
                        <Action name="Act_SedanGroup_Change_Lateral_Offset2">
                            <PrivateAction>
                                <LateralAction>
                                    <LaneOffsetAction continuous="false">
                                        <LaneOffsetActionDynamics dynamicsShape="cubic"/>
                                        <LaneOffsetTarget>
                                            <AbsoluteTargetLaneOffset value="0"/>
                                        </LaneOffsetTarget>
                                    </LaneOffsetAction>
                                </LateralAction>
                            </PrivateAction>
                        </Action>
                        <StartTrigger>
                            <ConditionGroup>
                                <Condition name="Phase_State" conditionEdge="none" delay="0.0000000000000000e+0">
                                    <ByValueCondition>
                                        <StoryboardElementStateCondition storyboardElementType="event" storyboardElementRef="Act_SedanGroup_Event" state="completeState"/>
                                    </ByValueCondition>
                                </Condition>
                            </ConditionGroup>
                        </StartTrigger>
                    </Event>
```

OSC 2.0

```
sedan_route_start_point: route_point = map.odr_to_route_point(road_id: 7, lane_id: -1, s: 16.4739m, t: -4.85879e-6m)
sedan_route_lane: lane with:
    keep('net.asam.opendrive: roadId:7, laneId:-1' in it.anchors)
sedan_route: route = map.create_route([sedan_route_start_point, sedan_route_lane], connect_points_by: lane)

compact_car_route_start_point: route_point = map.odr_to_route_point(road_id: 7, lane_id: -2, s: 16.9211m, t: 2.9193e-5m)
compact_car_route_lane: lane with:
    keep('net.asam.opendrive: roadId:7, laneId:-2' in it.anchors)
compact_car_route: route = map.create_route([compact_car_route_start_point, compact_car_route_lane], connect_points_by: lane)

speed_shape_1: common_speed_shape with:
    keep(it.rate_profile == constant)
    keep(it.rate_peak ==  10mpss)

do root_phase: parallel:
    parallel:
        mini.drive() with:
            along(compact_car_route)
        sedan.drive() with:
            along(sedan_route)
    phase_1: parallel:
        phase_2: serial:
            phase_3: sedan.assign_speed(15mps)
            phase_4: sedan.follow_lane(0.5m, rate_profile: smooth, duration: 5s)
            phase_5: sedan.follow_lane(0m, rate_profile: smooth, duration: 5s)
        phase_6: serial:
            phase_7: mini.drive() with:
                speed(8mps, faster_than: sedan, at: start, track: projected)
                until (mini.mw_actor_euclidean_dist(sedan)) >= 7m)
            phase_8: parallel:
                phase_9: serial:
                    mini.drive() with:
                        speed(5mps, slower_than: sedan, at: end, track: actual, shape: speed_shape_1)
                    mini.drive() with:
                        speed(5mps, slower_than: sedan, at: all, track: actual)
                phase_10: serial:
                    phase_11: mini.follow_lane(-0.5m, rate_profile: smooth, duration: 5s)
                    phase_12: mini.change_lane(1, side: left, rate_profile: smooth, rate_peak: 4mps)
            phase_13: mini.change_speed(0mps, rate_profile: smooth, duration: 1s)
    with:
        until ((environment.datetime - sample(environment.datetime, @root_phase.start) >= 60s) or (sedan.time_to_collision(mini) == 0s))
```

OSC 2.0 is Human Readable

OSC 1.x vs OSC 2.0 – Scenario Abstraction

Version	Concrete	Logical	Abstract	Functional
V1.1.1	✔	✘	✘	✘
V1.2.0	✔	✔	✘	✘
V2.0.0	✔	✔	✔	✘

Levels of Scenario abstraction – OSC 1.x vs OSC 2.0

OSC 1.x vs OSC 2.0 – Structural Complexity

- **“Structural Complexity”** (for a scenario)

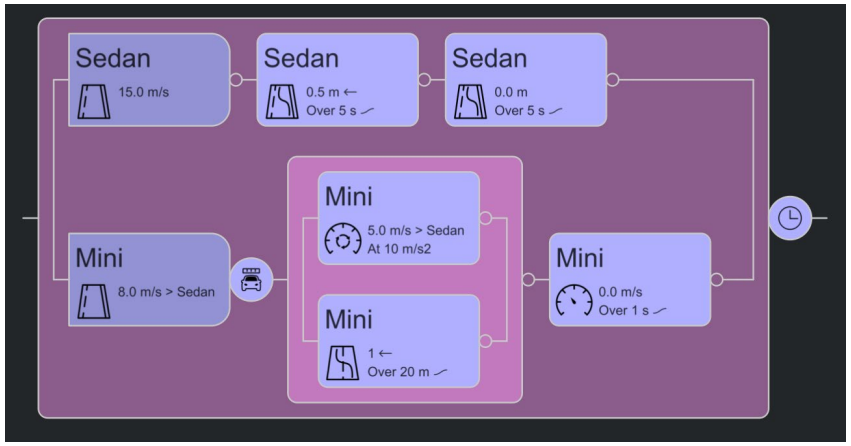
$$= \frac{\text{Number of structural (non-action) elements}}{\text{Number of actions}}$$

  - Lower = simpler
- OSCv1.x has many structural elements

OSC 1.x vs OSC 2.0 – Structural Complexity

- Example (equivalent scenario):
  - OSCv2.0 structural complexity = 4/7 = **.57**
  - OSCv1.x structural complexity = 15/7 = **2.14**

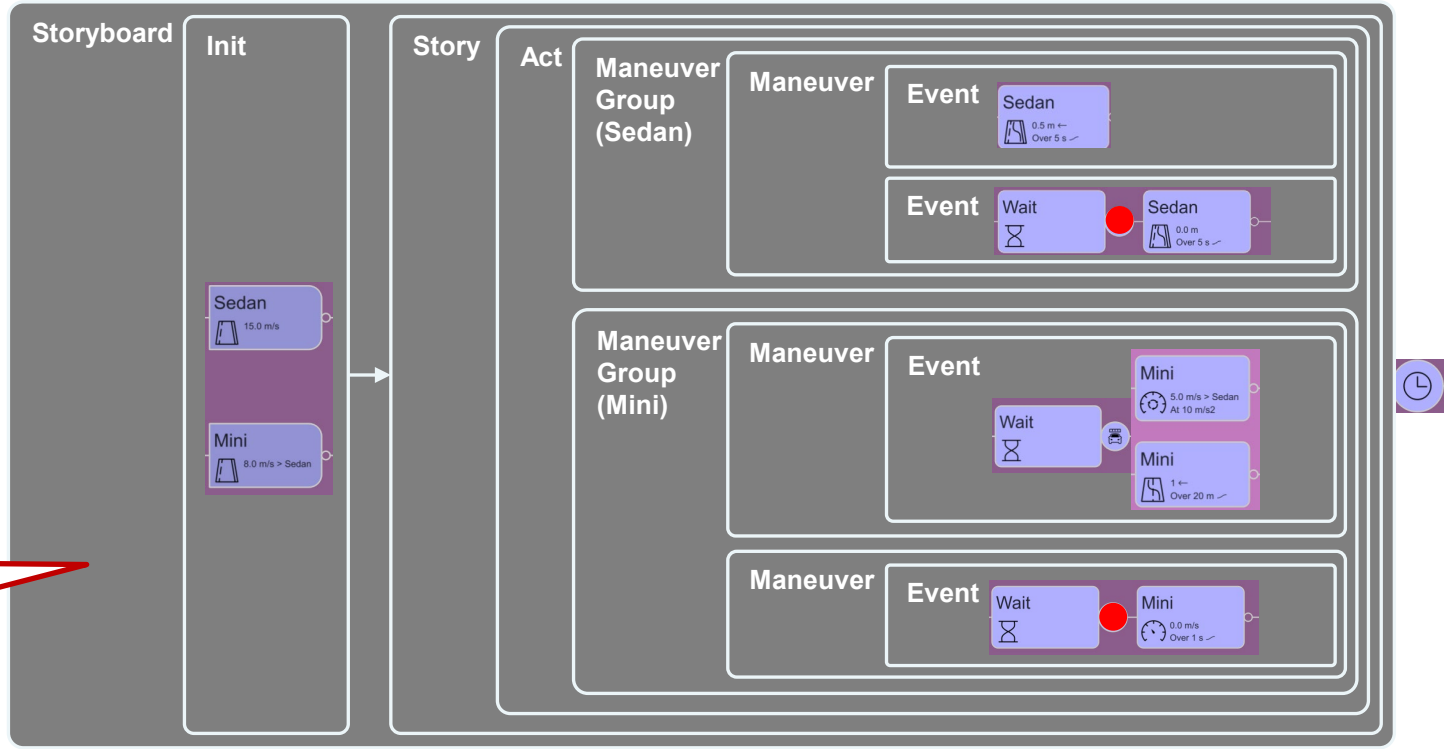
OSCv2.0 representation



7 actions, 4 compositional elements

High Structural Complexity for OSC1.x

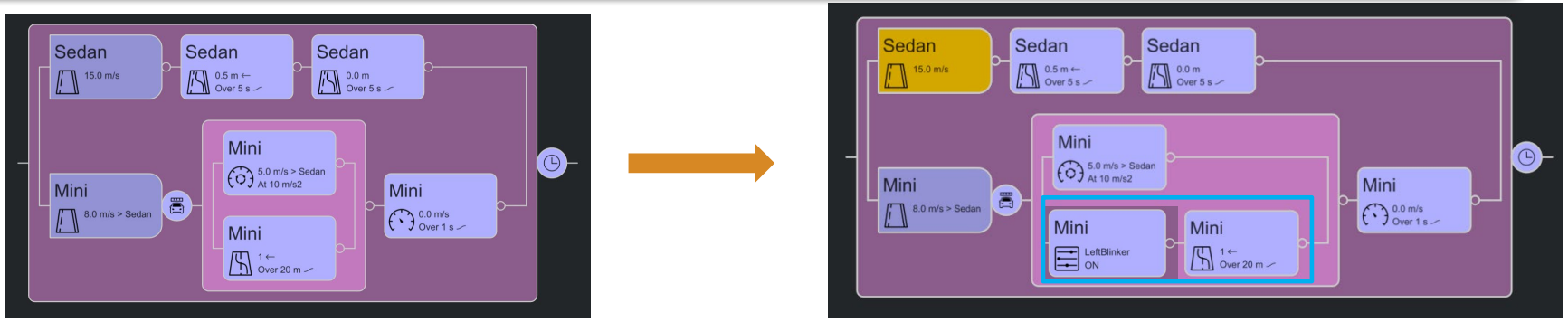
Equivalent OSCv1.x (most compact representation – manually created)



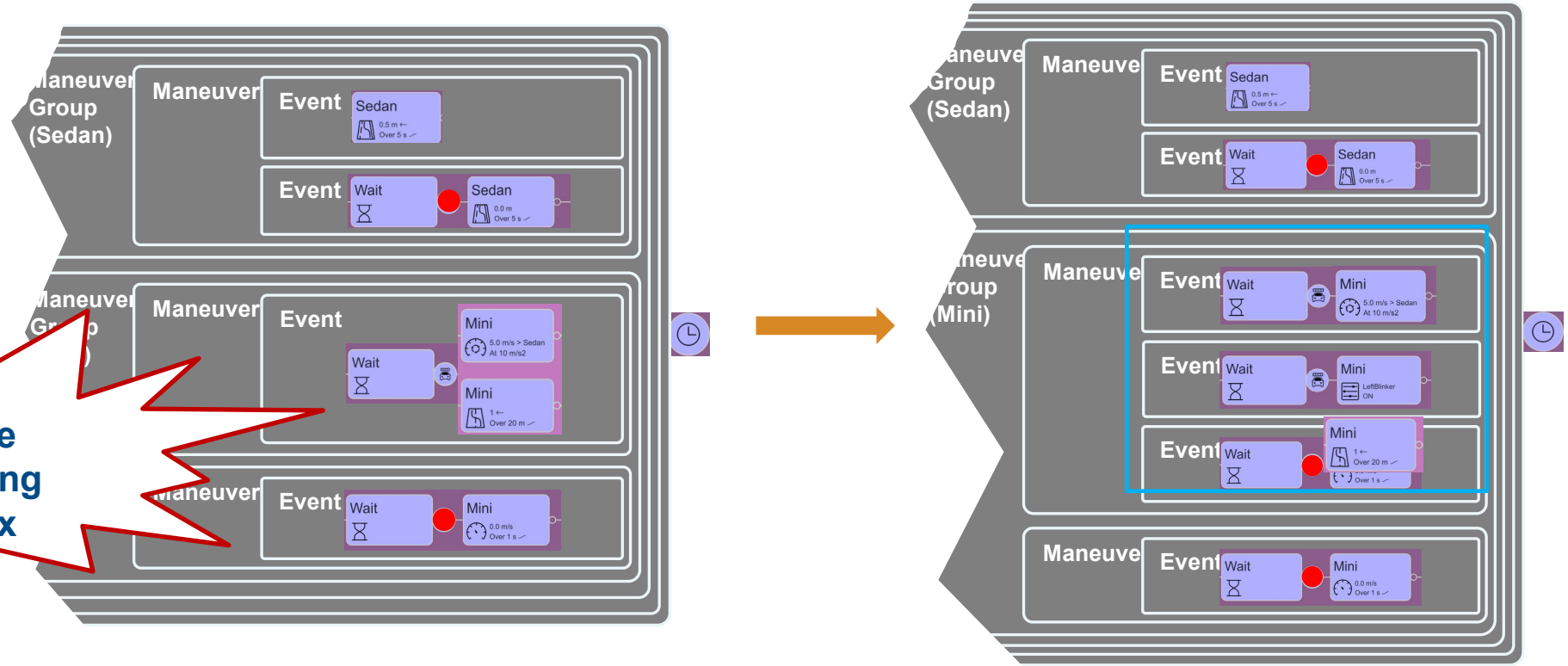
7 actions, 13 structural elements + 2 StoryboardElementStateConditions (●) for sequentialization

OSC 1.x vs OSC 2.0 – Model Editing

OSCv2.0 edit operation



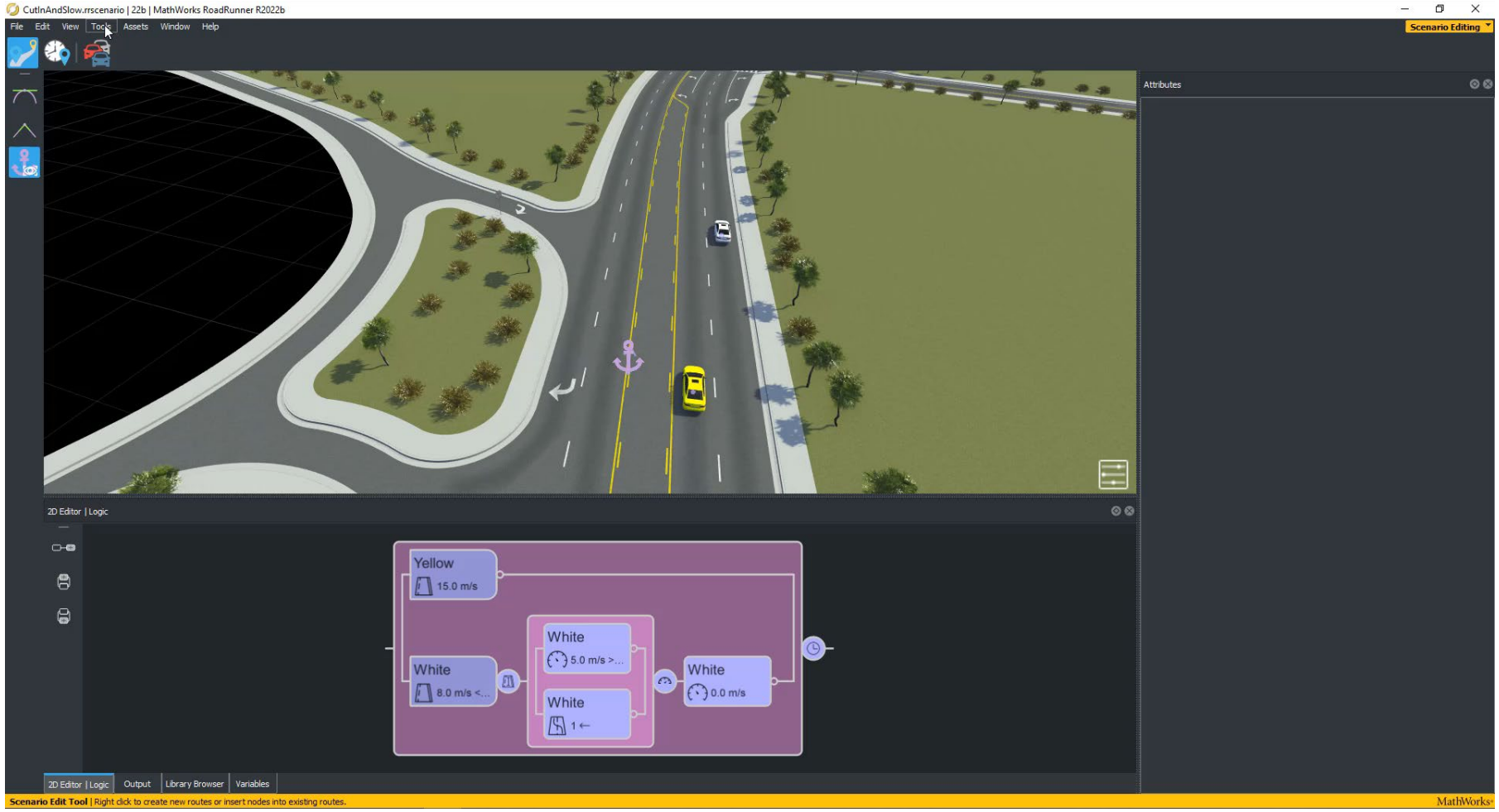
OSCv1.x edit operation



Unintuitive Model Editing for OSC1.x

Transform to OSC 1.X Model Structure

Addressing challenges in conversion

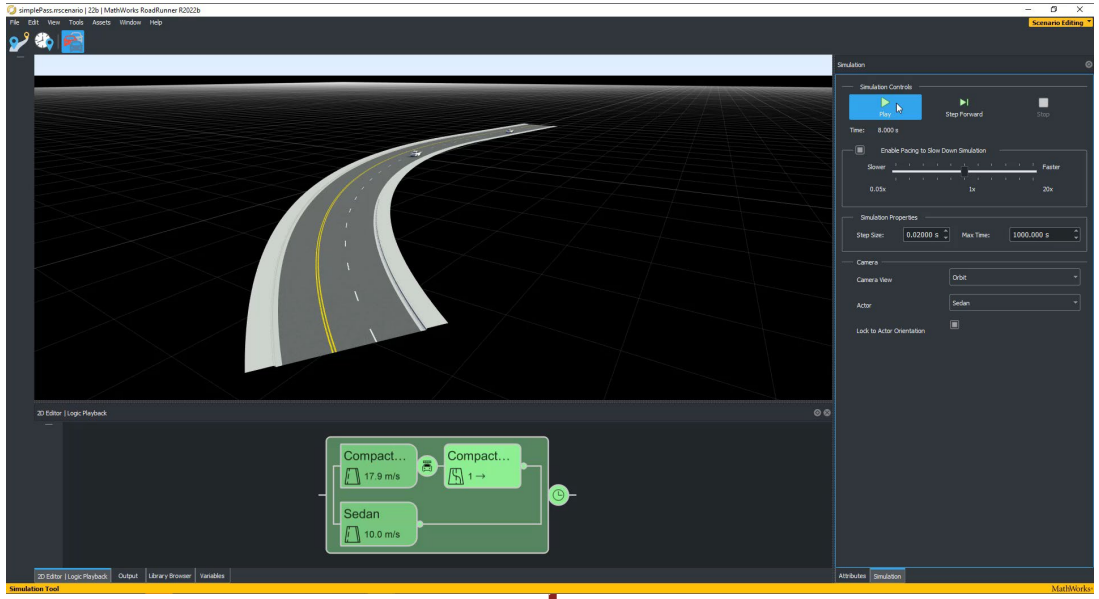


Agenda

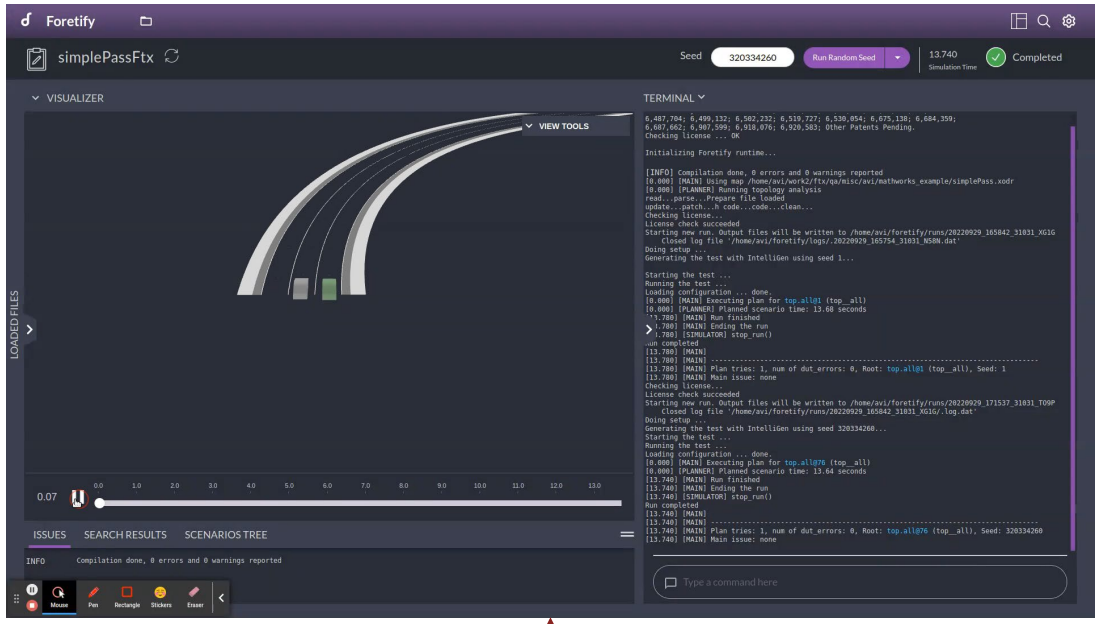
- ASAM OpenSCENARIO® V2.0
- OSC 1.x vs OSC 2.0
- Inter-tool Compatibility
- Recommendation
- Migration & Roadmap

Inter-tool Compatibility – with Foretify

RoadRunner Scenario

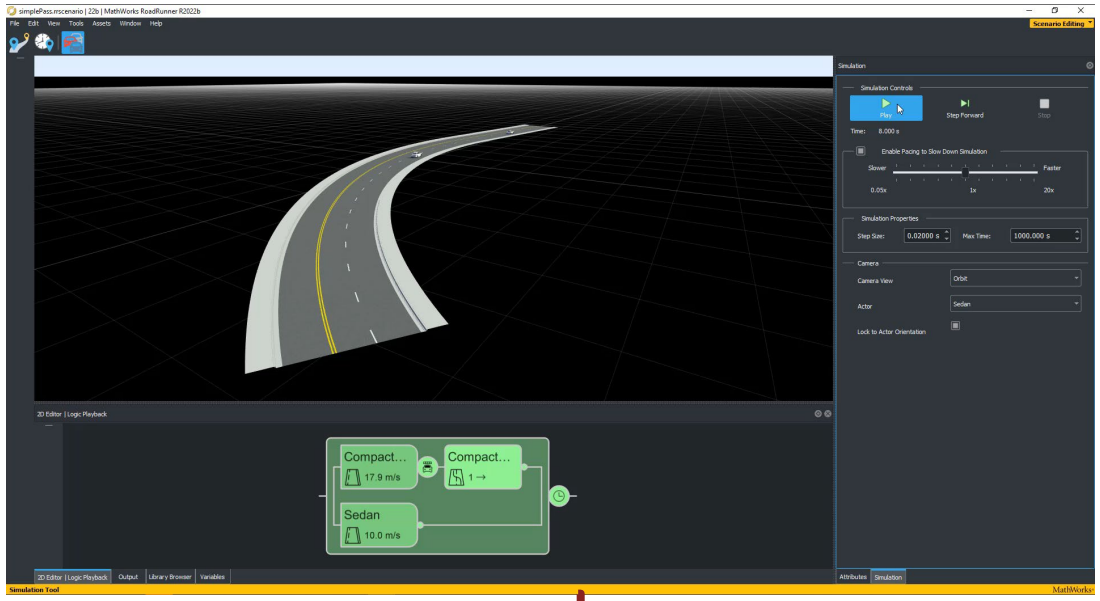


Foretify



ASAM OSC 2.0

Inter-tool Compatibility – with 51World



ASAM
OSC 2.0

Inter-tool Compatibility – Challenges/Puzzles

- Usage of modifiers/actions across tools is different
- How to handle customization?
- Which type of scenario are people giving priority to
  - More concrete or More Abstract?
  - Is everyone looking for logical or pure concrete?

Agenda

- ASAM OpenSCENARIO® V2.0
- OSC 1.x vs OSC 2.0
- Inter-tool Compatibility
- Recommendation
- Migration & Roadmap

Recommendation

- Address the problems faced by OSC1.x format in OSC 2.0
  - OSC 1.x is still not ready for inter-tool compatibility
  - Everyone is struggling with structures
  - Most data is either bad or overcomplicated

- Close the gaps between OSC1.x and OSC 2.0

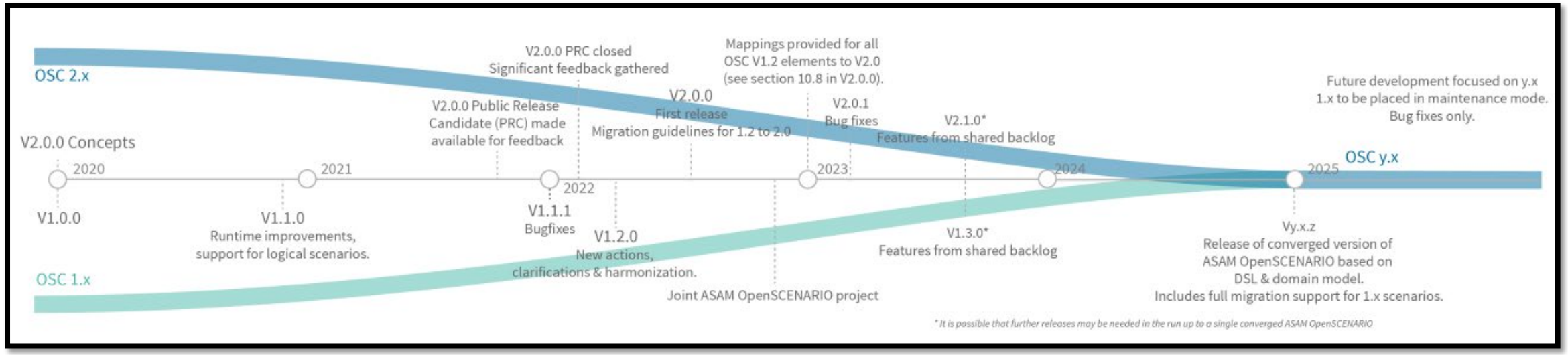
- Support for domains beyond Automotive

Call to action: Re-initiate implementer's forum
It would be good to collaborate and discuss across all tool vendors and understand the usage of the standard and address challenges.

Agenda

- ASAM OpenSCENARIO® V2.0
- OSC 1.x vs OSC 2.0
- Inter-tool Compatibility
- Recommendation
- Migration & Roadmap

Migration & Roadmap



Updated Roadmap for OSC

Migration & Roadmap

P_2022_06 ASAM OpenSCENARIO

PROJECT NUMBER	P_2022_06
PROJECT TYPE	Minor Version Dev.
DOMAIN	Simulation
PROPOSAL WORKSHOP	Nov 08, 2022 (Register)
ENROLL BY	tbd
PROJECT START	Dec 2022
PROJECT END	Sep 2023
RELEASE	Nov 2023

This proposal is for a joint ASAM OpenSCENARIO 1.x and 2.x project and the first to follow ASAM's roadmap for OpenSCENARIO. The proposal suggests launching a minor project, which will focus on the further development of ASAM OpenSCENARIO 1.x and 2.0 as well as on their migration and future steps.

LEARN MORE

Example: Time to Collision definition

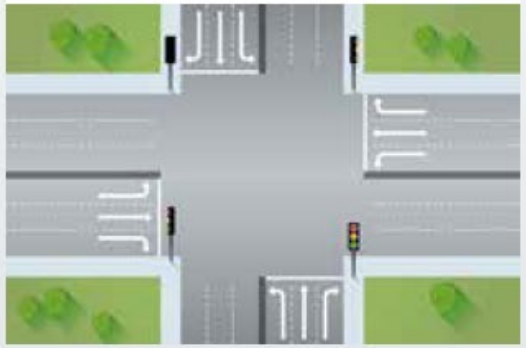
OSC2.0 - Time that is left until a possible collision between a traffic_participant and a reference physical_object takes place.

OSC 1.X - Time to collision is calculated as the distance divided by the relative speed.

MathWorks Support for ASAM OpenX Standards

Static Road Network Desription

ASAM OpenDRIVE®

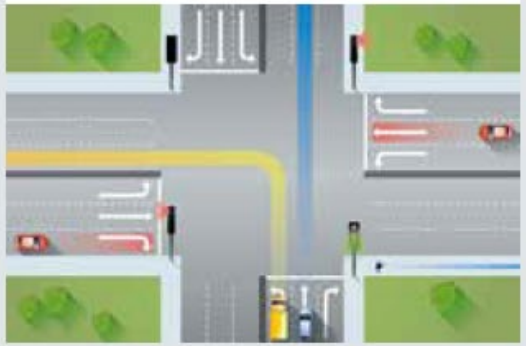


ASAM OpenCRG®

Static Road Surface Desription

Dynamic Scenario Desription

ASAM OpenSCENARIO®



ASAM OSI®

Interface for Simulation

Thank you