

Generate ASAM OpenSCENARIO from recorded sensor data



Seo-Wook Park
Principal Application Engineer for ADAS/Automated Driving
MathWorks

Sept. 19, 2023



Contents

- Why generate virtual scenarios from recorded data?
- What is a common workflow to generate virtual scenarios?
- How does the RoadRunner Scenario implement the ASAM OpenSCENARIO?
- Interoperability using ASAM OpenSCENARIO 1.x

Contents

- Why generate virtual scenarios from recorded data?
- What is a common workflow to generate virtual scenarios?
- How does the RoadRunner Scenario implement the ASAM OpenSCENARIO?
- Interoperability using ASAM OpenSCENARIO 1.x

Why generate virtual scenarios from recorded data?

Why ASAM OpenSCENARIO?

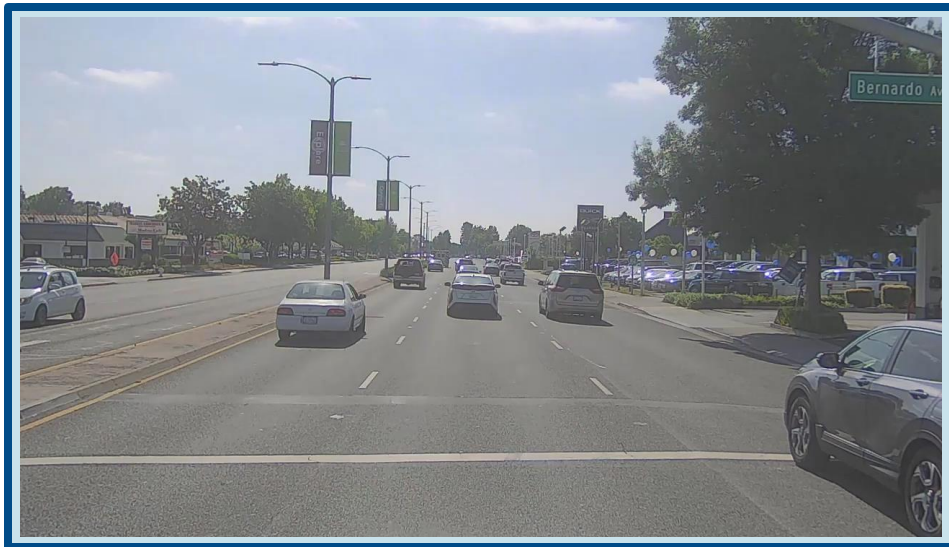
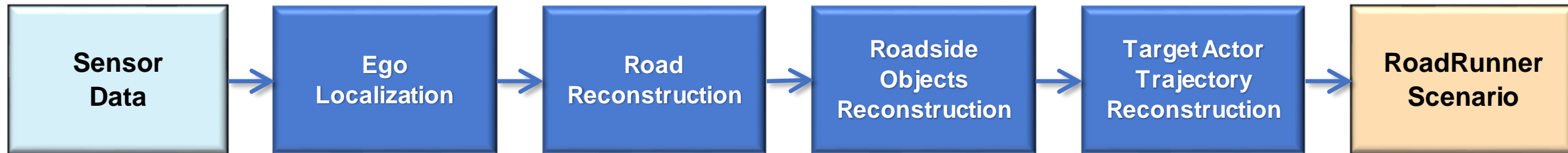
- Challenges
 - Big data size from “tens of thousands of miles” test drive
 - Not easy to reproduce a real-world traffic situation with closed-loop simulation
- Virtual driving scenarios
 - Reduce development time
 - Enable closed-loop simulation to identify the root causes for unwanted system behavior
 - Run the regression testing of ADAS/AD algorithms
- Why ASAM OpenSCENARIO?
 - Exchange scenarios for multiple tools (interoperability)
 - Create scenario catalogs for validating ADAS/AD algorithms.



Contents

- Why generate virtual scenarios from recorded data?
- **What is a common workflow to generate virtual scenarios?**
- How does the RoadRunner Scenario implement the ASAM OpenSCENARIO?
- Interoperability using ASAM OpenSCENARIO 1.x

What is a common workflow to generate virtual scenarios?



Sensor data



RoadRunner Scenario



Camera



Lanes



Radar



Lidar



Lanes

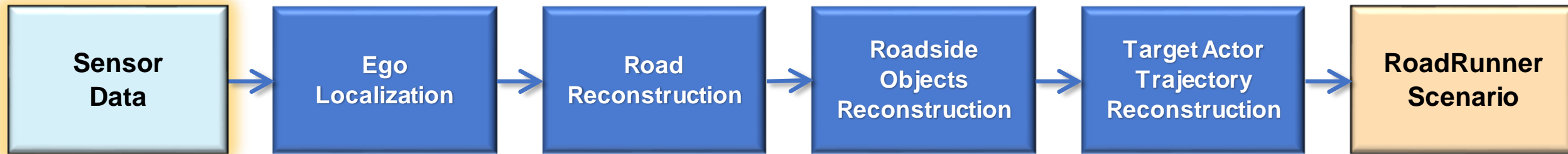


Vehicles

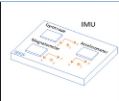


Trees, Buildings, Poles

Ego localization



GPS



IMU

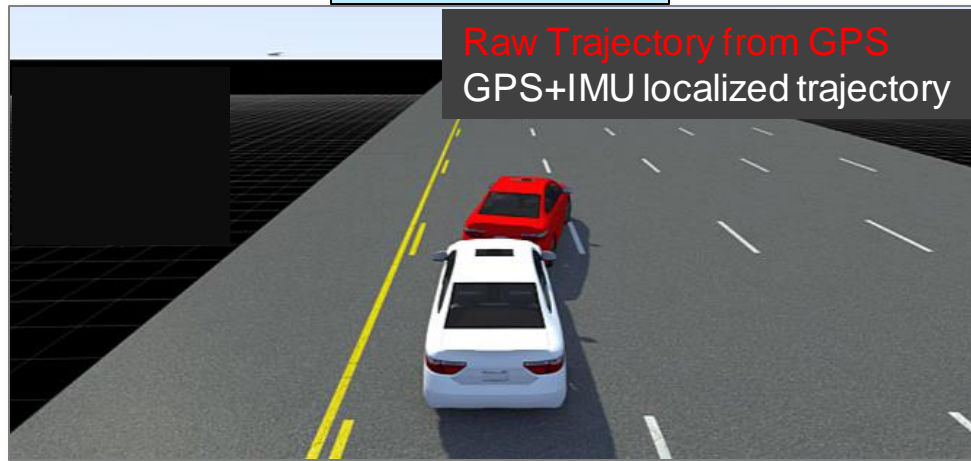


Lanes



HD Map

Pose correction



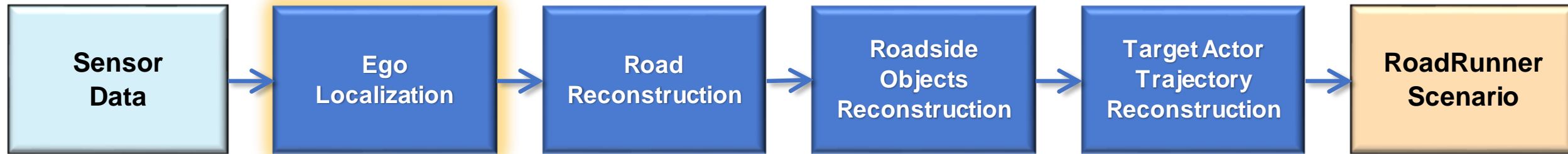
- Correct position and orientation of ego actor using GPS and IMU fusion

Offset correction



- Correct single/multi-lane level offsets using GPS, lane information and HD maps

Road reconstruction

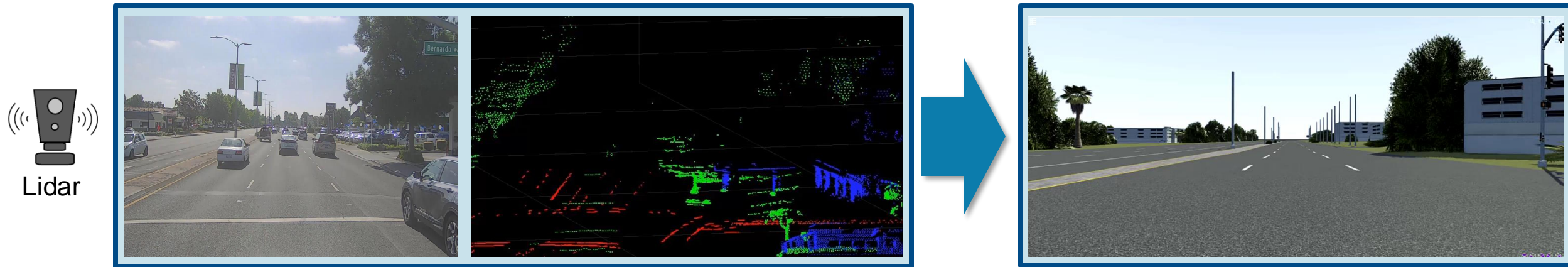
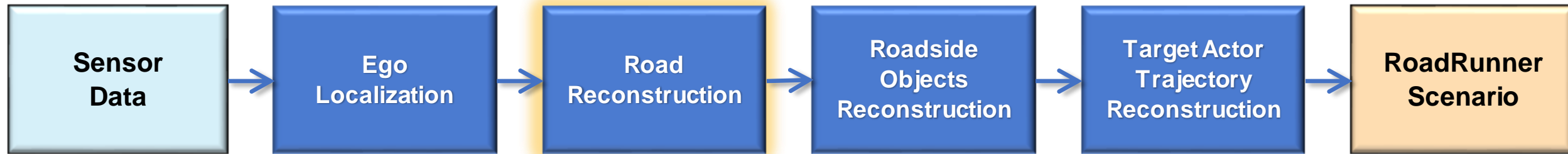


- Extract lanes, road boundaries from camera and lidar data



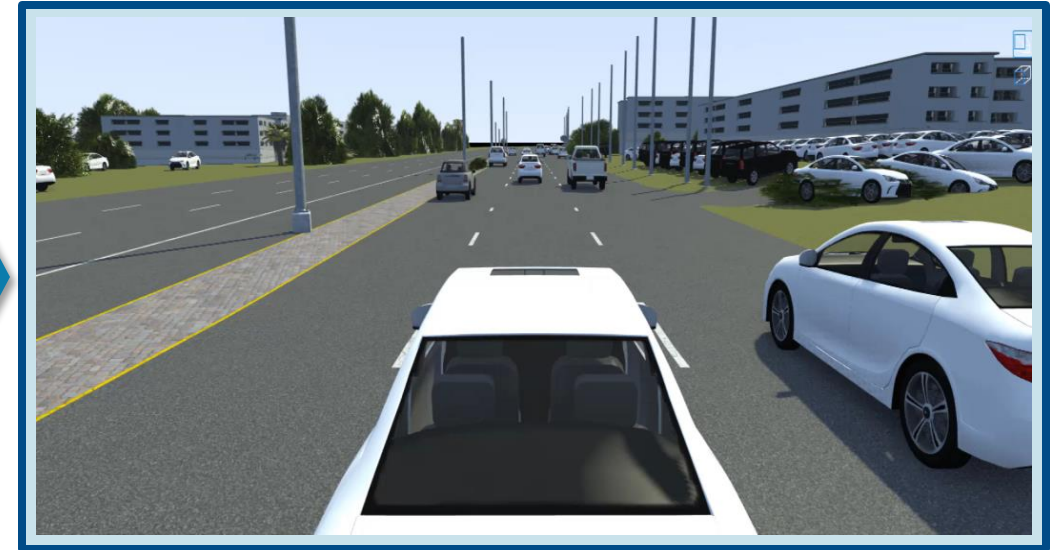
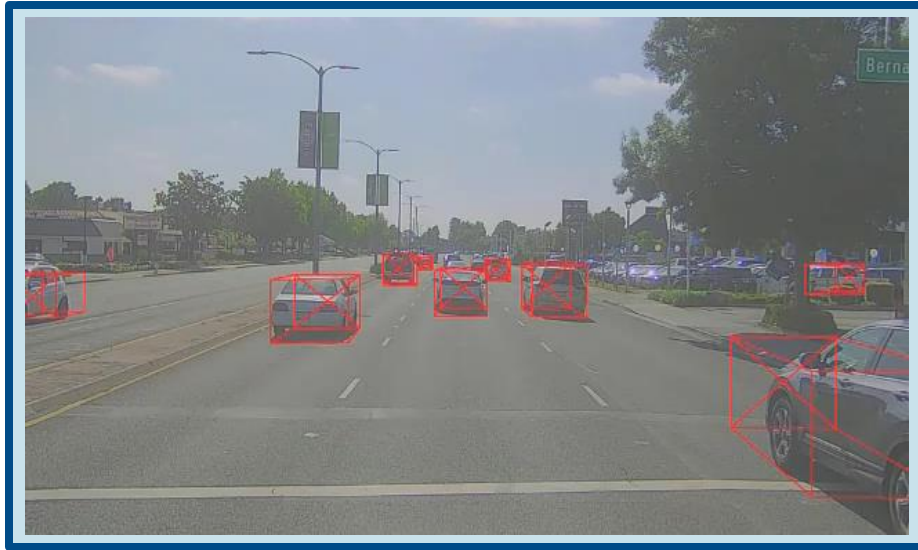
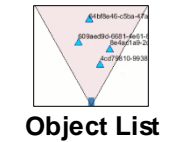
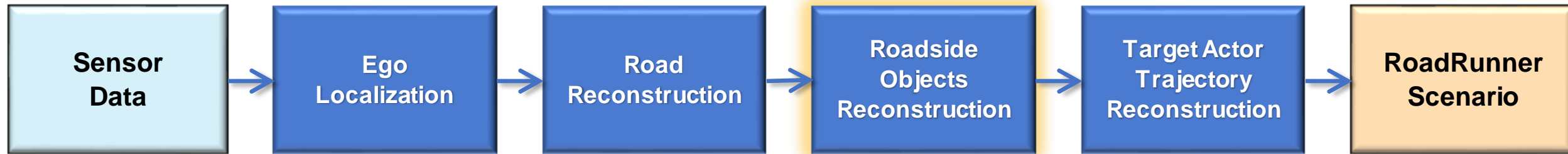
- Reconstruct road with lane add/drop, road curvature and junctions

Roadside objects reconstruction



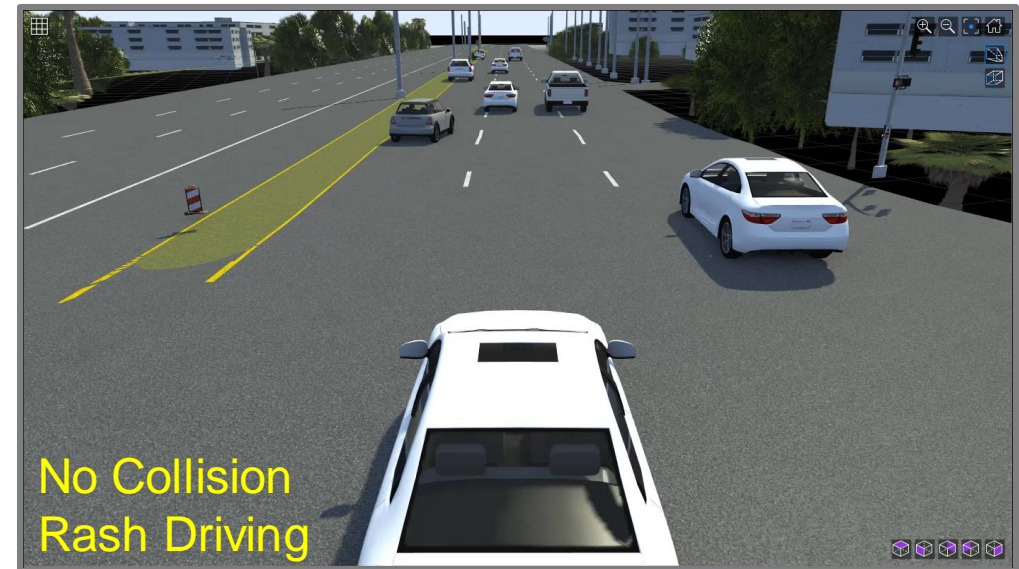
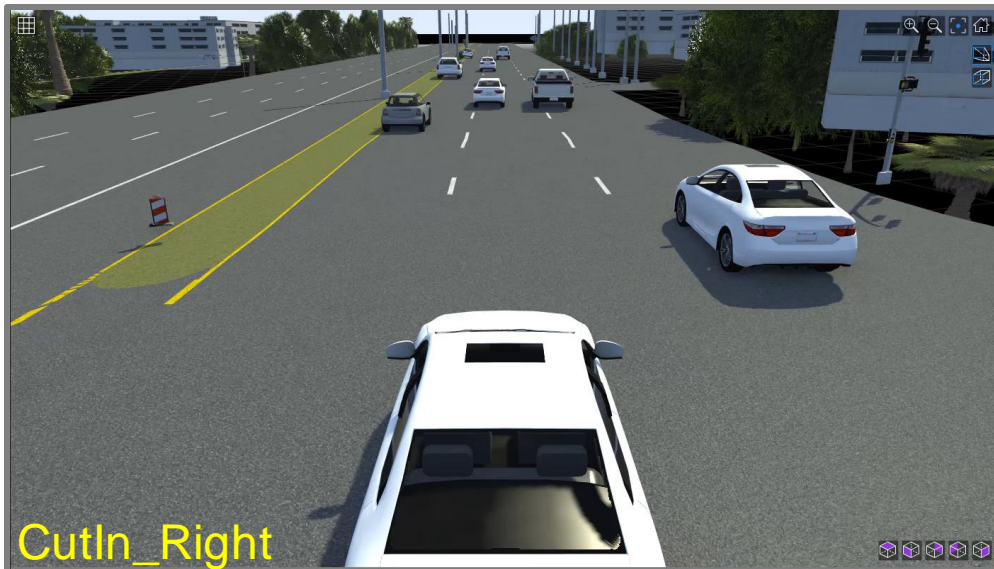
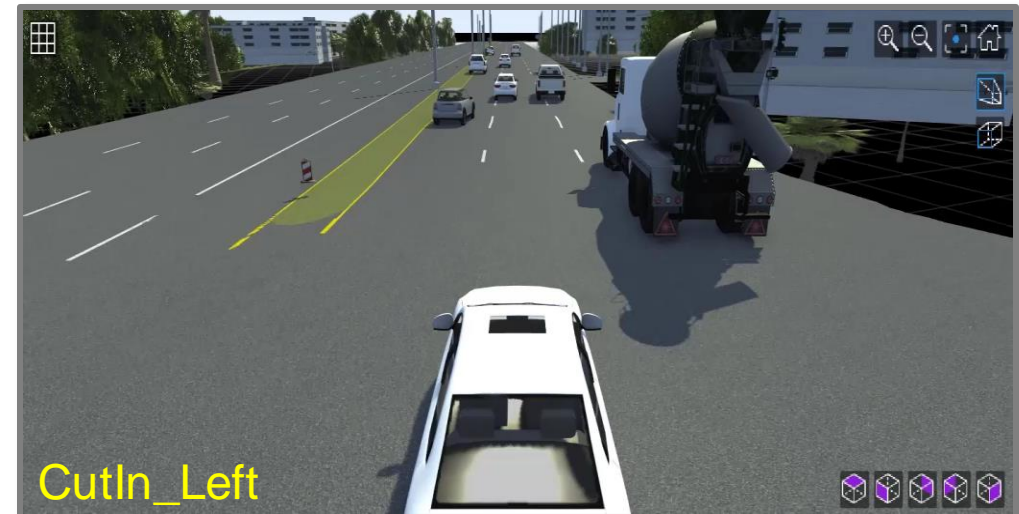
- Labelled Lidar data is used to reconstruct trees, buildings and other roadside objects.
 - Labels supported: buildings, trees, bushes, traffic cones, pylons, barricades, and electric poles
- You can also extract trees and buildings from raw lidar data in absence of labels.
- Alternatively use Camera + GPS to get approximate scene with roadside objects.

Target actor trajectory reconstruction

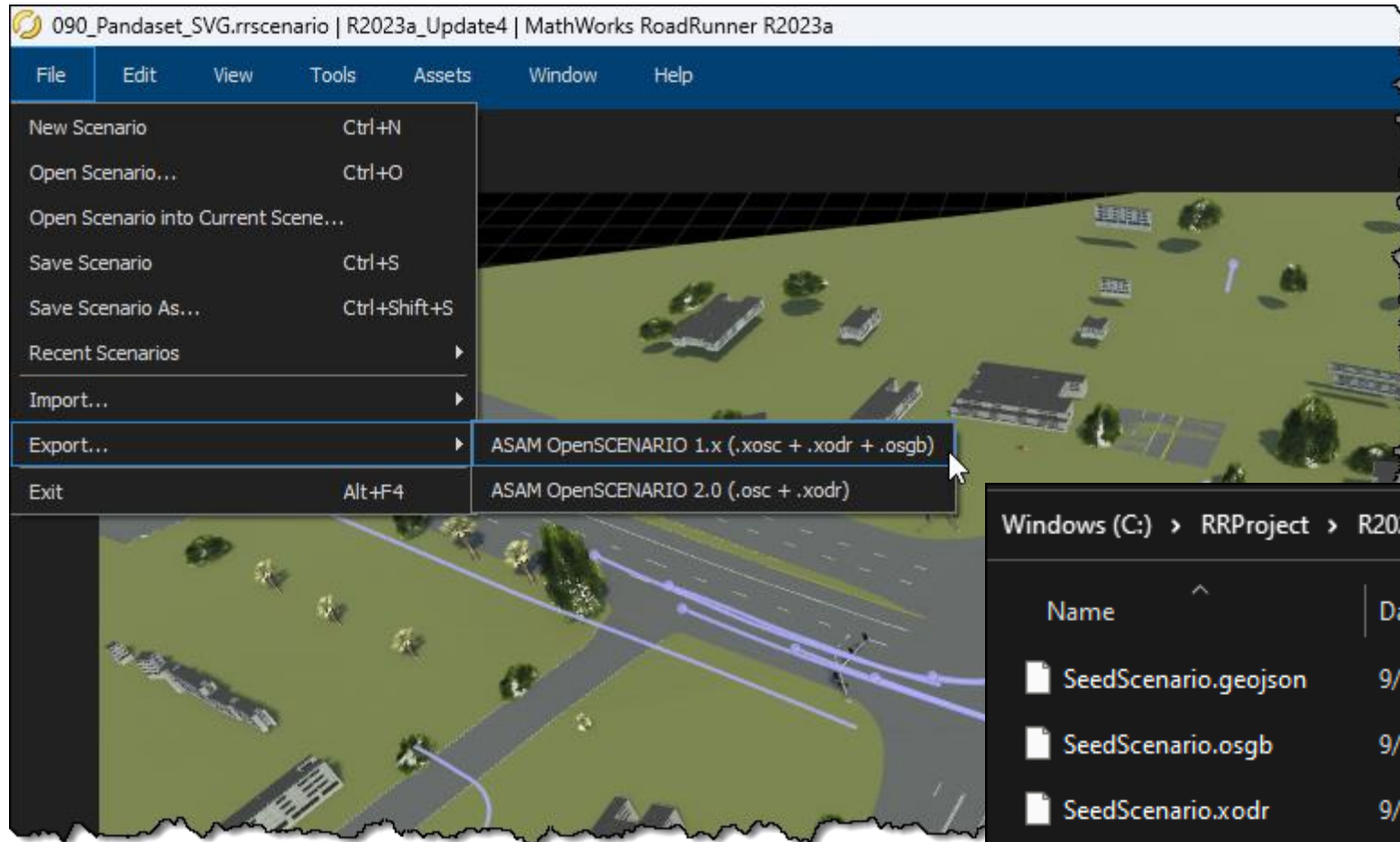


- Reconstruct dynamic actor tracklists, vehicles from camera or lidar or radar data and its combinations
- Lidar sensor data can enable extraction of objects from all the sides of the ego vehicle whereas Radar sensor data can enable farther objects.
- Camera sensor data can help identify object classes (car, truck etc.)

Variant scenarios



Export to ASAM OpenSCENARIO 1.x



Windows (C:) > RRProject > R2023a_Update4 > Exports

Name	Date modified	Type	Size
SeedScenario.geojson	9/6/2023 5:00 PM	GEOJSON File	2,391 KB
SeedScenario.osgb	9/6/2023 5:01 PM	OSGB File	204,204 KB
SeedScenario.xodr	9/6/2023 5:00 PM	XODR File	1,475 KB
SeedScenario.xosc	9/6/2023 5:01 PM	XOSC File	884 KB

Export to ASAM OpenSCENARIO 1.x and 2.0

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <OpenSCENARIO>
3   <FileHeader revMajor="1" revMinor="0" date="2023-09-06T17:01:30" description="Exported from RoadRunner versionR20
4   <ParameterDeclarations/>
5   <CatalogLocations/>
6   <RoadNetwork>
7     <LogicFile filepath="SeedScenario.xodr"/>
8     <SceneGraphFile filepath="SeedScenario.osgb"/>
9   </RoadNetwork>
10  <Entities>
402  <Storyboard>
403    <Init>
404      <Actions>
9614    </Init>
9615    <Story name="EmptyStory">
9616      <Act name="EmptyAct">
9617        <ManeuverGroup name="EmptyGroup" maximumExecut
9618          <Actors selectTriggeringEntities="false"/>
9619        </ManeuverGroup>
9620        <StartTrigger>
9621          <ConditionGroup>
9622            <Condition name="SimulationStart" cond
9623              <ByValueCondition>
9624                <SimulationTimeCondition value
9625              </ByValueCondition>
9626            </Condition>
9627          </ConditionGroup>
9628        </StartTrigger>
9629        <StopTrigger/>
9630      </Act>
9631    </Story>
9632    <StopTrigger>
9633      <ConditionGroup>
9634        <Condition name="" conditionEdge="none" delay=
          <ByValueCondition>
```

```
# Vendor="MathWorks" Program="RoadRunner" Version="R2023a Update 4 (1.6.4.b4405703230

import osc.standard

# Action Declarations
action mw_assign_time_data_based_speed

scenario pandaset_osc2:

map: map
keep(map.map_file == "Pandaset_OSC2.xodr")

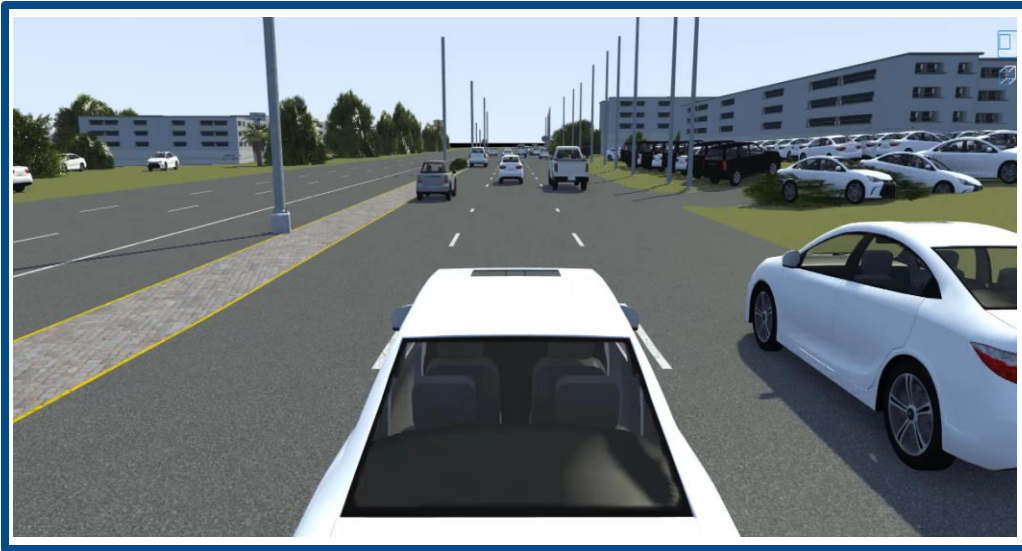
# Scenario Variables
var sim_start_time: time = sample(environment.datetime, @root_phase.start)

# Scenario Parameters
environment: environment

ego: vehicle with:
  keep(it.color == white)
  keep(it.geometry_reference == "Vehicles/Sedan.fbx")
  keep(it.bounding_box.center.x == -5.96046e-08m)
  keep(it.bounding_box.center.y == -0.00706172m)
  keep(it.bounding_box.center.z == 0.727468m)
  keep(it.bounding_box.length == 4.55561m)
  keep(it.bounding_box.width == 1.93778m)
  keep(it.bounding_box.height == 1.45382m)
  keep(it.center_of_gravity.x == -5.96046e-08m)
  keep(it.center_of_gravity.y == -0.00706172m)
```

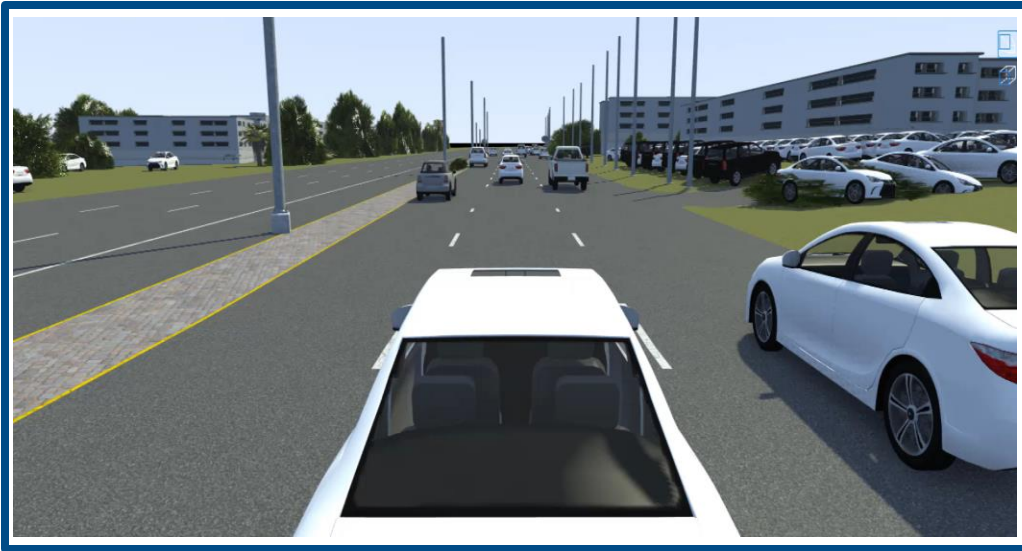

Review exported ASAM OpenSCENARIO 1.x with esmini

RoadRunner Scenario



RoadRunner Scenario vs. Carla

RoadRunner Scenario




Exports.xosc



Carla



Contents

- Why generate virtual scenarios from recorded data?
- What is a common workflow to generate virtual scenarios?
- How does the RoadRunner Scenario implement the ASAM OpenSCENARIO?
- Interoperability using ASAM OpenSCENARIO 1.x

Graphical interface for defining scenario logic

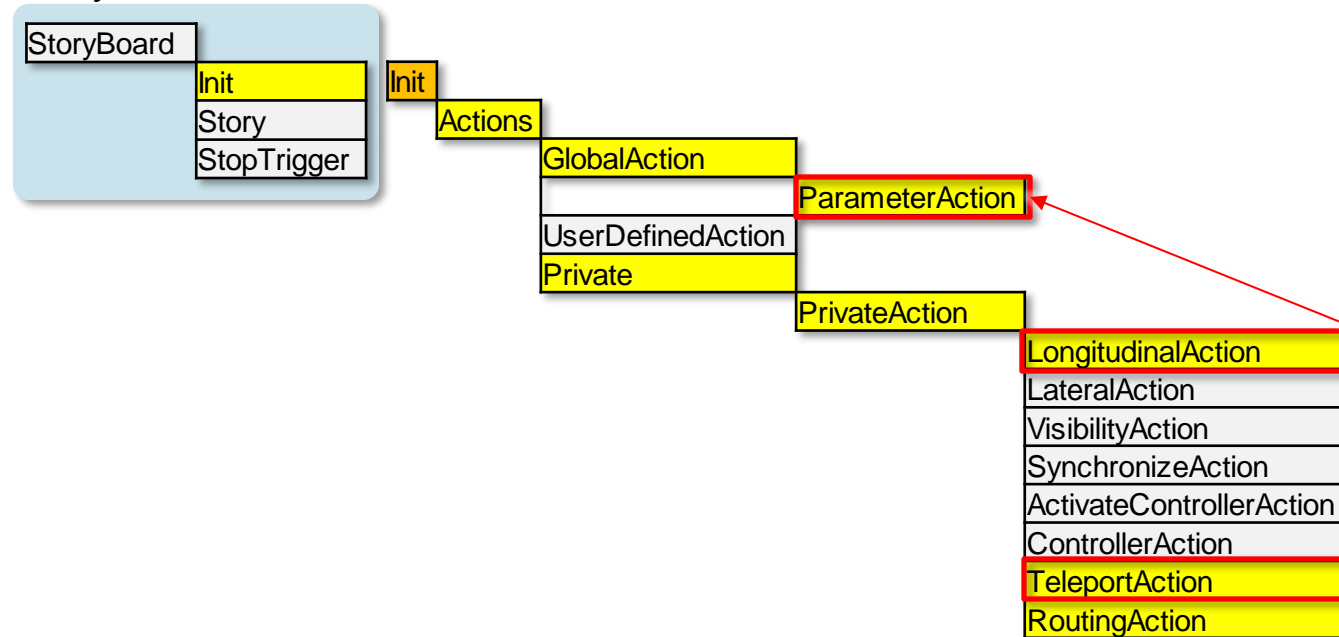
The screenshot displays the MathWorks RoadRunner R2023a graphical interface for defining scenario logic. The main window is titled "CutOutExampleTest.rsscenario | R2023a_Update4 | MathWorks RoadRunner R2023a". The interface is divided into several sections:

- 2D Editor | Logic:** This section shows a logic diagram with various nodes and connections. The nodes include:
 - Ego:** A red car icon with a speed of 10.0 m/s and a "LaneCha" label.
 - other_veh...:** Two nodes, each with a speed of 10.0 m/s and a "1 ← stati..." label.
 - stationary...:** Two nodes, each with a speed of 0.0 m/s and a "0.0 m/s" label.
 - Wait:** A node with a clock icon and a "CompactCar" label.
 - Sedan:** A node with a speed of 17.9 m/s.
 - CompactCar:** A node with a speed of 10.0 m/s.
- Library Browser:** This section displays a list of assets and vehicles. The "Vehicles" folder is expanded, showing a grid of vehicle icons including:
 - ADT Vehicles
 - Vehicle Textures
 - Ambulance
 - CementTruck
 - CompactCar
 - DeliveryVan
 - GarbageTruck
 - PickupTruck
 - SchoolBus
 - Sedan
 - SemiTruck
 - SemiTruck_railer01
- Scenario Editing:** This panel on the right contains attributes and actions for the selected scenario. It includes fields for:
 - Attributes:** Name, Actor (Ego), and Speed (10.00 m/s).
 - Actions:** A list of actions including "Initialize Speed", "Follow Lane", "Change Behavior Parameter", and "LaneChangeDistance (string)".

A yellow dashed box highlights the "2D Editor | Logic" section, and a yellow text overlay asks: "How does the scenario logic map to ASAM OpenSCENARIO?".

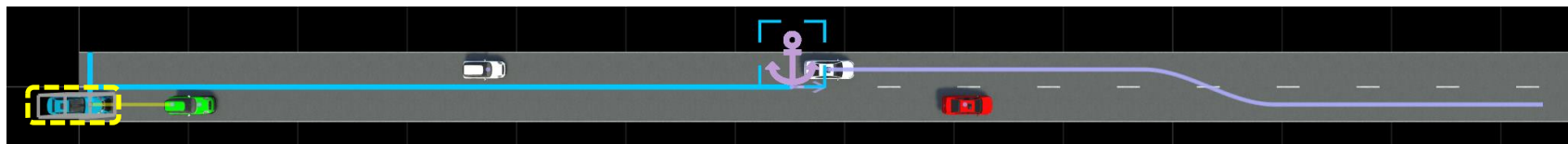
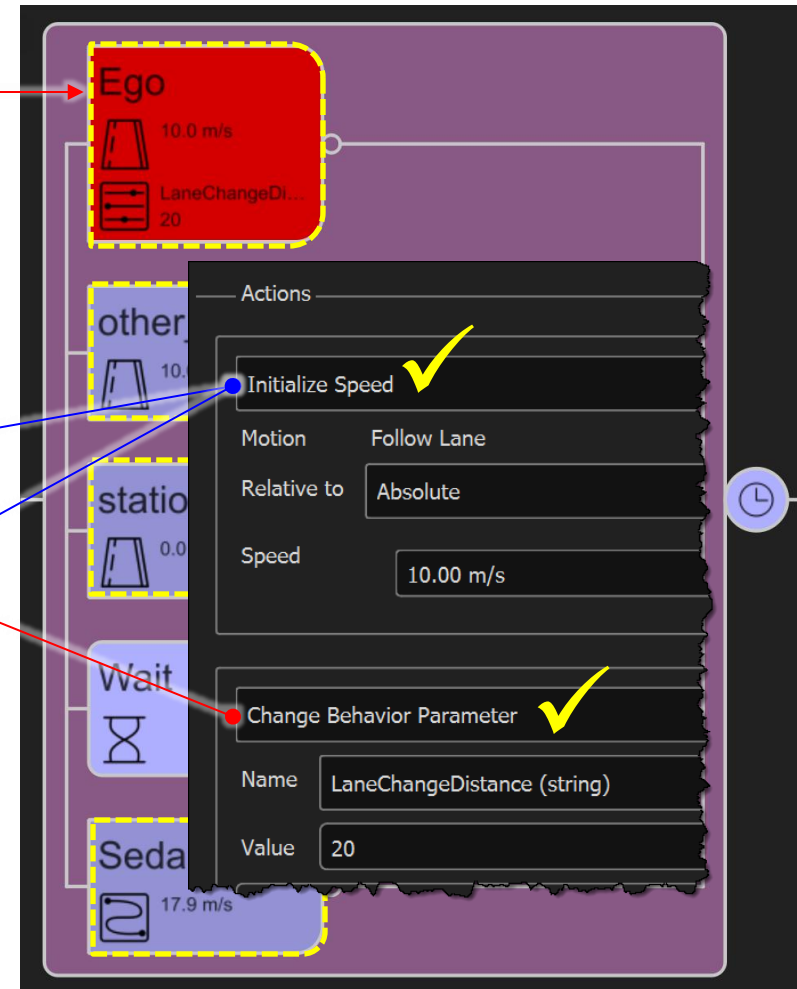
Initialization element

ASAM OpenSCENARIO 1.x
Storyboard



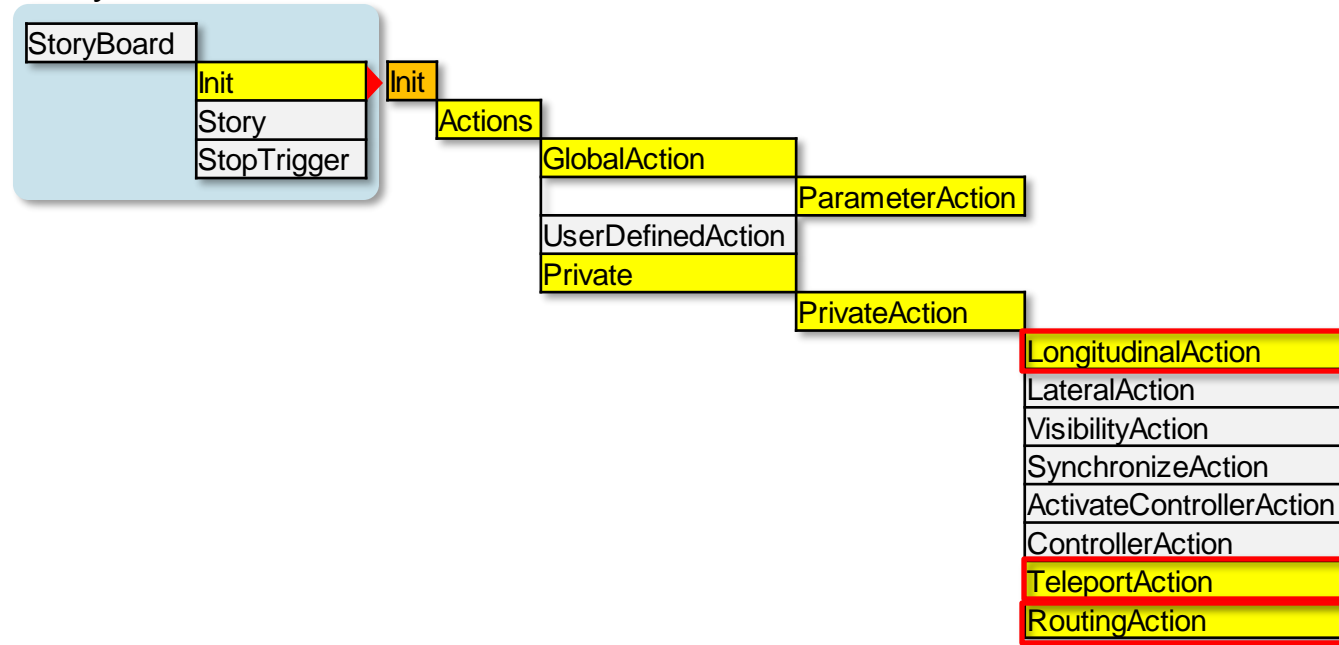
Initial
Action Phase

RoadRunner Scenario – Logic Editor



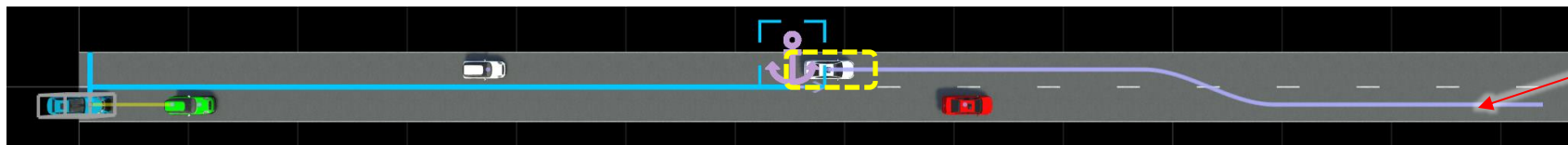
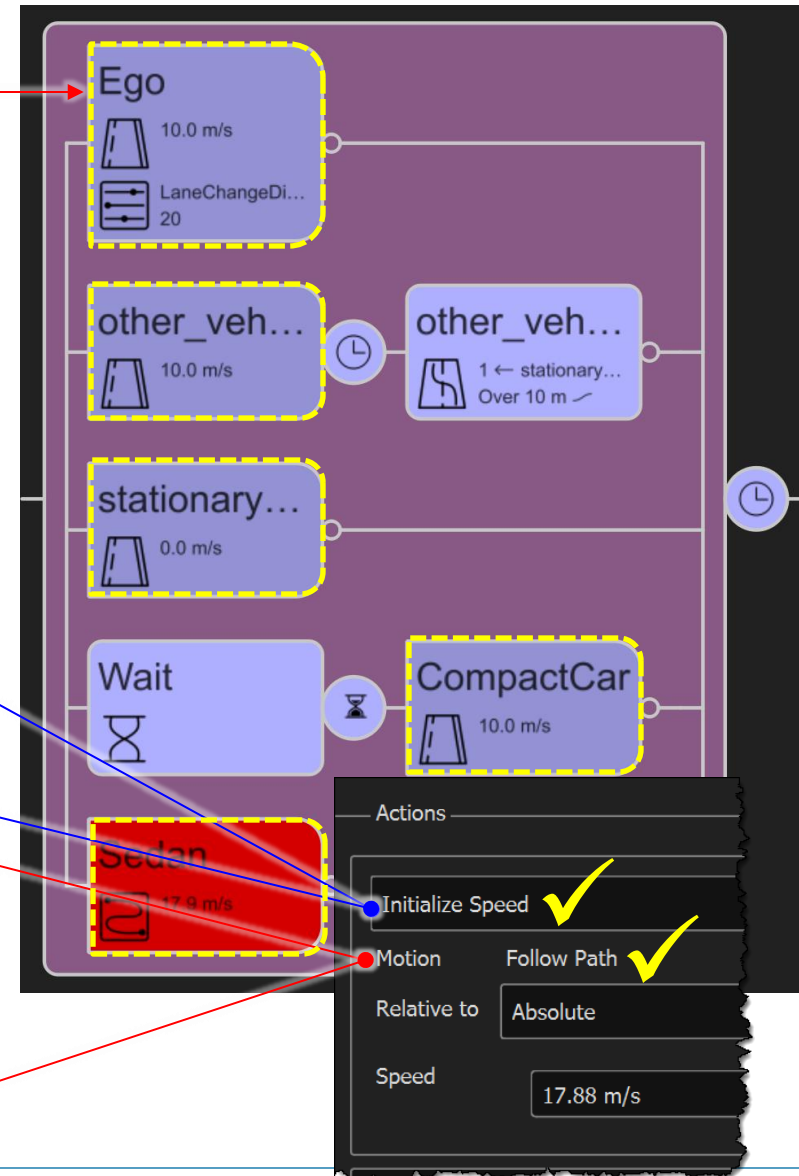
Initialization element

ASAM OpenSCENARIO 1.x
Storyboard



Initial
Action Phase

RoadRunner Scenario – Logic Editor



Story – Action element

ASAM OpenSCENARIO 1.x
Storyboard

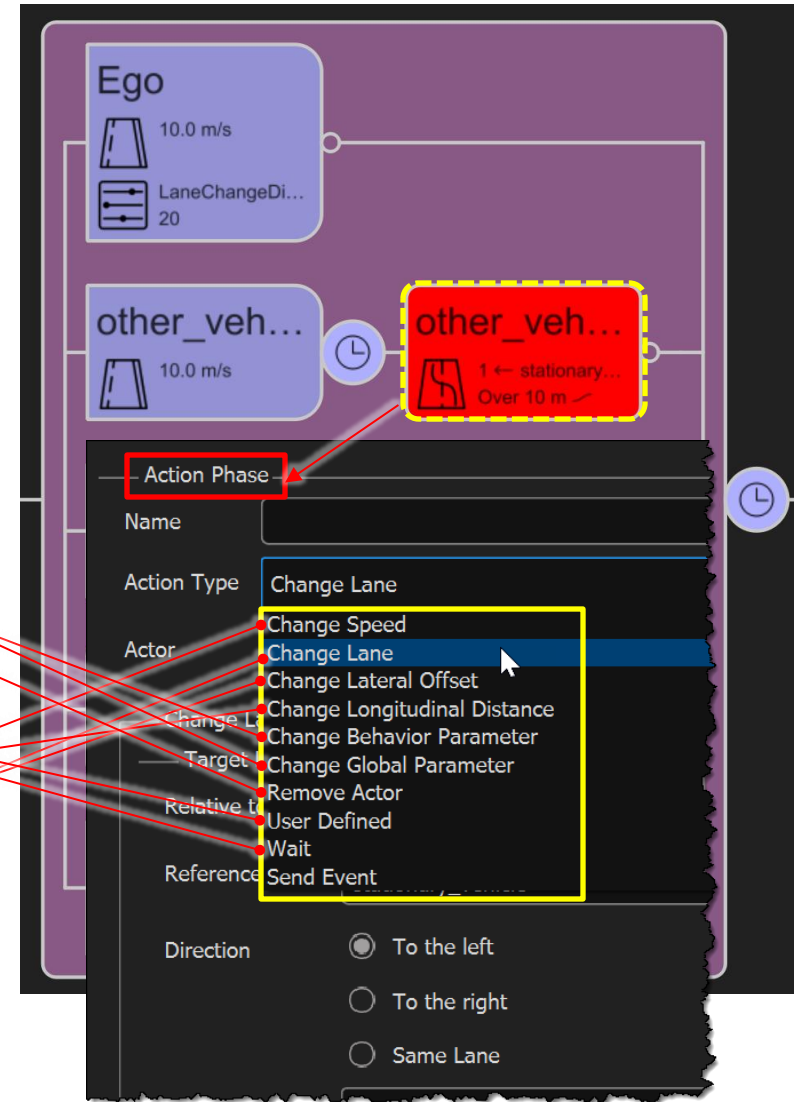
“who” is doing “what” and
“when” in a scenario.

“who”

“what”

“when”

RoadRunner Scenario – Logic Editor



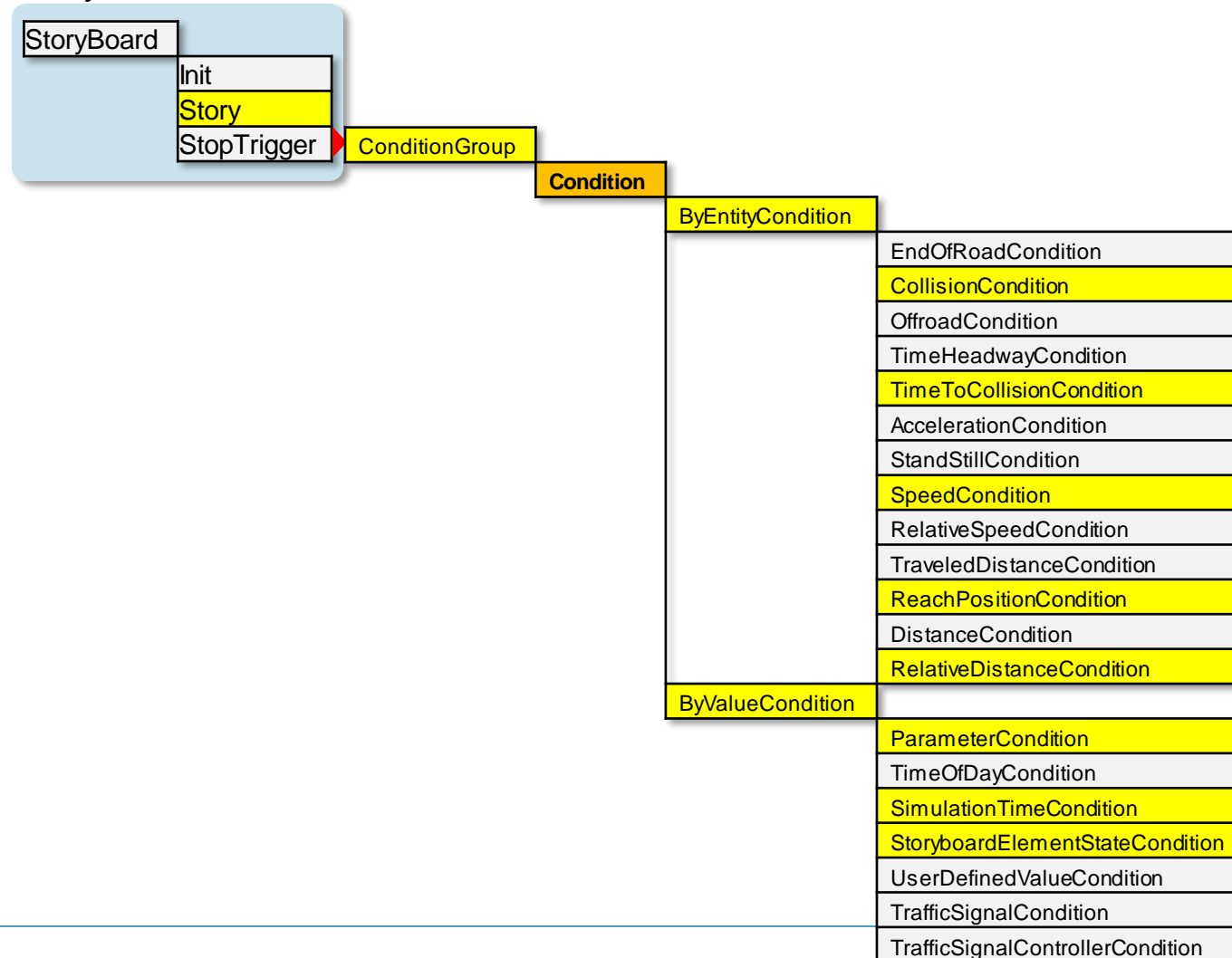
ASAM OpenSCENARIO 1.x Storyboard



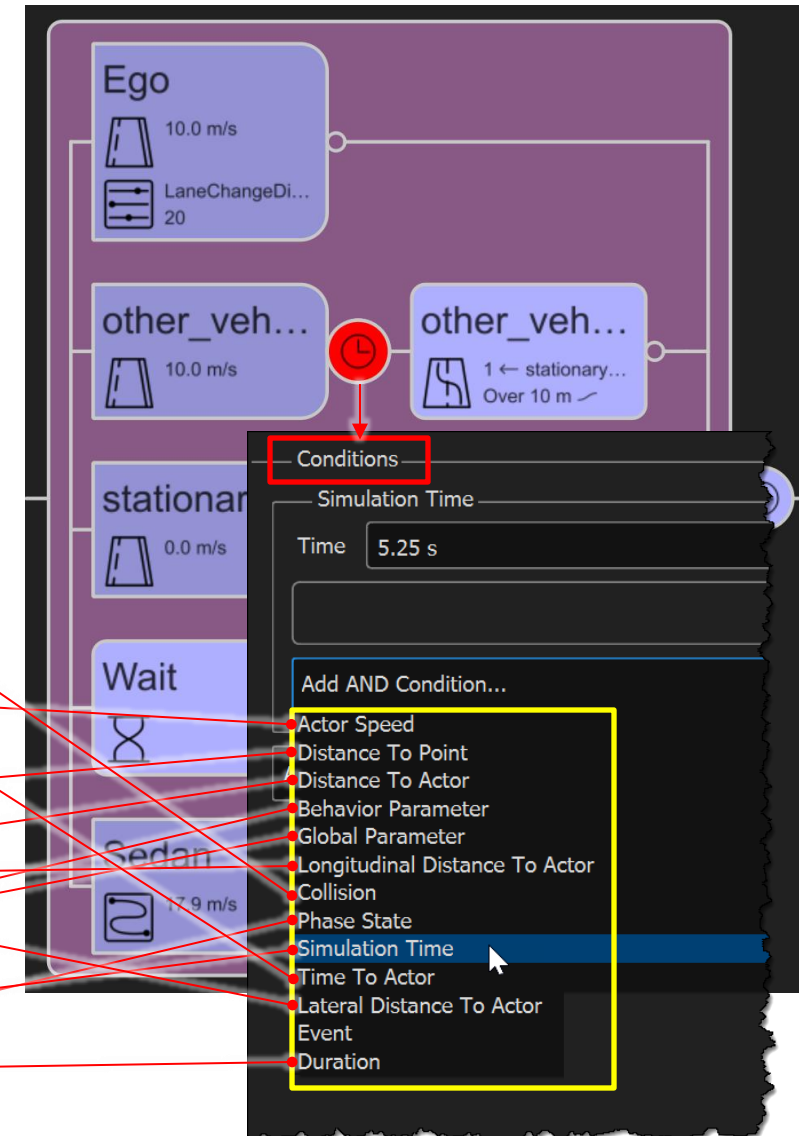
The screenshot displays the SCADE4 simulation environment. On the left, a state machine diagram is visible with several states: "Ego" (speed 10.0 m/s, LaneChangeDi... 20), "other_veh..." (speed 10.0 m/s), "stationar" (speed 0.0 m/s), "Wait" (hourglass icon), and "Sedan" (speed 17.9 m/s). A red clock icon is positioned between the "other_veh..." states. On the right, a configuration panel is open, showing the "Conditions" tab. The "Simulation Time" condition is selected, and the "Time" is set to 5.25 s. A yellow box highlights the "Add AND Condition..." button and the list of conditions, with a mouse cursor pointing to "Simulation Time".

Story – Trigger condition element

ASAM OpenSCENARIO 1.x
Storyboard



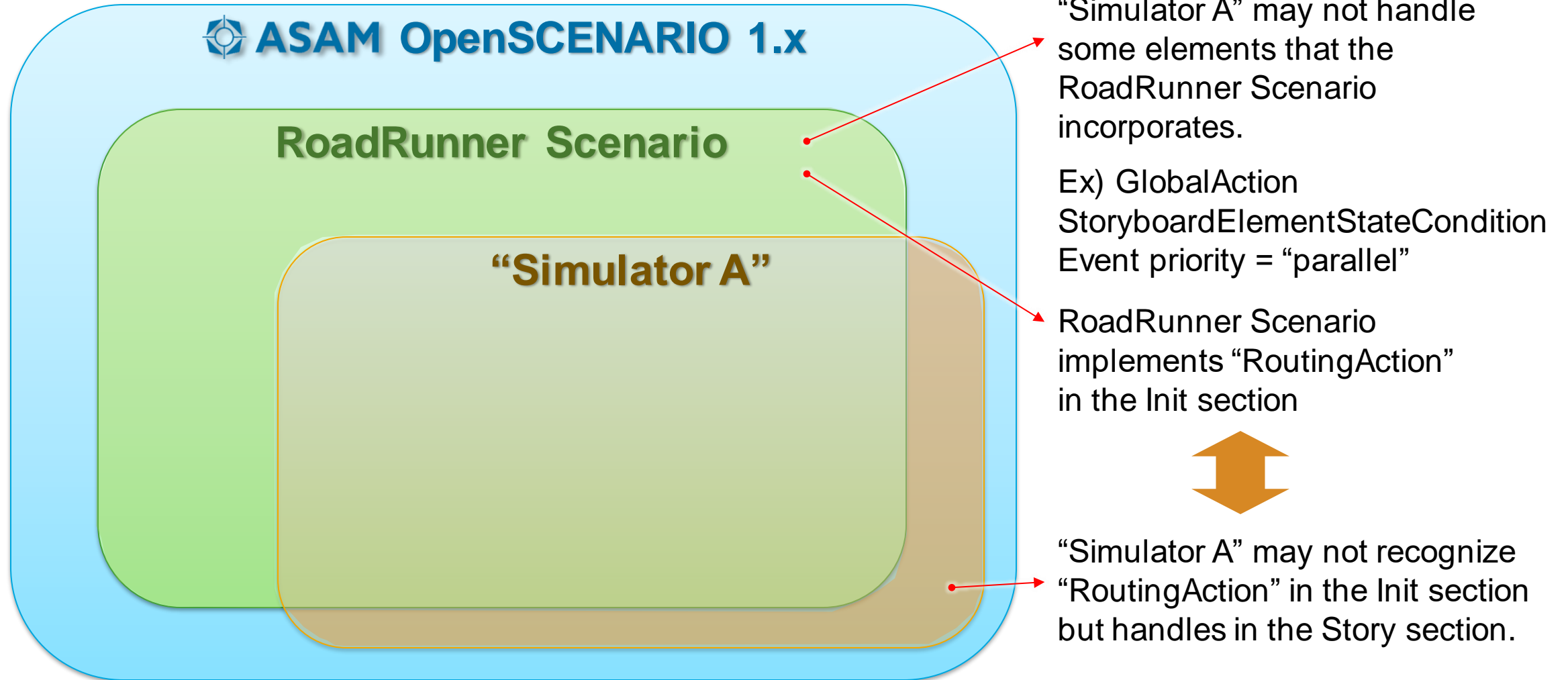
RoadRunner Scenario – Logic Editor



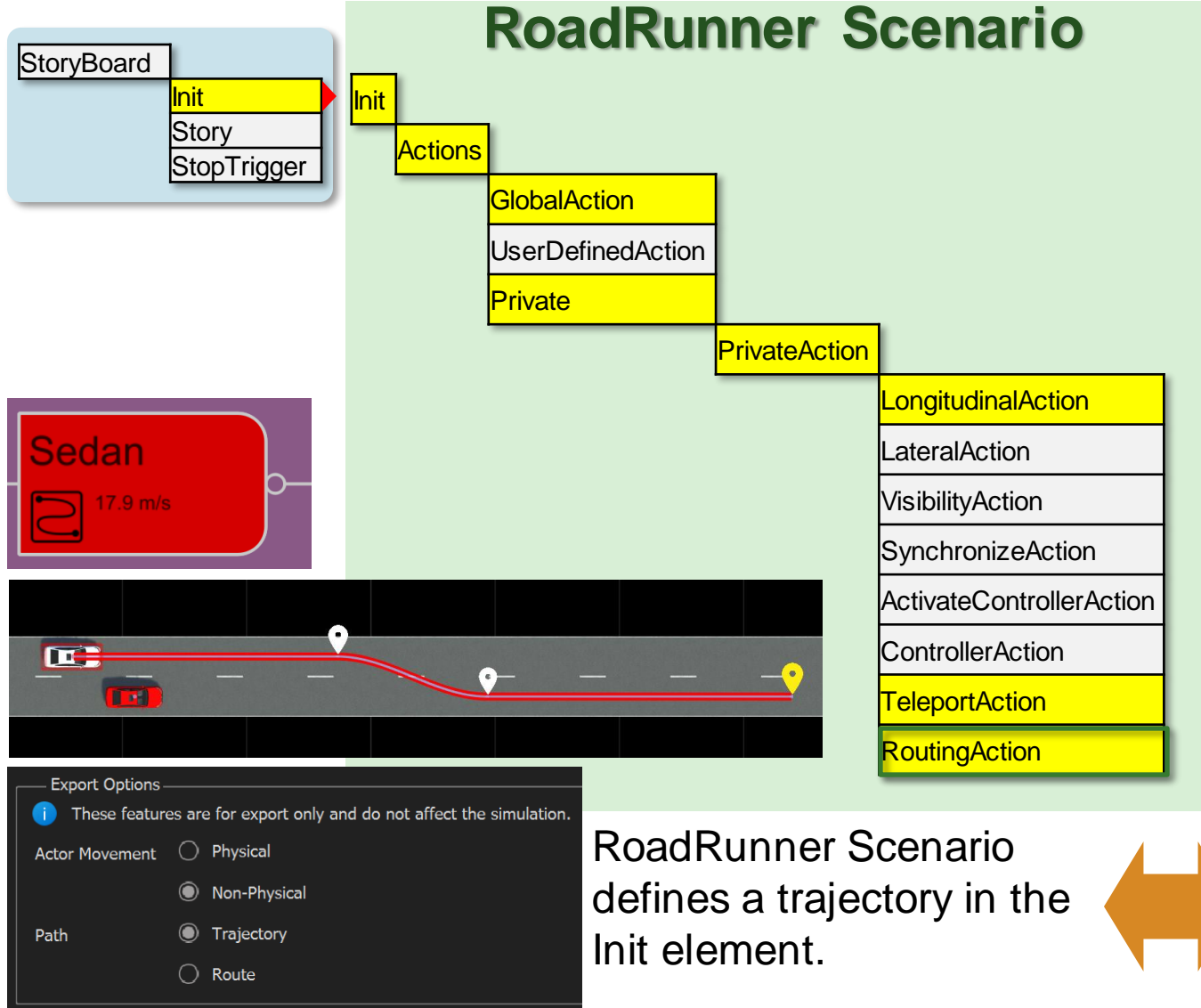
Contents

- Why generate virtual scenarios from recorded data?
- What is a common workflow to generate virtual scenarios?
- How does the RoadRunner Scenario implement the ASAM OpenSCENARIO?
- Interoperability using ASAM OpenSCENARIO 1.x

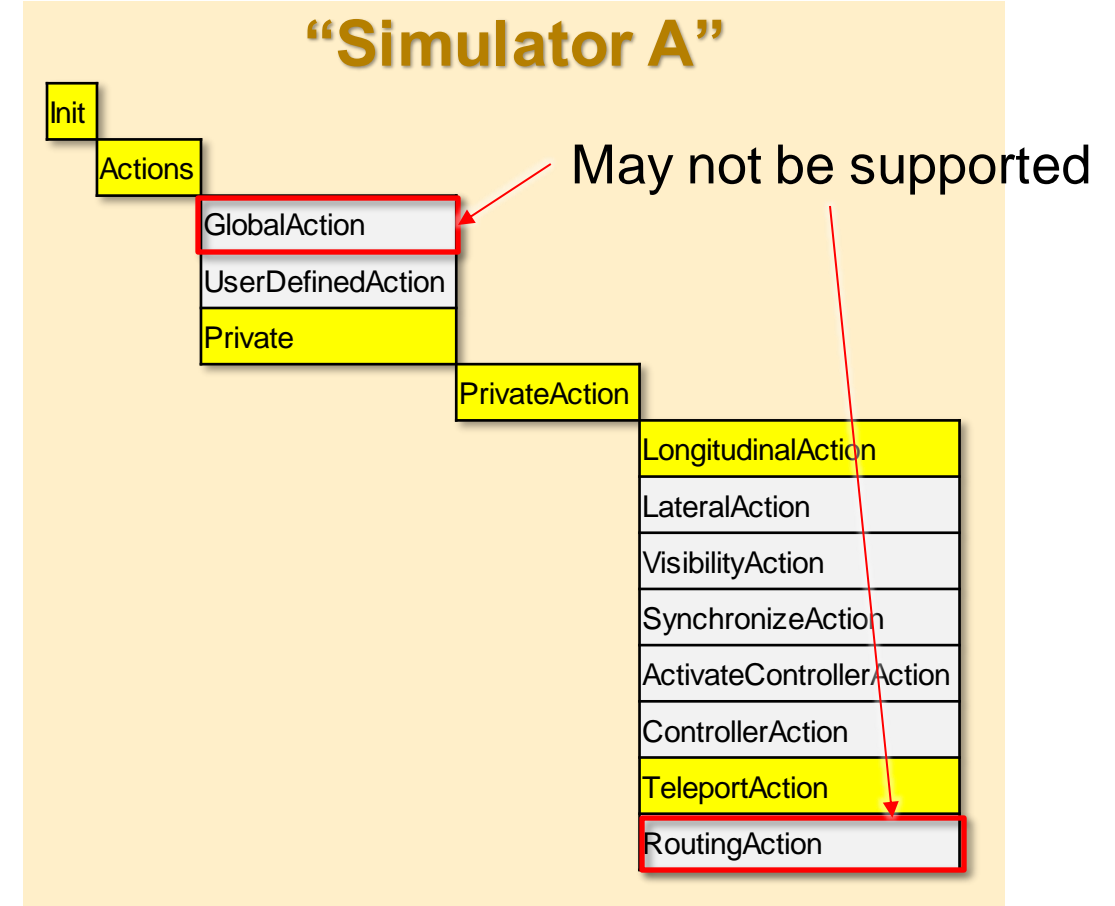
Interoperability using ASAM OpenSCENARIO 1.x



Init element – Interoperability example

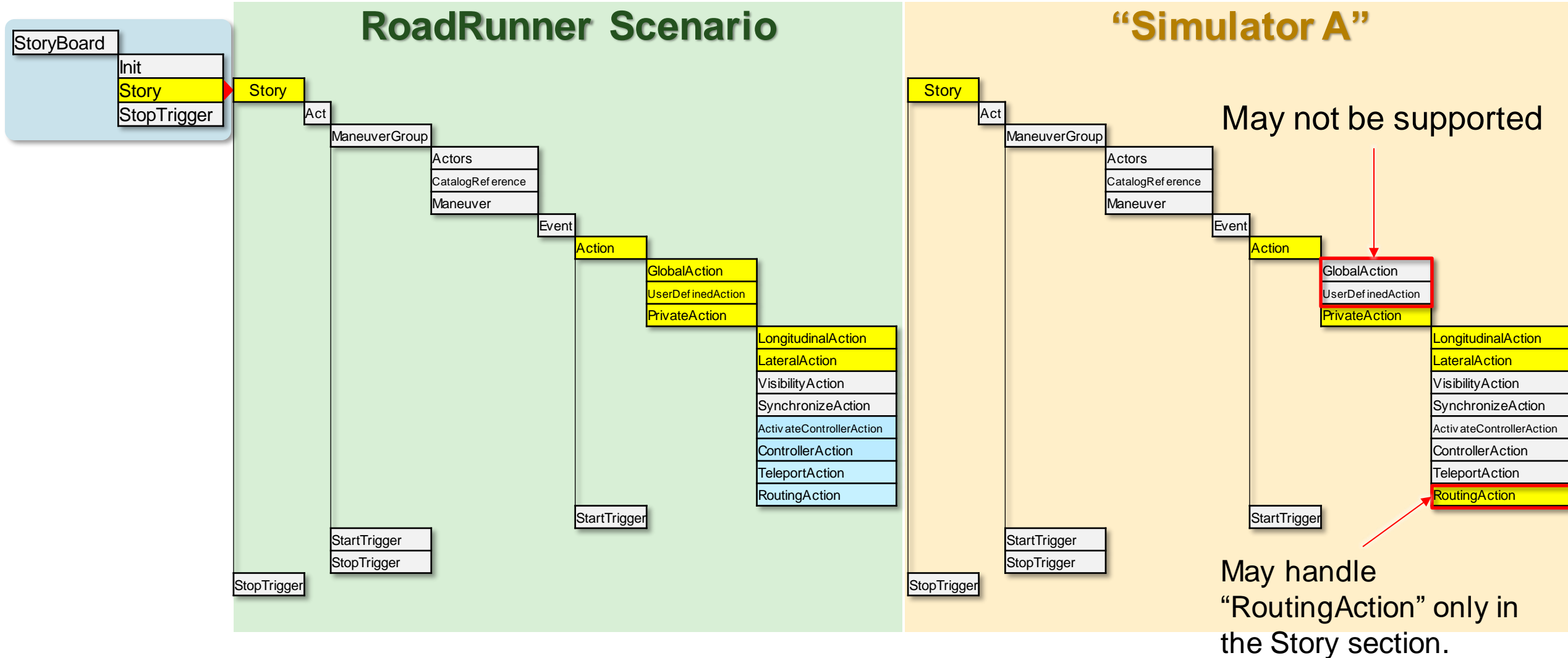


RoadRunner Scenario defines a trajectory in the Init element.

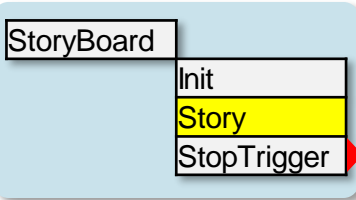


In “Simulator A”, trajectory may not be defined in the Init section. (Init is supposed only to set the initial conditions for the vehicles?)

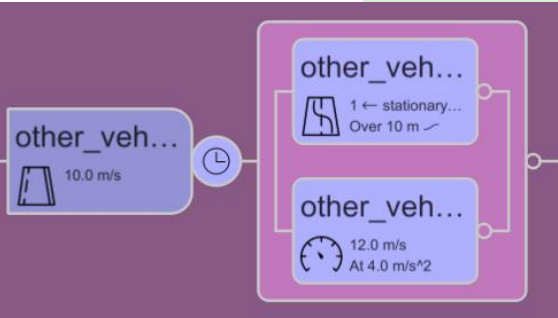
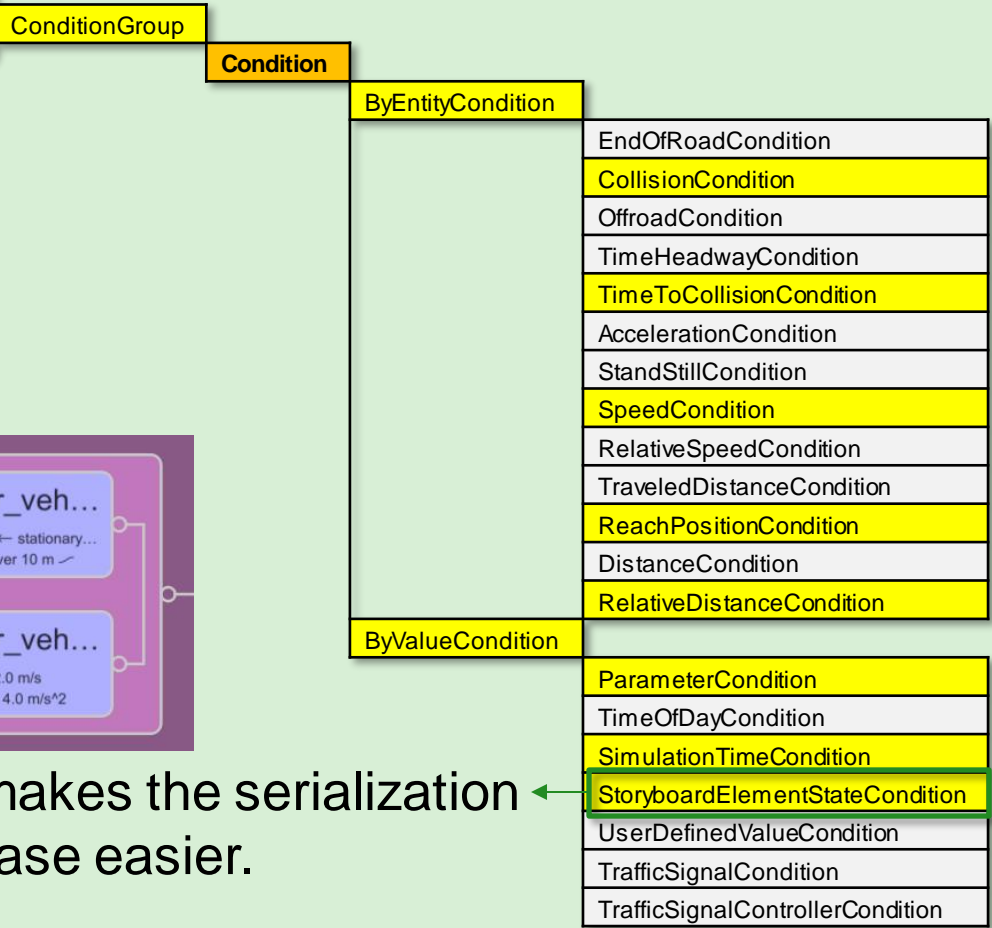
Story element – Interoperability example



Trigger condition element – Interoperability example

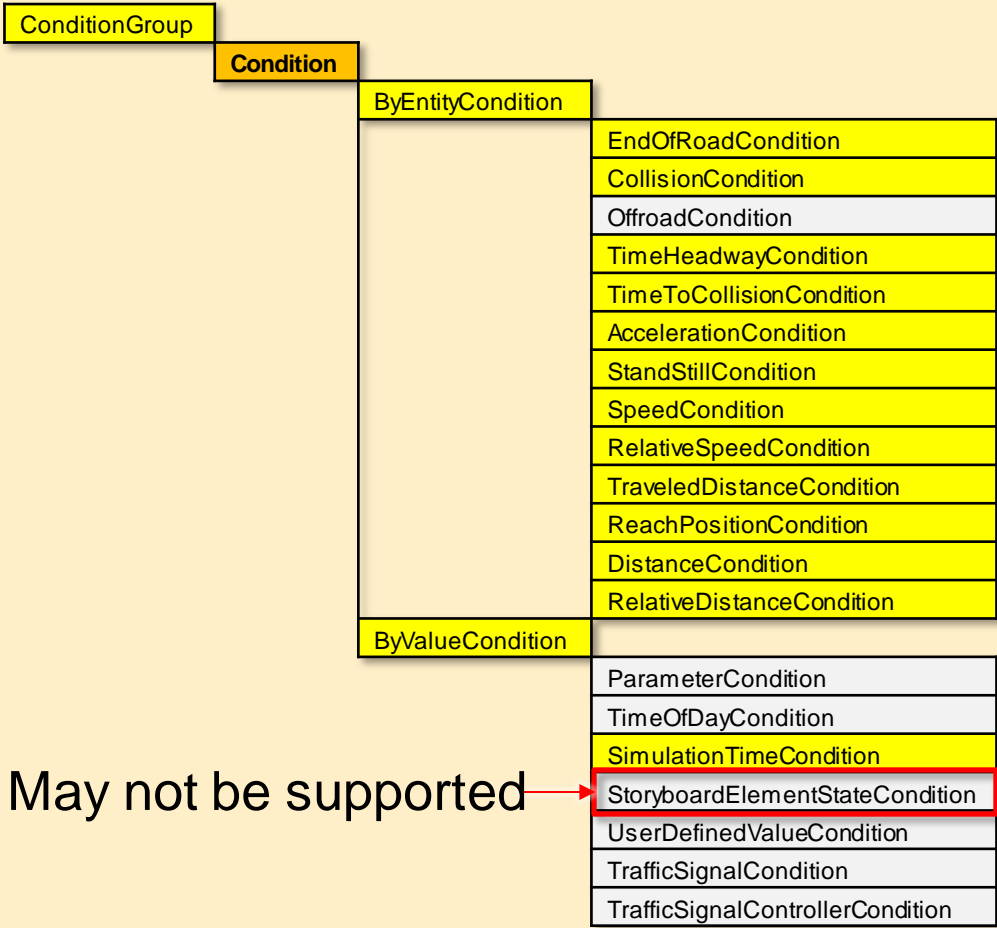


RoadRunner Scenario



This condition makes the serialization of the action phase easier.

“Simulator A”



May not be supported

Questions/Suggestions on Interoperability

Interoperability using ASAM OpenSCENARIO 1.x

- The ASAM OpenSCENARIO V1.x schema should remove any ambiguity to resolve the interoperability concerns.
- What's the best practice for defining RoutingAction? Init or Story or both?
- StoryboardElementStateCondition makes the serialization of the action phase easier. But some simulators do not implement this condition.
- Can we bring in concept of “Serial” and “Parallel” nodes into OSC1.x version which can address most of issues we are seeing today?

Key Takeaways

Generate ASAM OpenSCENARIO from recorded sensor data

- **Virtual scenarios** generated from recorded data enables closed-loop simulation for root-cause analysis and regression testing.
- **A workflow to generate virtual scenarios** typically includes localizing the ego and reconstructing roads, static objects, and actor trajectories.
- You can get started with this workflow using examples from **Scenario Builder for Automated Driving Toolbox**
- RoadRunner Scenario can export **ASAM OpenSCENARIO (1.x and 2.0)** for the virtual scenario.

Thank you for your attention!

Please contact me at spark@mathworks.com with questions.

Seo-Wook Park

Principal Application Engineer for ADAS/Automated Driving

MathWorks