Transmitted by the expert from SAFE

Informal document **GRVA-14-08**
14th GRVA, 26-30 September 2022
Provisional agenda item 4

ASAM OpenSCENARIO 2.0.0

Update to WP.29/GRVA

Gil Amid (gil.amid@foretellix.com)
SAFE
(Member of ASAM Technical Steering Committee)

6. September 2022

Association for Standardization of Automation and Measuring Systems

Agenda

The presentation file itself contains backup slides with more details and examples.

Key Messages

- ASAM – Industry-driven standardization organization continues to deliver standards toward ADS development and Validation
- ASAM is preparing a full suite of standards aimed at ADS development, V&V, testing

- ASAM has released ASAM OpenSCENARIO® 2.0.0 – a major OpenSCENARIO revision. Released on July-20th.

- ASAM OpenSCENARIO 2.0.0 is opening new paths for ADS safety assurance validation, testing and certification.

ASAM – Intro (www.asam.net)

- **ASAM (Association for Standardization of Automation and Measuring Systems) is a non-profit organization that promotes standardization for tool chains in automotive development and testing.**

- **ASAM e.V was founded on Dec. 1st, 1998 in Stuttgart Germany. An initiative of German car manufactures AUDI, BMW, Daimler, Porsche, VW.**

- **ASAM Standards focus on defining data models, file formats, communication APIs, software component APIs, and communication protocols**
- **A Partner in the Standardization Community**
  - **Several Liaison Agreements with ISO**
  - **MoU with SAE, discussions with AVSC and ORAD committees**
  - **Attendee Agreement with AUTOSAR**
  - **Eclipse Foundation: discussions on common activities in OpenX, openMDM**
  - **Observer seat at IAMTS Executive Committee, collaboration with IAMTS WGs**

SAFE
Securing America's Future Energy

ASAM Membership

More than 400 member organizations from around the world develop and apply ASAM standards

OEMs

Tier-1 Suppliers

Tool Vendors / Service Providers

Universities / Research Institutes

Status: Apr 27, 2022

ASAM

ASAM Standards Portfolio

ASAM is currently active in 7 domains

Simulation

OpenCRG | OpenDRIVE | OpenLABEL | OpenSCENARIO | OSI

Measurement & Calibration

ARTI | CDF | CPX | HMS | MCD-1 CCP / XCP | MCD-1 POD | MCD-2 MC | MCD-2 CERP | MDF

Data Management & Analysis

CEA | ODS

Diagnostics

MCD-2 D | MCD-3 D

Test Automation

ACI | ASAP 3 | ATX | GDI | iLinkRT | MCD-3 MC | XIL | OTX Extensions

ECU Networks

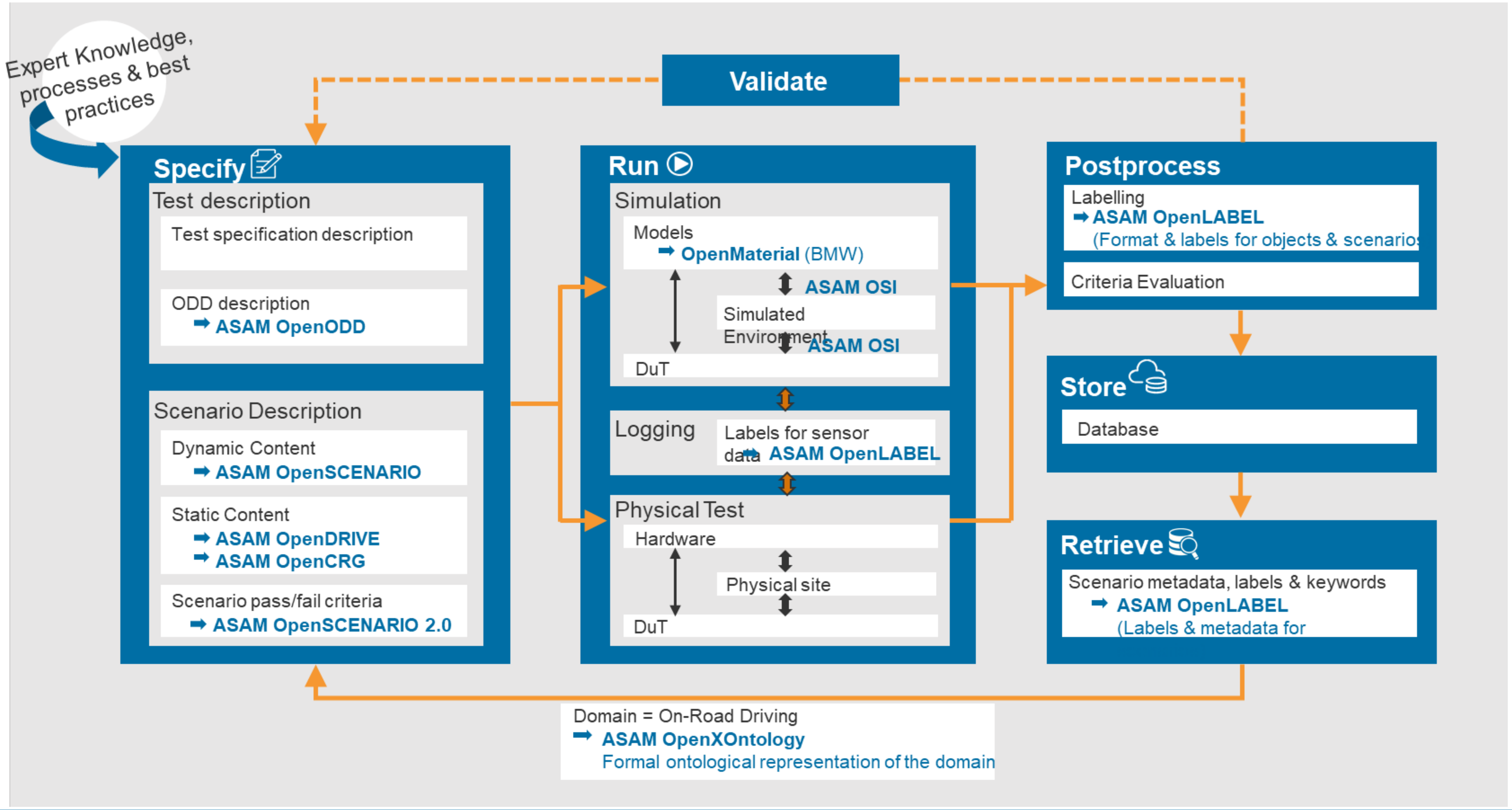
MCD-2 NET

Software Development

CC | FSX | ISSUE | LXF | MBFS | MDX | SCDL

https://www.asam.net/standards/

Scenario-Based Validation - OpenX Standards @ ASAM:



ASAM OpenSCENARIO 2.0.0

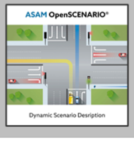
ASAM OpenSCENARIO® 2.0.0 – In plain language

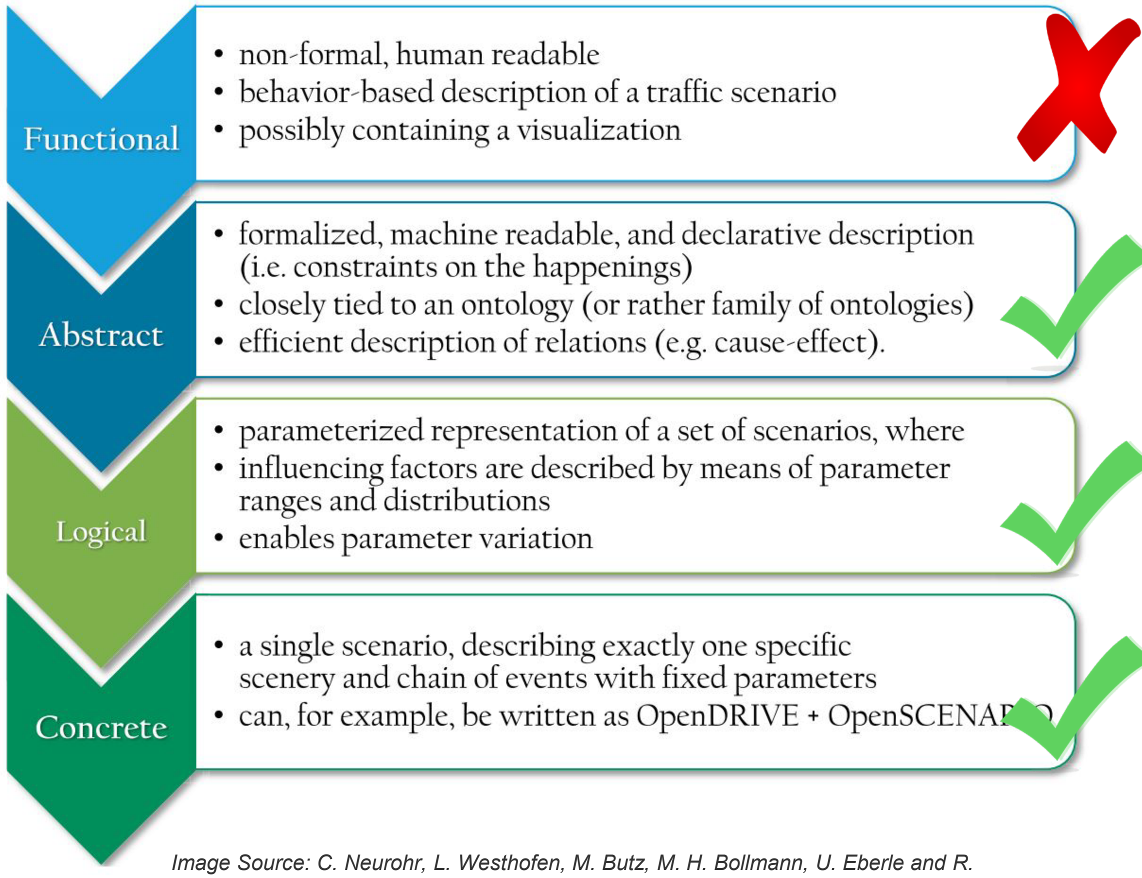
- A Major revision of ASAM OpenSCENARIO® Scenario Description Language.

- This revision enables new use models for:
  - Developers of ADS
  - Testers and qualifiers of ADS (Technical Services)
  - Regulatory entities.
  - The language has built-in capabilities for testing, data aggregation (Coverage data and checks accumulation)

ASAM OpenSCENARIO 2.0 – In slightly less plain language

- This is a declarative domain specific **programming language** combined with a domain model, specifying entities with their properties
- The language supports **abstract, logical** and **concrete** levels of abstraction

	Concrete	Logical	Abstract	Functional
OpenSCENARIO 1	✓	✓		
OpenSCENARIO 2 	✓	✓	✓	



Functional
- non-formal, human readable
- behavior-based description of a traffic scenario
- possibly containing a visualization

✗

Abstract
- formalized, machine readable, and declarative description (i.e. constraints on the happenings)
- closely tied to an ontology (or rather family of ontologies)
- efficient description of relations (e.g. cause-effect).

✓

Logical
- parameterized representation of a set of scenarios, where
- influencing factors are described by means of parameter ranges and distributions
- enables parameter variation

✓

Concrete
- a single scenario, describing exactly one specific scenery and chain of events with fixed parameters
- can, for example, be written as OpenDRIVE + OpenSCENARIO

✓

Image Source: C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle and R. Galbas, "Criticality Analysis for the Verification and Validation of Automated Vehicles," in IEEE Access, vol. 9, pp. 18016-18041, 2021, doi: 10.1109/ACCESS.2021.3053159.

Abstraction levels – for different use cases

```
# Concrete – specific test
ego.drive() with:
    speed(10kph)


# Logical  (R-157 speed range)
ego.drive() with:
    speed(0..60kph])

# Abstract
ego.drive() with:
    keep(it.speed <= speed_limit)
```

"The ADS shall not cross the legal speed limit "

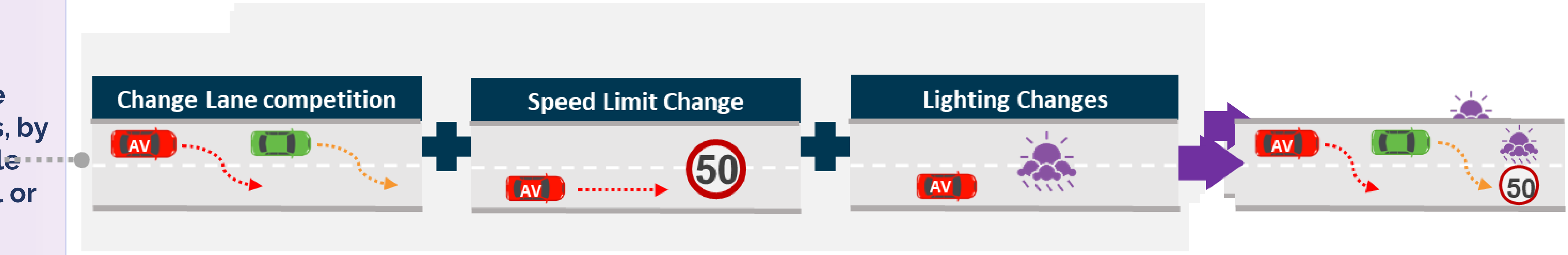
Formal, abstract scenarios enable the same test scenarios to be used in different ODDs, different laws, and different requirements - just define required value.

ASAM OpenSCENARIO 2.0.0 – Enabling new capabilities

- The language features enable:
- **Developments and maintenance of test scenarios that are independent of specific map and geography** (abstract road networks). Can automatically be mapped to a geography or to a required ODD.
- **Consistent reporting and documentation** of error checking and scenario test coverage results – using built in checking mechanism and coverage accumulation
- **Shared and consistent definition of ADS errors**, thresholds and pass/fail criteria – using constraints, events and built-in checking.
- **Consistent definition of required testing ranges** for every value/parameter (from colors of the vehicles to acceleration …..)
- **Reuse or combine individual scenarios** from libraries/catalogs in order to create more complex scenarios.

Scenario Composition - use catalogs/libraries to create complex interactions and ODD conditions.

Easily compose complex scenarios, by combining simple scenarios in serial or parallel



Scenarios are building blocks for more complex scenarios and tests.
Usage: ALKS multi vehicle interactions can be built upon single ALKS scenarios.

Summary

- ASAM has released ASAM OpenSCENARIO® 2.0.0 – a major OpenSCENARIO revision. Released on July-20th.

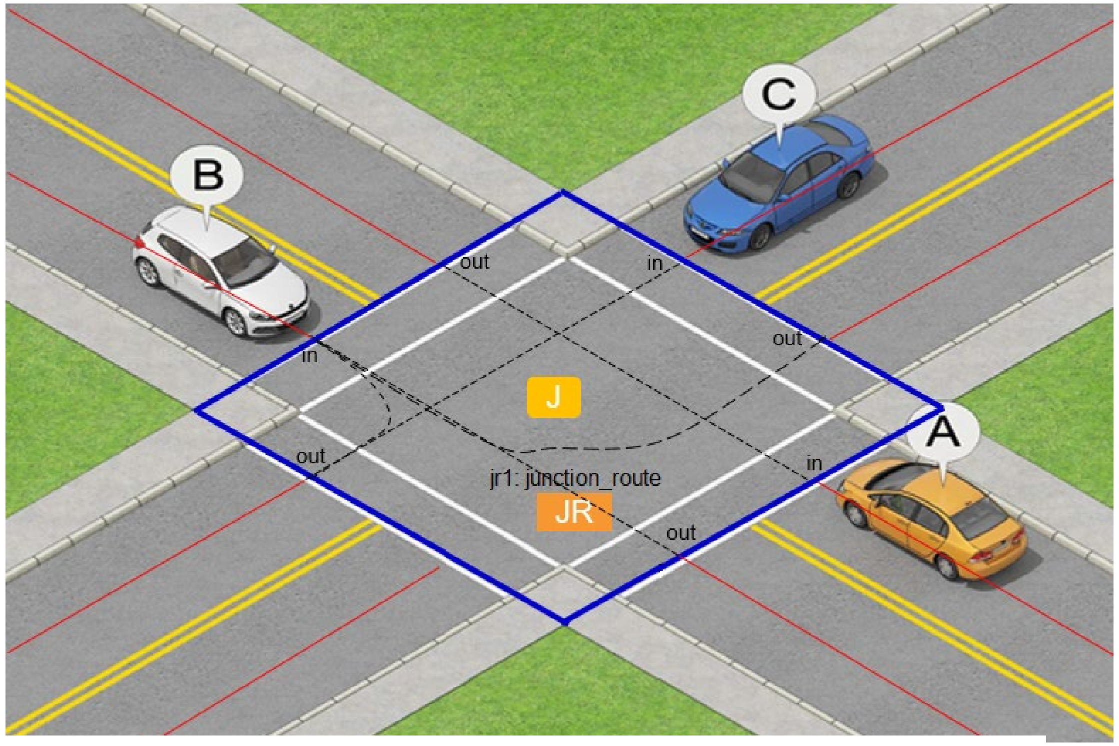
- ASAM OpenSCENARIO® 2.0.0 is supplying tools and mechanisms for regulators to communicate with developers, for better ADS safety assurance validation, testing and certification.

Note: ASAM OpenSCENARIO 2.0 supplies new functionality and, in addition, covers a large majority of what can be done in 1.x. Later releases of 2.x will continue to improve this coverage, with the goal that 2.x becomes a full superset to 1.x. The previous ASAM OpenSCENARIO® version [1.x] has gained a large amount of traction in the industry and in tools for concrete/logical scenarios. ASAM aims to support this through continually improved migration from 1.x to the superset and higher levels of abstraction supported by 2.x.

Backup slides – Road Abtractions

ASAM OpenSCENARIO 2.0 – Road Abstraction

- **An abstract description of the features of the road network that influence the behavior of the actors during the scenario.**
- **Usage: Scenario can automatically be placed on any location within the map, where these features exist.**
- **Usage: Ability to describe actor behaviors on these features.**



```
my_junction: junction
road_1, road_2, road_3, road_4: road
jr_12, jr_34: road

r1: map.roads_follow_in_junction(junction: my_junction, in_road: road_1, out_road: road_2, junction_route: jr_12)
r3: map.roads_follow_in_junction(junction: my_junction, in_road: road_3, out_road: road_4, junction_route: jr_34)
```

Backup slides – Abstraction levels in details.

Abstraction Levels in OpenSCENARIO V2.0.0

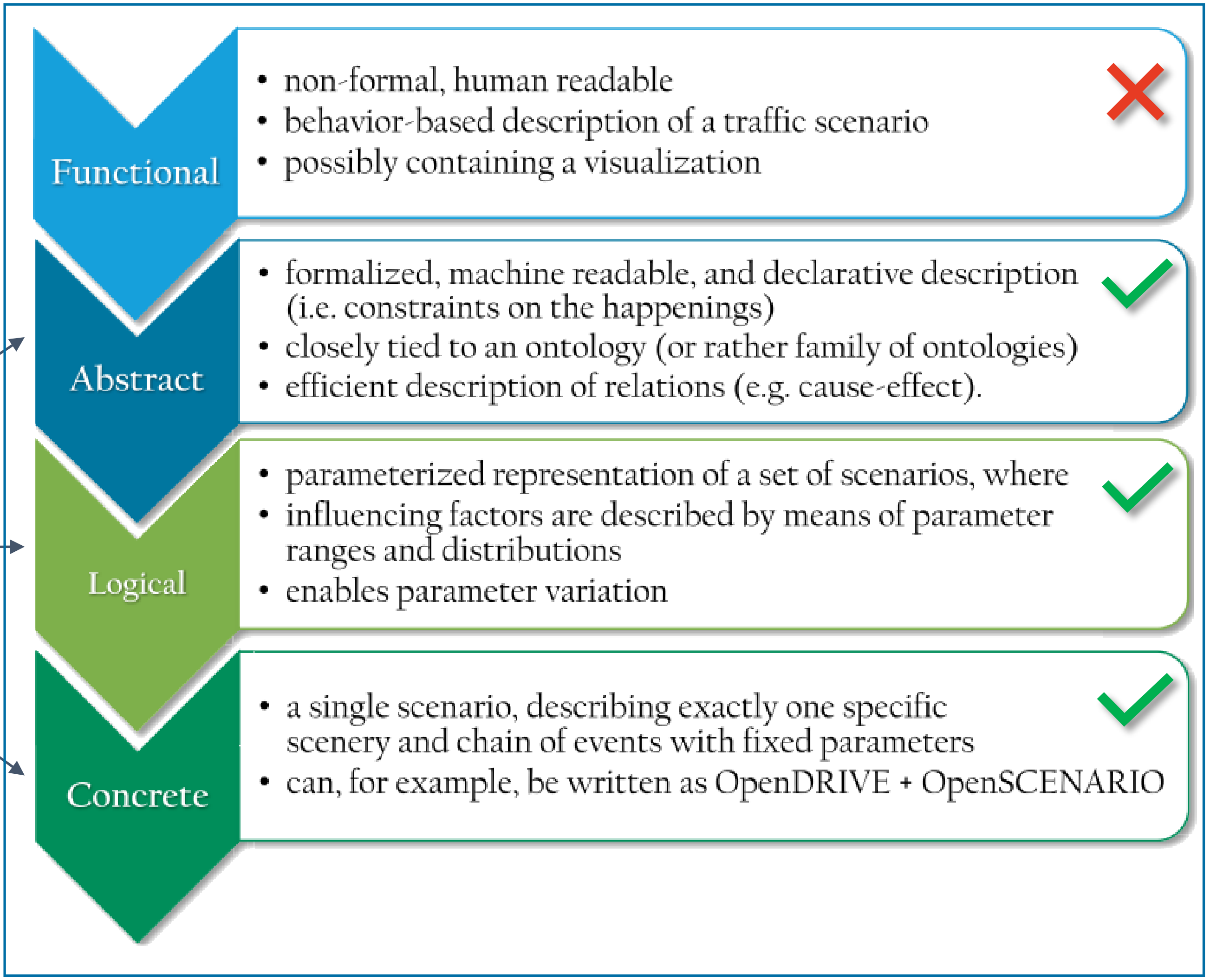
Not in the scope of OSC 2.0

→ Manual conversion to another level required

Can be stored in OpenSCENARIO

→ Automation of conversion is possible

One standardized format for all three levels



Functional ✗
- non-formal, human readable
- behavior-based description of a traffic scenario
- possibly containing a visualization

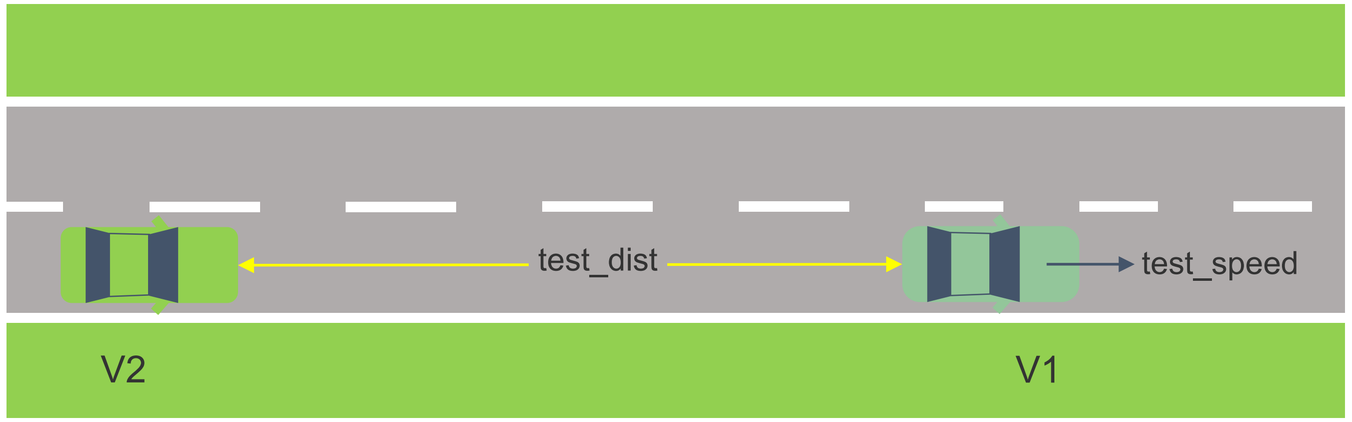
Abstract ✓
- formalized, machine readable, and declarative description (i.e. constraints on the happenings)
- closely tied to an ontology (or rather family of ontologies)
- efficient description of relations (e.g. cause-effect).

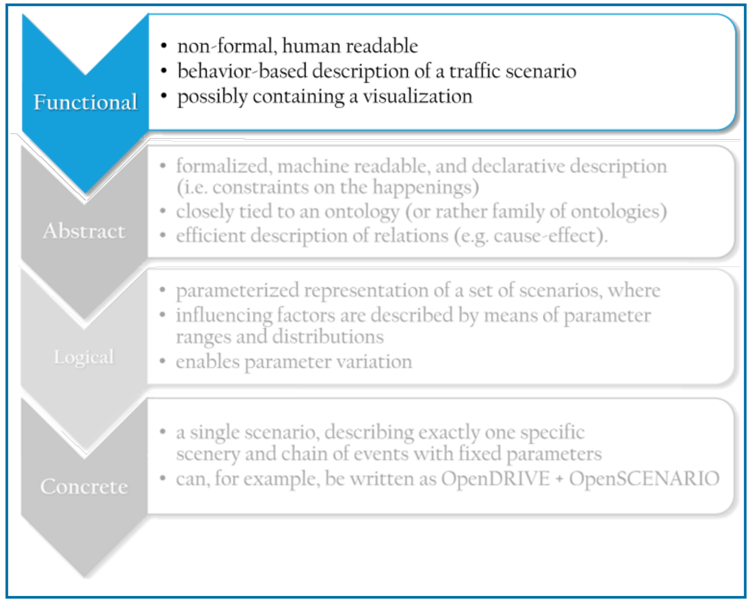
Logical ✓
- parameterized representation of a set of scenarios, where
- influencing factors are described by means of parameter ranges and distributions
- enables parameter variation

Concrete ✓
- a single scenario, describing exactly one specific scenery and chain of events with fixed parameters
- can, for example, be written as OpenDRIVE + OpenSCENARIO

Abstraction Levels in OpenSCENARIO V2.0.0

- The scenario describes two vehicles, v1 and v2
- v1 is driving at a constant speed (test_speed)
- v2 is following v1 at a certain distance (test_dist)





Functional
- non-formal, human readable
- behavior-based description of a traffic scenario
- possibly containing a visualization

Abstract
- formalized, machine readable, and declarative description (i.e. constraints on the happenings)
- closely tied to an ontology (or rather family of ontologies)
- efficient description of relations (e.g. cause-effect).

Logical
- parameterized representation of a set of scenarios, where
- influencing factors are described by means of parameter ranges and distributions
- enables parameter variation

Concrete
- a single scenario, describing exactly one specific scenery and chain of events with fixed parameters
- can, for example, be written as OpenDRIVE + OpenSCENARIO

Abstraction Levels in OpenSCENARIO V2.0.0

```
# Simple scenario where the Ego vehicle follows
# another vehicle at a constant distance
import osc.standard

scenario follow:
    v1, v2:     vehicle

    test_speed: speed
    test_dist:  length

    do parallel(overlap: equal):
        v1.drive() with:
            keep_lane()
            speed(speed: test_speed)

        serial:
            v2.change_space_gap(reference: v1,
                                direction: behind
                                target: test_dist) with:
                lane(same_as: v1)
            v2.keep_space_gap(reference: v1,
                              direction: longitudinal)
```

standard library import

scenario declaration

actor parameters

physical type parameters

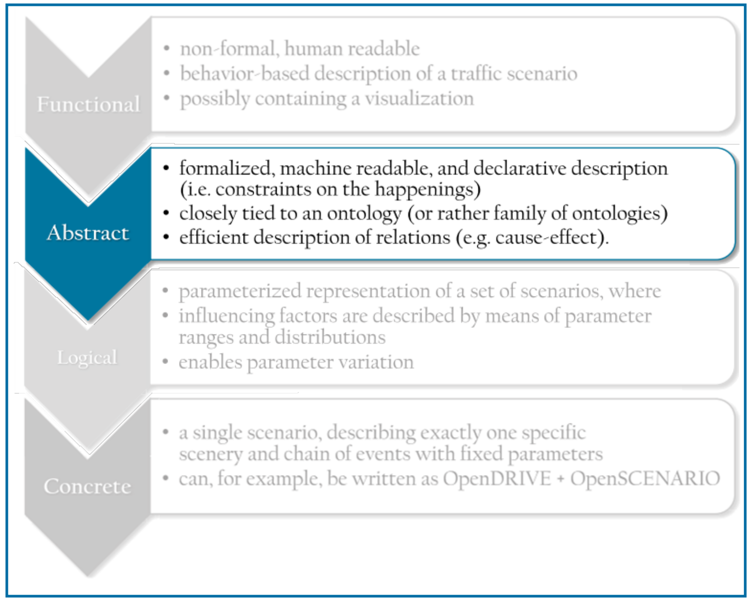
behavior

scenario composition operator

generic action

modifiers

specialized action



Functional
- non-formal, human readable
- behavior-based description of a traffic scenario
- possibly containing a visualization

Abstract
- formalized, machine readable, and declarative description (i.e. constraints on the happenings)
- closely tied to an ontology (or rather family of ontologies)
- efficient description of relations (e.g. cause-effect).

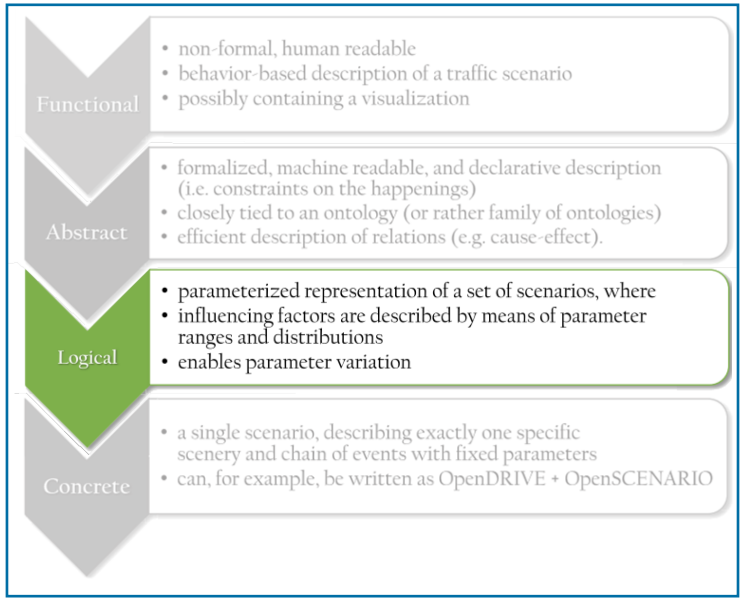
Logical
- parameterized representation of a set of scenarios, where
- influencing factors are described by means of parameter ranges and distributions
- enables parameter variation

Concrete
- a single scenario, describing exactly one specific scenery and chain of events with fixed parameters
- can, for example, be written as OpenDRIVE + OpenSCENARIO

Abstraction Levels in OpenSCENARIO V2.0.0

constraints

```
# Simple scenario where the Ego vehicle follows
# another vehicle at a constant distance
import osc.standard

scenario follow:
    v1, v2:     vehicle

    test_speed: speed
    test_dist:  length

    keep(test_dist in [30m .. 50m])
    keep(test_speed in [40kph .. 60kph])

    do parallel(overlap: equal):

        v1.drive() with:
            keep_lane()
            speed(speed: test_speed)

        serial:
            v2.change_space_gap(reference: v1,
                                direction: behind
                                target: test_dist) with:
                lane(same_as: v1)
            v2.keep_space_gap(reference: v1,
                              direction: longitudinal)
```



Functional:
- non-formal, human readable
- behavior-based description of a traffic scenario
- possibly containing a visualization

Abstract:
- formalized, machine readable, and declarative description (i.e. constraints on the happenings)
- closely tied to an ontology (or rather family of ontologies)
- efficient description of relations (e.g. cause-effect).

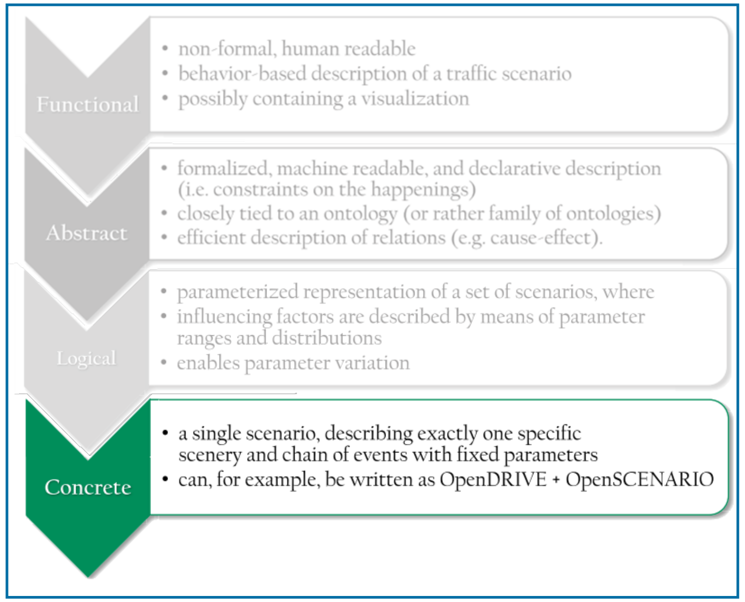
Logical:
- parameterized representation of a set of scenarios, where
- influencing factors are described by means of parameter ranges and distributions
- enables parameter variation

Concrete:
- a single scenario, describing exactly one specific scenery and chain of events with fixed parameters
- can, for example, be written as OpenDRIVE + OpenSCENARIO

Abstraction Levels in OpenSCENARIO V2.0.0

```
# Simple scenario where the Ego vehicle follows
# another vehicle at a constant distance
import osc.standard

scenario follow:
    v1, v2:     vehicle

    test_speed: speed
    test_dist:  length

    keep(test_dist == 34m)
    keep(test_speed == 55kph)

    do parallel(overlap: equal):

        v1.drive() with:
            keep_lane()
            speed(speed: test_speed)

        serial:
            v2.change_space_gap(reference: v1,
                                direction: behind
                                target: test_dist) with:
                lane(same_as: v1)
            v2.keep_space_gap(reference: v1,
                              direction: longitudinal)
```

constraints



Functional
- non-formal, human readable
- behavior-based description of a traffic scenario
- possibly containing a visualization

Abstract
- formalized, machine readable, and declarative description (i.e. constraints on the happenings)
- closely tied to an ontology (or rather family of ontologies)
- efficient description of relations (e.g. cause-effect).

Logical
- parameterized representation of a set of scenarios, where
- influencing factors are described by means of parameter ranges and distributions
- enables parameter variation

Concrete
- a single scenario, describing exactly one specific scenery and chain of events with fixed parameters
- can, for example, be written as OpenDRIVE + OpenSCENARIO

Backup slides – Implementations and open source tools.

ASAM

The industry reacts - initial implementations being published

Foretellix OpenSCENARIO 2.0.0 Execution

ASAM OpenSCENARIO 2.0

The ASAM standard is revolutionizing the way safety is being developed into automated driving systems.

Foretellix is the pioneer and expert in OSC2.0, leading the standard development project and being the first to offer a native implementation in its Foretify™ platform.

OSC 2.0 Native Implementation
The only build to standard implementation build using abstract scenarios, automation & coverage analytics

OSC 2.0 Expertise & Methodology Leadership
Foretellix's technology, methodology, and experts have led the standard development

OSC 2.0 V-suite Packages
Ready-made content packages for L2 ADAS, L3 and L4 to get you started in no time

ASAM

Open Source Tools

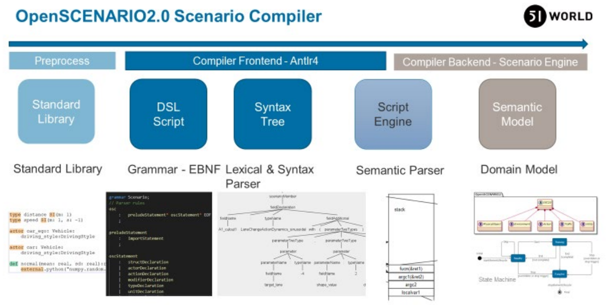
51WORLD OpenSCENARIO2 Grammar Checker (osc2checker)

https://github.com/51WORLD/osc2checker

PMSF py-osc2 Framework

https://github.com/PM

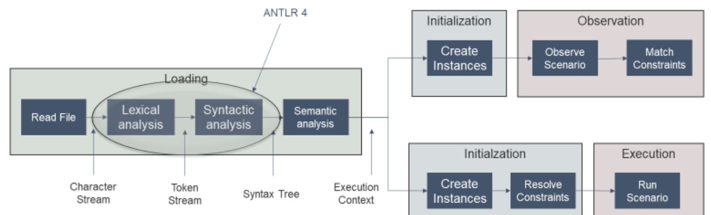
OpenSCENARIO2.0 Scenario Compiler



ASAM

Akka Implementation

Scenario Execution / Observation Steps

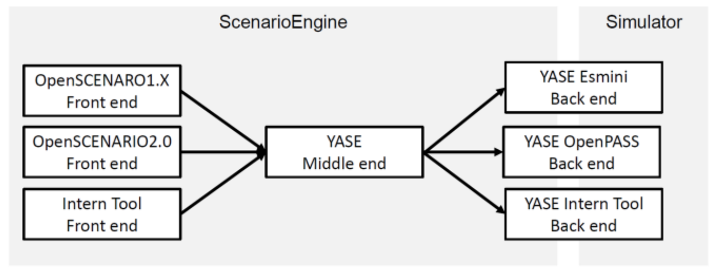


ASAM

YASE (Bosch)

Three phases of scenario compiler/ ScenarioEngine

SET Level



ASAM