ADAS Verification in Co-Simulation: Towards a Meta-Model for Defining Test Scenarios

F. Basciani, V. Cortellessa, S. Di Martino, D. Di Nucci, D. Di Pompeo, C. Gravino, and **L. L. L. Starace**

Università degli Studi di Napoli Federico II, Italy
luigiliberolucio.starace@unina.it
https://luistar.github.io/

19th Workshop on Advances in Model Based Testing (A-MOST 2023), Dublin, Ireland

Advanced Driver Assistance Systems (ADAS)

- Solutions to assist drivers in driving functions
- Can detect obstacles or driver errors, and respond accordingly
- Anti Blocking Systems (ABS), Forward Collision Warning, Emergency Braking, Automated Lane Keeping
- Can largely **improve road safety**, preventing up to 60% of total traffic deaths [1]
- Safety-critical, need to be properly tested

[1] US National Safety Council, Injury Facts, 2022
https://injuryfacts.nsc.org/motor-vehicle/occupant-protection/advanced-driver-assistance-systems/data-details/

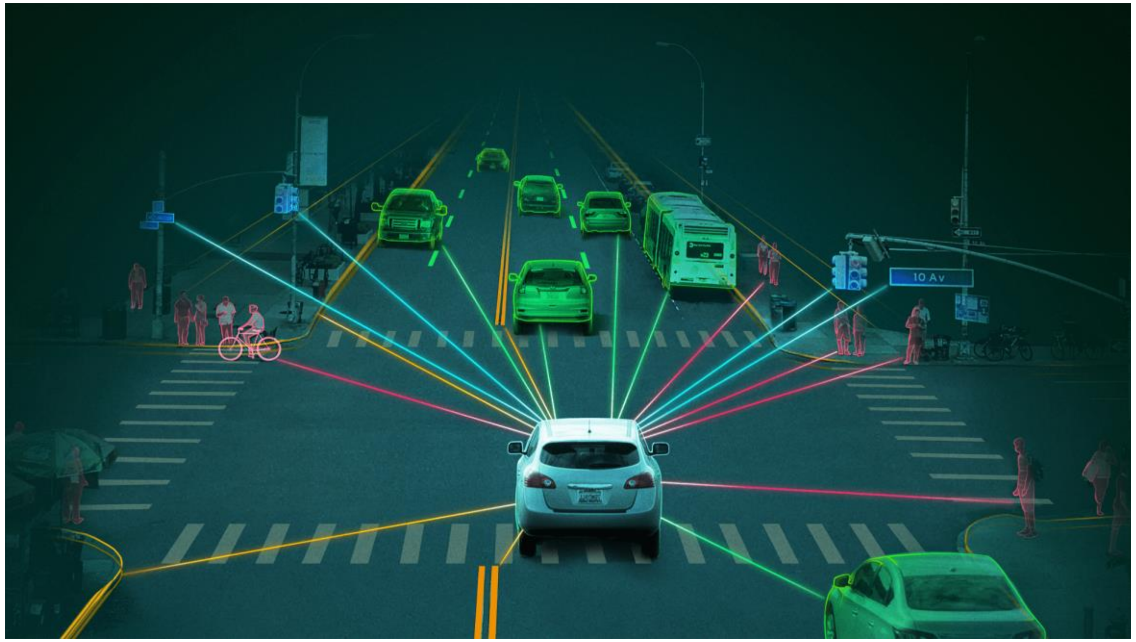
Testing «Traditional» ADAS

- Model-Driven Engineering
- Model-in-the-Loop (MiL)
  - Model is tested using efficient simulation-based methods (Matlab/Simulink)
- Hardware-in-the-Loop (HiL)
  - ECU running the ADAS is tested when connected to a simulated car using real time systems (e.g.: dSpace)

https://www.imc-tm.nl/solutions/e-mobility/

AI-based ADAS

- Next-gen ADAS are more complex
- Pedestrian detection systems
- Emergency braking
- Road sign detection
- Leverage more complex inputs
  - Video streams from cameras
  - Data from lidars

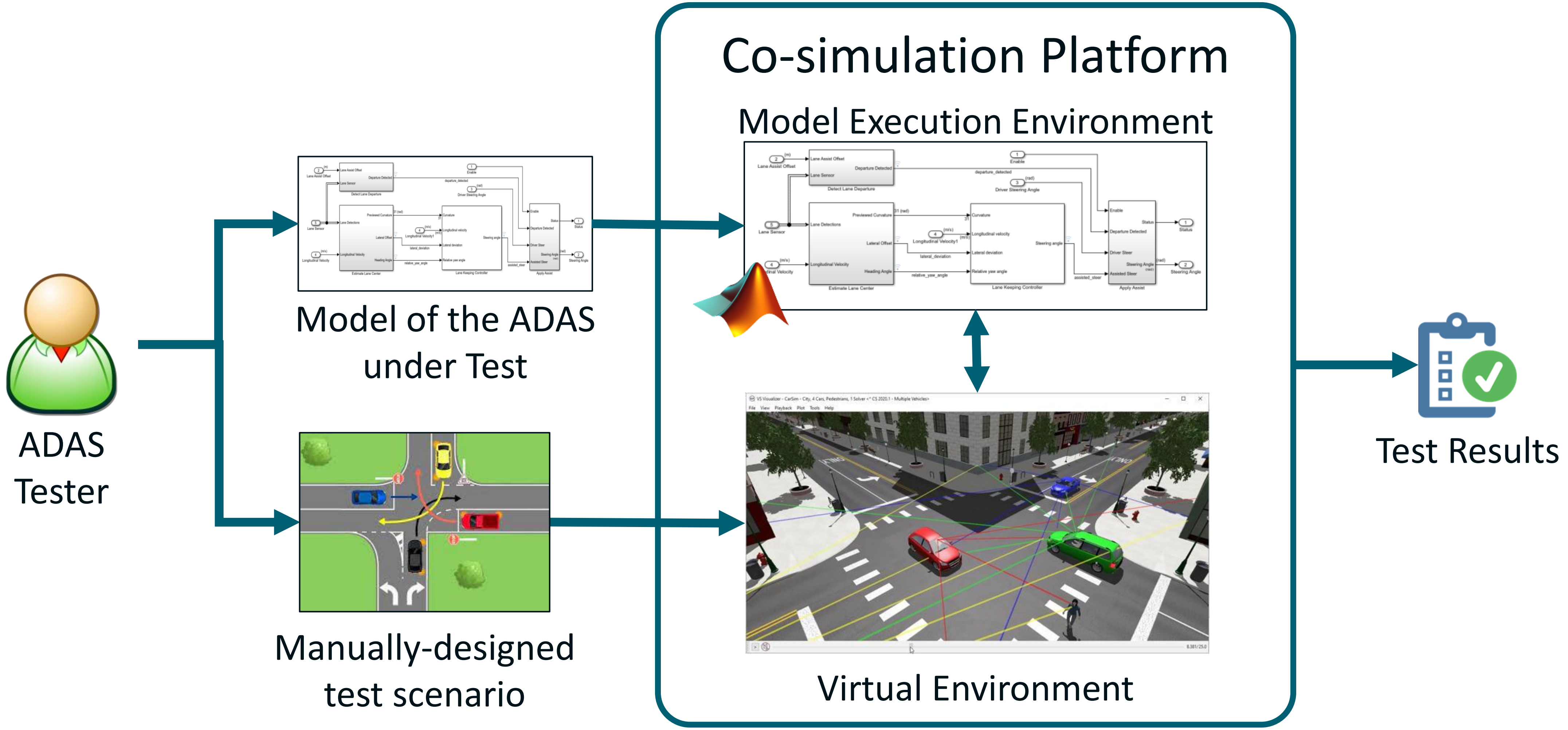


Testing AI-based ADAS

- Hard to reproduce inputs in traditional MiL
- On-field testing also unfeasible
  - Up to billions of kilometers of test drives to guarantee that ADAS are safer than human drivers [1]
  - Safety Issues!

[1] Kalra, N., and Paddock S. M., "*Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability*?" Transportation Research Part A: Policy and Practice 94 (2016)

New Trend: Testing ADAS in Co-Simulation



Challenges in Co-Simulation-based Testing

- Existing solutions highly-tailored to specific cases
    - Lack of a solid reference framework
- Significant manual work required for the set-up of a test scenario
- Domain experts might lack programming competences

Related Works

- Schütt et al. presented a meta-model to define ADAS test scenarios
  - Lacks the possibility to express assertions
  - Lack of a visual editor to design test scenarios in a WYSIWYG-like fashion

SceML - A Graphical Modeling Framework for Scenario-based Testing of Autonomous Vehicles

Barbara Schütt
FZI Research Center for Information Technology
schuett@fzi.de

Thilo Braun
FZI Research Center for Information Technology
braun@fzi.de

Stefan Otten
FZI Research Center for Information Technology
otten@fzi.de

Eric Sax
FZI Research Center for Information Technology
sax@fzi.de

ABSTRACT

Ensuring the functional correctness and safety of autonomous vehicles is a major challenge for the automotive industry. However, exhaustive physical test drives are not feasible, as billions of driven kilometers would be required to obtain reliable results. Scenario-based testing is an approach to tackle this problem and reduce necessary test drives by replacing driven kilometers with simulations of relevant or interesting scenarios. These scenarios can be generated or extracted from recorded data with machine learning algorithms or created by experts. In this paper, we propose a novel graphical scenario modeling language. The graphical framework allows experts to create new scenarios or review ones designed by other experts or generated by machine learning algorithms. The scenario description is modeled as a graph and based on behavior trees. It supports different abstraction levels of scenario description during software and test development. Additionally, the graph-based structure provides modularity and reusable sub-scenarios, an important use case in scenario modeling. A graphical visualization of the scenario enhances comprehensibility for different users. The presented approach eases the scenario creation process and increases the usage of scenarios within development and testing processes.

CCS CONCEPTS

• **Software and its engineering → Domain specific languages**; *Visual languages*; *Integrated and visual development environments*; • **Computing methodologies** → Simulation languages.

Vehicles. In *ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems (MODELS '20), October 18–23, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3365438.3410933

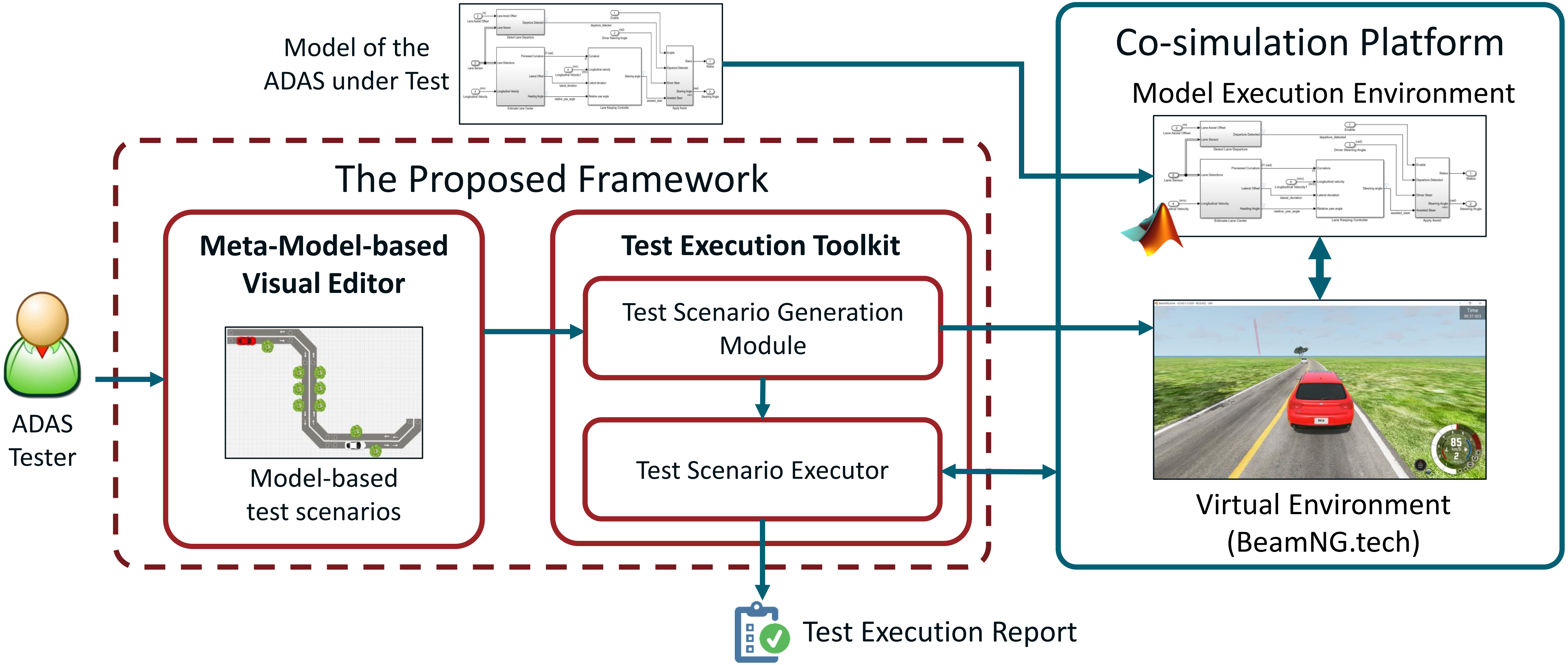
1 INTRODUCTION

In software development, testing is an important step to realize develop systems. In particular, in the automotive industry software validation and verification cannot be neglected and constitutes an integral part of the development process to make sure requirements are met and the system fulfills the intended use cases [26, p. 76]. According to Wachenfeld and Winner [24], real world test drives, known as distance-based testing, are a valid testing method and give the most reliable results. However, distance-based testing is not feasible: Almost 8 billion kilometers are necessary to make a statistically significant comparison at a confidence of 95% to show the failure rate of an autonomous vehicle is lower than the human driver failure rate. This results in 225 years of driving for a fleet of 100 vehicles and if there are changes or variations in the software, all the driving has to be repeated [13, 24].

An alternative is offered by scenario-based testing. Compared to the random test cases emerging during distance-based testing, scenario-based testing is a systematic approach to find new test cases [18]. The goal is to set up a collection of validated and relevant scenarios, depending on the test subject. This approach can be used at different stages during the development process [16]. Scenarios are developed at an abstract level during the concept phase, e.g.

[1] Schütt, Barbara, et al. "*SceML: A graphical modeling framework for scenario-based testing of autonomous vehicles.*" in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*. 2020.

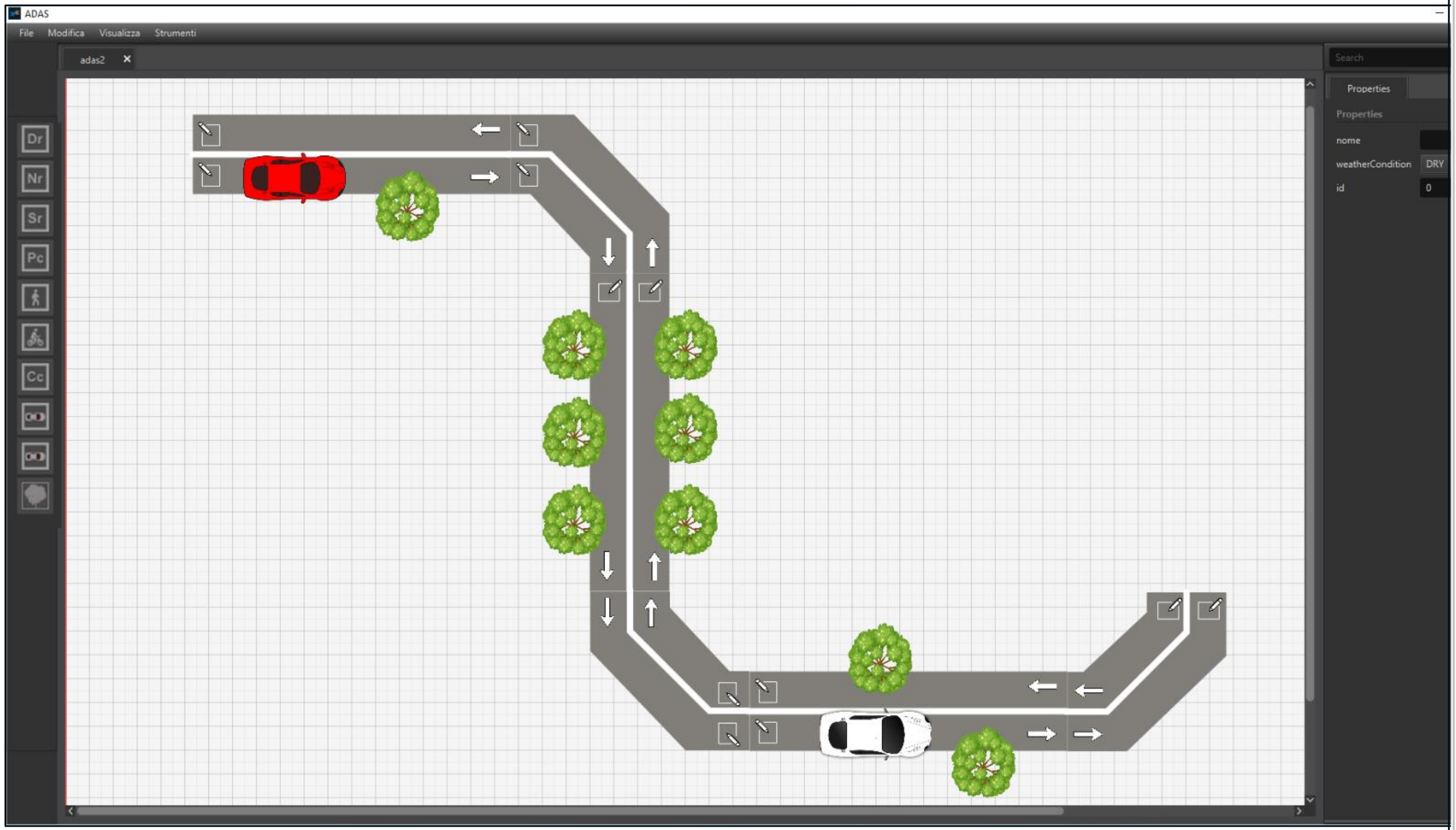
Proposal: A Meta-model Based Framework

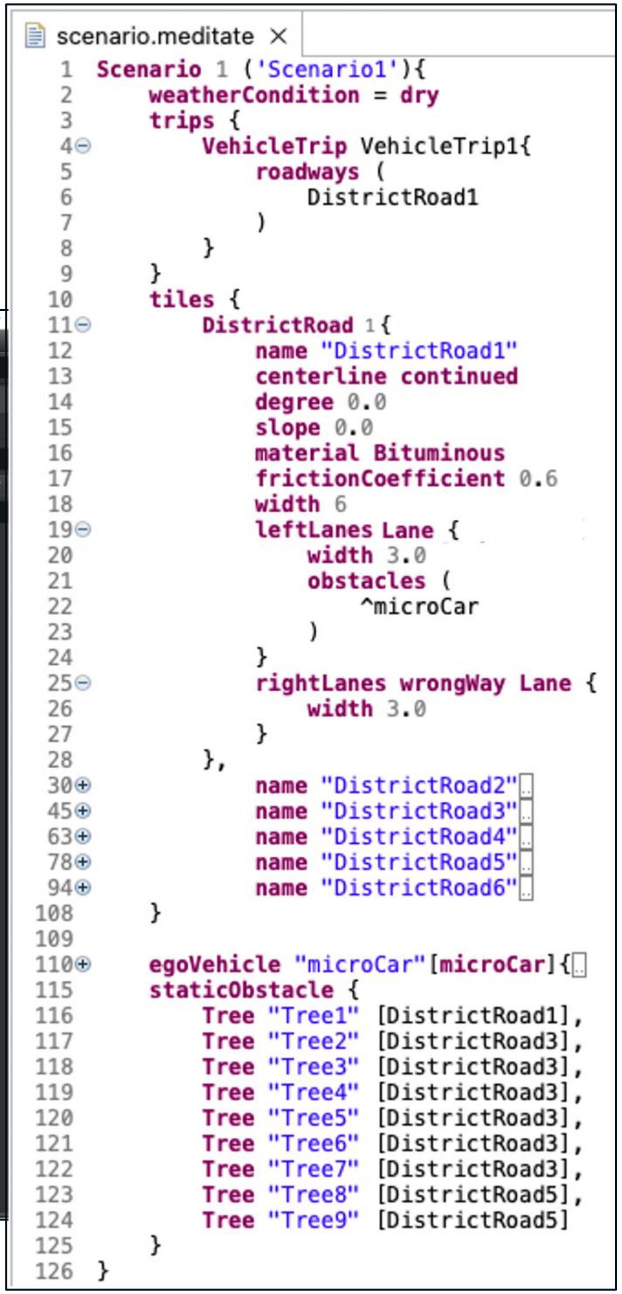


Proposed Meta-model for ADAS Test Scenarios

- Built on-top of the Eclipse Modelling Framework
- Allows for the definition of static aspects
  - Road topology
  - Weather conditions
  - Static obstacles (i.e., trees, fences, vehicles, etc.)
- As well as dynamic aspects
  - Dynamic of the ego-vehicle (i.e., path it follows, max speed, etc.)
  - Dynamic obstacles (i.e., other vehicles)
- Allows for the definition of assertions

Visual and Textual Editors



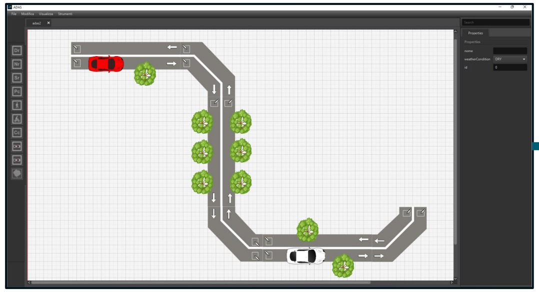



```
Scenario 1 ('Scenario1'){
    weatherCondition = dry
    trips {
        VehicleTrip VehicleTrip1{
            roadways (
                DistrictRoad1
            )
        }
    }
    tiles {
        DistrictRoad 1{
            name "DistrictRoad1"
            centerline continued
            degree 0.0
            slope 0.0
            material Bituminous
            frictionCoefficient 0.6
            width 6
            leftLanes Lane {
                width 3.0
                obstacles (
                    ^microCar
                )
            }
            rightLanes wrongWay Lane {
                width 3.0
            }
        },
            name "DistrictRoad2"
            name "DistrictRoad3"
            name "DistrictRoad4"
            name "DistrictRoad5"
            name "DistrictRoad6"
    }

    egoVehicle "microCar"[microCar]{
    staticObstacle {
        Tree "Tree1" [DistrictRoad1],
        Tree "Tree2" [DistrictRoad3],
        Tree "Tree3" [DistrictRoad3],
        Tree "Tree4" [DistrictRoad3],
        Tree "Tree5" [DistrictRoad3],
        Tree "Tree6" [DistrictRoad3],
        Tree "Tree7" [DistrictRoad3],
        Tree "Tree8" [DistrictRoad5],
        Tree "Tree9" [DistrictRoad5]
    }
}
```

Proof of Concept

- We developed a Python-based Test Execution Toolkit



Test Scenario defined using the proposed Meta-model

Test Execution Toolkit

Test Scenario Generation Module

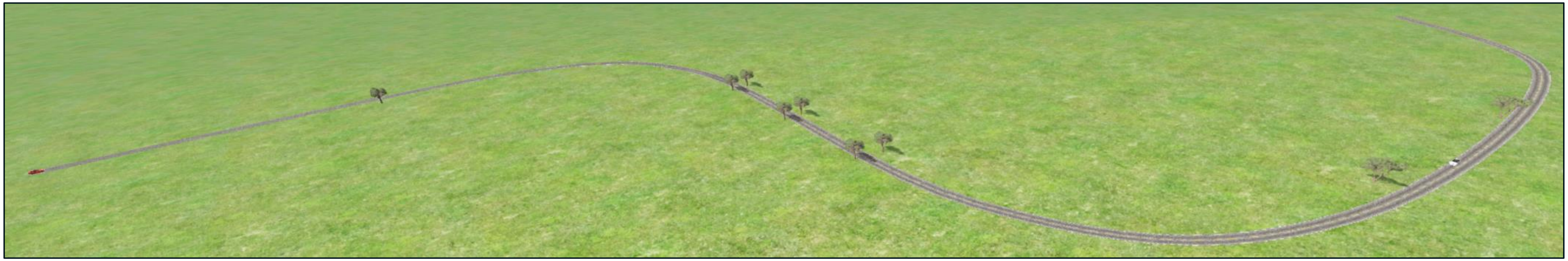
Test Scenario Executor

BeamNG.tech

Virtual Environment

Test Execution Data

Co-simulation (Example)

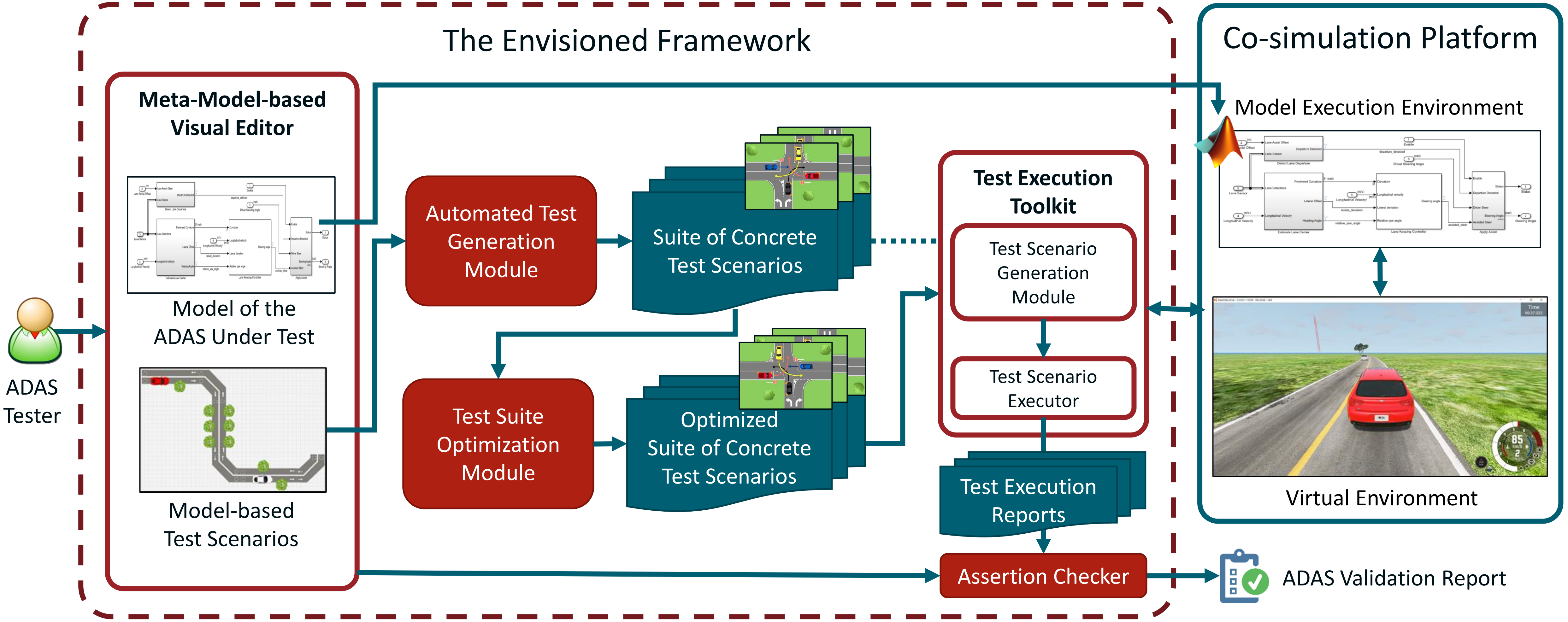


Collected Co-simulation Data

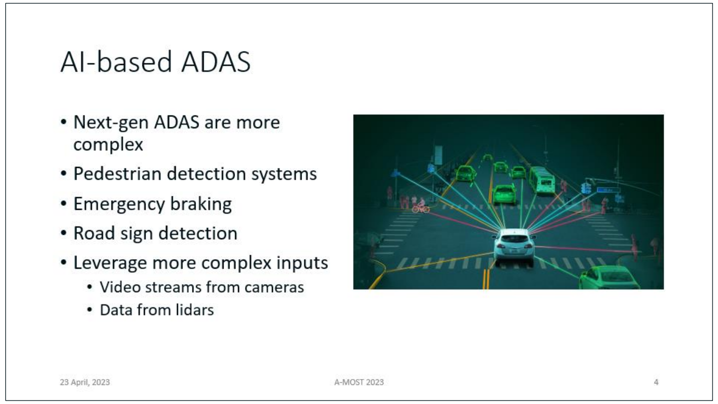
- Leveraging BeamNG APIs, we equipped the ego vehicle with a number of virtual sensors
- Recording data about its state during the simulation
  - Absolute position in the virtual map
  - Speed, rotational acceleration and forces
  - Accelerator, Brake, and steering inputs
  - Fuel Level, ABS status
  - Surface and Core brake temperatures for each wheel

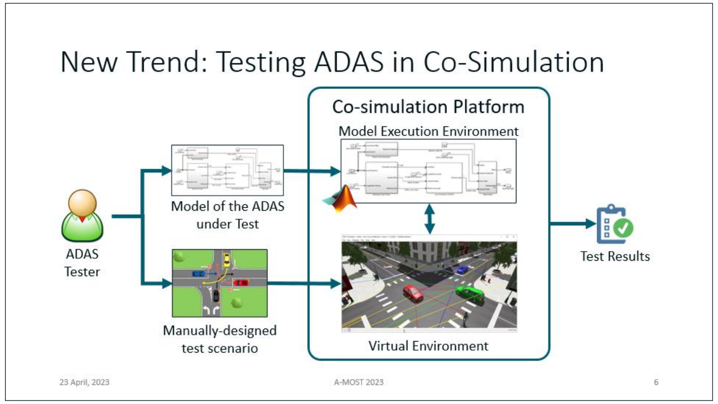

```
[
  {
    "state": {
      "pos": [16.9, -36.5, -27.7],
      …
    },
    "electrics": {
      "fuel": 0.85,
      "gear": 2,
      "wheelspeed": 40.45,
      "parkingbrake": 0,
      "throttle_input": 0.80,
      "clutch_input": 0,
      "brake_input": 0,
      "reverse": 0,
      "abs": 0,
      "steering_input": -0.03,
      "turnsignal": 0,
      "rpm": 1528.68,
      "wheelThermals": {
        "RR": {
          "brakeSurfaceTemperature": 14.99,
          "brakeCoreTemperature": 14.99
        },
        "FR": {
          "brakeSurfaceTemperature": 14.99,
          "brakeCoreTemperature": 14.99
        },
        "RL": {
          "brakeSurfaceTemperature": 14.99,
          "brakeCoreTemperature": 14.99
        },
        "FL": {
          "brakeSurfaceTemperature": 14.99,
          "brakeCoreTemperature": 14.99
        }
      }, …
  }, …
]
```

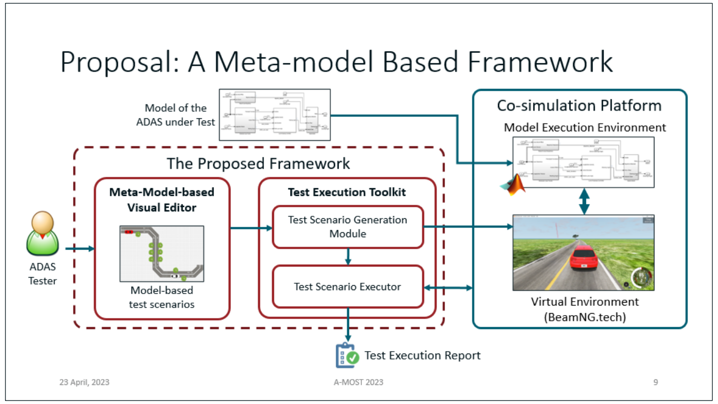
Envisioned Framework

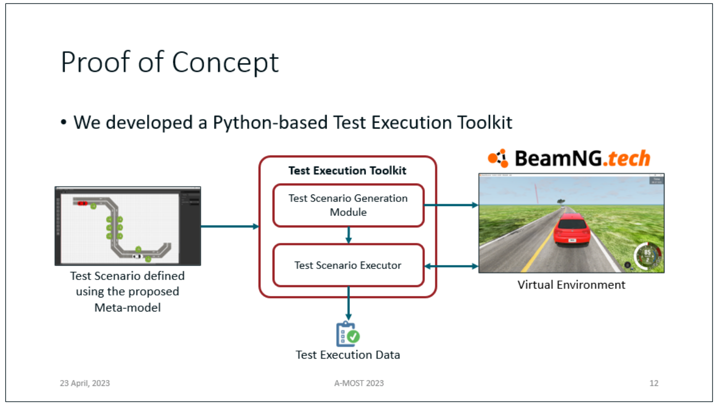


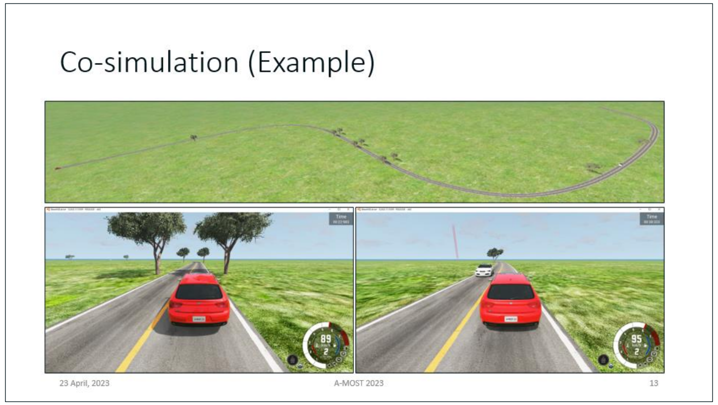
ADAS Verification in Co-Simulation: Towards a Meta-Model for Defining Test Scenarios

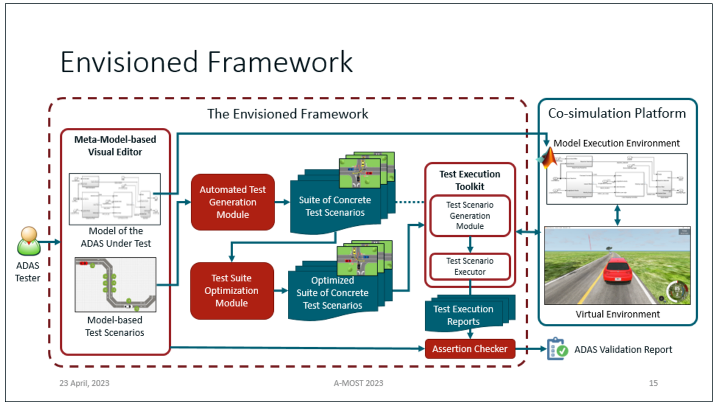












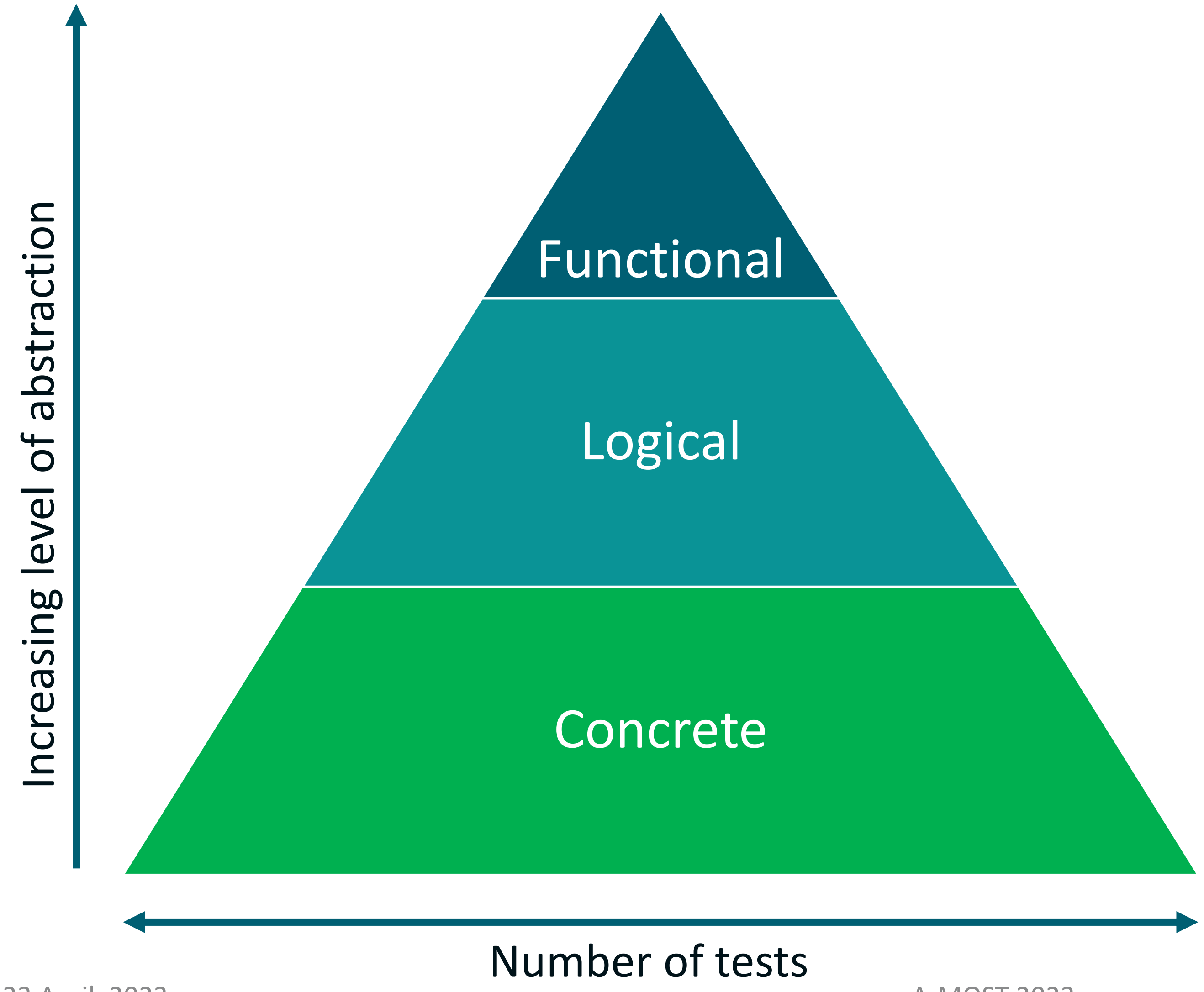
Luigi Libero Lucio Starace

luigiliberolucio.starace@unina.it

https://luistar.github.io

Backup slides

ADAS Test Scenarios: Abstraction Levels



Functional Test Scenario

ADAS: Emergency Braking
Setting: Urban intersection; Ego-vehicle takes a turn; Pedestrian crosses the street
Expected: No collision

Logical Test Scenario

Vehicle speed: [10 – 50 km/h]
Weather: {sun, rain, fog, snow}
Pedestrian speed: [2 – 10 km/h]
Turn direction: {left, right}
…

Concrete Test Scenario

Vehicle speed: 45 km/h
Weather: fog
Pedestrian speed: 10 km/h
Turn direction: right
…

Proposed Meta-Model: Details