OpenSCENARIO

bringing content to the road

2nd OpenSCENARIO Meeting
June 29th, 2016
Agenda

Session 1
- Welcome
- Before we start...
- Motivation / Presentations by Core-Team Members
- Status of OpenSCENARIO and Technical Report #1

Session 2
- Status of OpenSCENARIO and Technical Report #2
- Questions and Discussion of OpenSCENARIO Format
- Tools and Examples
- Interaction with PEGASUS
- Organizational Issues
Before we start....
Declaration

The parties recognize that they are (actual or potential) competitors. For this very reason they will avoid discussing commercially sensitive information to prevent improper communications or even the appearance of impropriety, as each party understands the implications of competition laws. Accordingly, the parties have agreed to follow these guidelines during their discussion or negotiation of the Agenda Items.

Discussions between the parties will be limited to topics necessary to discuss or negotiate the Agenda Items. The parties will not discuss information that relates to their competitive activities, such as information related to the marketing or pricing of goods or the procurement of inputs.

For example, with respect to pricing and marketing, the parties WILL NOT DISCUSS:

- Either party’s marketing strategies.
- Either party’s pricing policies; pricing plans and strategies; price differentials; mark-ups; discounts or rebates; payment, credit or other trading terms and conditions, etc.
- Either party’s views as to future price trends.
- Either party’s views as to future sales projections.
- Either party’s product plans and market shares

With respect to inputs and capacity, the parties WILL NOT DISCUSS:

- Either party’s current or anticipated future costs of production and operations.
- Either party’s views as to market-wide production or capacity trends.
- Either party’s capacity plans as a general matter or with respect to a specific facility.
- Either party’s production strategy; plans to expand or reduce output; current or expected levels of orders; plant designs, etc.
- Technical plans for development; research projects; planned innovations; trade secrets; or other proprietary technology and data, etc.

The Parties will ensure that an appropriate person will take meeting minutes and distribute them afterwards. Senior representatives of the parties attending the meeting will ensure that these guidelines (and the competition law compliance policies of the relevant party) will be adhered to and will stop any discussion of the above topics (other than the Agenda Items). All attendees recognize the importance in adhering to this procedure and to the Agenda Items.

Each person in attendance, and counsel for each party (whether or not in attendance) must receive a copy of the subsequent minutes. Email transmission of the minutes is acceptable. Retain permanently for your records.
Motivation

(what we are aiming for)
Core Team
**OpenSCENARIO – Road Map**

**Definition Phase, Stage 2**

**Results**
- Support/Website
- Test Tools
- Validation Tools
- Format Specification

**Core Team**
- VIRES
- ASC(S)
- Member A
- Member B
- Member C
- Member D
- Member E
- Member F
- Member G

**Requirements**

**Partners and Users**
- Tool Suppliers (dSPACE, IAV, IPG etc.)
- OEMs (Audi, BMW, Volkswagen etc.)
- Research (DLR, fka, FKFS, HLRS etc.)
- Assoc. Partners (Daimler, Opel, Porsche)
- Tier1 (tbd.)
- 3rd parties (e.g. authorities)
as of June 28, 2016 (co-members of PEGASUS are highlighted - company only not necessarily persons)

- Adam Opel AG, Rüsselsheim, Germany
- ASCS e.V., Stuttgart, Germany
- AUDI Electronics Venture GmbH, Gaimersheim, Germany
- AVL List GmbH, Graz, Austria
- BMW Group, Munich, Germany
- Daimler AG, Sindelfingen, Germany
- DLR e.V., Braunschweig, Germany
- dSPACE GmbH, Paderborn, Germany
- fka SV, San Jose, CA, USA
- HLRS, Stuttgart, Germany
- IAV GmbH, Gifhorn, Germany
- ika - RWTH Aachen, Aachen, Germany
- Porsche AG, Weissach, Germany
- Rheinmetall Defence Electronics GmbH, Bremen, Germany
- Robert Bosch GmbH, Abstatt, Germany
- University of Leeds, Leeds, United Kingdom
- Valeo Schalter und Sensoren GmbH, Bietigheim-Bissingen, Germany
- VIRES Simulationstechnologie GmbH, Bad Aibling, Germany
- Volkswagen AG, Wolfsburg, Germany
- Volvo Car Corporation, Gothenburg, Sweden
- VTI, Linköping, Sweden

Jochen Schaffnit
Alexander Frederic Walser / Ping Wang
Andreas Kern
Christian Schyr / Thomas Weck
Mohammad Bahram
Andreas Kaden
Andreas Richter / Julian Schindler
Hagen Haupt
Alexander Tenbrock
Uwe Wössner
Kathrin Symkenberg
Christian Roesener / Robert Krajewski
Ingo Krems
Jens Becker
Stefan Küperkoch / Thomas Grosser
Richard Romano
Oliver Cuzic
Andreas Biehn / Marius Dupuis
Jens Krause / Sebastian Grysczyk
Emil Knabe / Fredrik Persson
Jonas Andersson Hultgren
Motivation / Presentation of Core Team
Who we are

What we do

Our Motivation

Getting things done!

Planned Support

native support in VIRES Virtual Test Drive (scenarios and replay files)

Schedule

• replay files already supported (based on draft D)
• native support by end of 2016
Status of OpenSCENARIO
Welcome to the World of OpenSCENARIO!

OpenSCENARIO is an open file format for the description of dynamic contents in driving simulation applications. The project is in its very early stage and will be made available to the public in the very near future. If you want to be informed about the initiative’s progress and major events, please register for the newsletter.

This website is maintained by VIRES Simulationstechnologie GmbH, Germany.
March 2016
Tracking System ("public" and "core team")

http://tracking.vires.com

Newsletter

newsletter@openscenario.org

Available Data

mind map of draft C
mind map and schema of draft D (discussed in PEGASUS and core team)
OpenSCENARIO – Status

First User

OpenSCENARIO is a key element in PEGASUS project
-> we really have to deliver!

PEGASUS will define various initial use cases.
Technical Report
What we did

After a couple of weeks: we took a closer look (again) at the format definition
OpenSCENARIO – Status

What we did

After a couple of weeks: we took a closer look (again) at the format definition

What we found

It looked confusing
OpenSCENARIO – Status

What we did

After a couple of weeks: we took a closer look (again) at the format definition

What we found

It looked confusing

What we did then

We started (almost) from scratch and simplified things
OpenSCENARIO – Status

What we did

After a couple of weeks: we took a closer look (again) at the format definition

What we found

It looked confusing

What we did then

We started (almost) from scratch and simplified things

What we have today (i.e. since yesterday)

A new, revised draft E
OpenSCENARIO – Status

What we did

After a couple of weeks: we took a closer look (again) at the format definition

What we found

It looked confusing

What we did then

We started (almost) from scratch and simplified things

What we have today (i.e. since yesterday)

A new, revised draft E

What we'll do now

Explain it all to you and ask for your input.....
Getting into the Details....
Example 1: simple steering command

- initialize at tbd. position
- after 2s: apply throttle with 40%, keep steering wheel straight
- after 10s: release throttle, apply tbd. torque to steering wheel
- after another 2s: end of scenario

storyboard
  story: owner = Ego
  act 1: condition = simTime > 2s
    sequence 1.1: actor = $owner
      maneuver 1.1.1: name = start driving
      event 1.1.1.1: condition = upon start of act
        action 1.1.1.1.1: apply throttle at 0.4
    sequence 1.2: actor = $owner
      maneuver 1.2.1: name = apply steering torque
      event 1.2.1.1: condition = 10s after start of act
        action 1.2.1.1.1: set throttle to 0.0
        action 1.2.1.1.2: apply 2.0Nm torque to steering wheel

Note: see also "questions" section.
Example 2: two lane changes in front of Ego vehicle, 3-lane motorway

- Upon the Ego vehicle approaching the vehicle in front, the following things shall happen:
  - player 1 in the front lane shall merge to the left
  - with 1s delay player 2 in the front right lane shall merge to the left

**Storyboard**

**story:** owner = Ego

**act 1:** condition = TTC to player 1 in front right lane < 3s

**sequence 1.1:** actor = player 1

  **maneuver 1.1.1:** name = complex lane change

  **event 1.1.1.1:** condition = upon start of act

  **action 1.1.1.1:** slow down by 5kph

  **action 1.1.1.2:** perform lane change to left within 4s

**sequence 1.2:** actor = player 2

  **maneuver 1.2.1:** name = complex lane change

  **event 1.2.1.1:** condition = 1s after start of act

  **action 1.2.1.1:** increase speed by 5kph

  **action 1.2.1.2:** perform lane change to left within 5s
Example 2 – alternative: two lane changes in front of Ego vehicle, 3-lane motorway

- Upon the Ego vehicle approaching the vehicle in front, the following things shall happen:
  - player 1 in the front lane shall decelerate and merge to the left
  - with 1s delay player 2 in the front right lane shall merge to the left

**Storyboard**

**Story:** owner = Ego

**Act 1:** condition = TTC to player 1 in front right lane < 3s
  - **Sequence 1.1:** actor = player 1
    - **Maneuver 1.1.1:** name = complex lane change
      - **Event 1.1.1.1:** condition = upon start of act
        - **Action 1.1.1.1.1:** slow down by 5kph
        - **Action 1.1.1.1.2:** perform lane change to left within 4s
  - **Sequence 1.2:** actor = player 2
    - **Maneuver 1.2.1:** name = complex lane change
      - **Event 1.2.1.1:** condition = 1s after start of act
        - **Action 1.2.1.1.1:** increase speed by 5kph
        - **Action 1.2.1.1.2:** perform lane change to left within 5s
Example 3: two lane changes in front of Ego vehicle, 3-lane motorway + emergency brake

- Upon the Ego vehicle approaching the vehicle in front, the following things shall happen:
  - player 1 in the front lane shall decelerate and merge to the left
  - with 1s delay player 2 in the front right lane shall merge to the left
  - with 2s delay Ego vehicle accelerates
  - Ego vehicle applies emergency brake

**Storyboard**

**story:** owner = Ego

**act 1:** condition = TTC to player 1 < 3s

**sequence 1.1:** actor = player 1

**maneuver 1.1.1:** name = complex lane change

**event 1.1.1.1:** condition = upon start of act

**action 1.1.1.1:** slow down by 5kph

**action 1.1.1.2:** perform lane change to left within 4s

**sequence 1.2:** actor = player 2

**maneuver 1.2.1:** name = complex lane change

**event 1.2.1.1:** condition = 1s after start of act

**action 1.2.1.1.1:** increase speed by 5kph

**action 1.2.1.1.2:** perform lane change to left within 5s

**sequence 1.3:** actor = $owner

**maneuver 1.3.1:** name = accelerate

**event 1.3.1.1:** condition = 1s after end of maneuver 1.2.1

**action 1.3.1.1:** accelerate by 2m/s² to v_{Target} = v_{Current} + 5 m/s

**act 2:** condition = TTC to player in front < 1.5s

**sequence 2.1:** actor = $owner

**maneuver 2.1.1:** name = emergency brake

**event 2.1.1.1:** condition = upon start of act

**action 2.1.1.1:** apply full brake (or: apply deceleration command for 9m/s²)
Example 4: „Lane change right/left“

- see figure

**storyboard**

**story**: owner = void

**act 1**: condition = start of simulation

**sequence 1.1**: actor = player 2

**maneuver 1.1.1**: name = keep distance to Ego

**event 1.1.1.1**: condition = upon start of act

**action 1.1.1.1.1**: keep 3s distance to Ego vehicle

**act 2**: condition = TTC to Ego < 1.5s

**sequence 2.1**: actor = player 1

**maneuver 2.1.1**: name = acquire speed

**event 2.1.1.1**: condition = upon start of act

**action 2.1.1.1**: acquire \( v_{\text{Target}} = v_{\text{Ego}} + 5\text{m/s} \) with \( 3\text{m/s}^2 \)

**act 3**: condition = longitudinal distance to Ego > 2m

**sequence 3.1**: actor = player 1

**maneuver 3.1.1**: name = lane change right

**event 3.1.1.1**: condition = upon start of act

**action 3.1.1.1**: perform lane change by 1 lane to right within 2s

**event 3.1.1.2**: condition = 0s after termination of event 3.1.1.1

**action 3.1.1.2.1**: perform lane change by 1 lane to left within 2s

**event 3.1.1.3**: condition = 2s after termination of event 3.1.1.2

**action 2.1.1.3.1**: continue in autonomous mode
Example 5: „Overtaking“

- see figure

storyboard

story: owner = player 1

act 1: condition = start of simulation

sequence 1.1: actor = $owner

maneuver 1.1.1: name = accelerate, stay in Ego lane

event 1.1.1.1: condition = upon start of act

action 1.1.1.1: acquire $Target = \text{v}_\text{Ego} + 5.6\text{m/s} \text{ with } 4\text{m/s}^2$

action 1.1.1.2: keep lane

event 1.1.1.2: condition = after termination of action 1.1.1.1

action 1.1.1.2.1: keep speed

act 2: condition = TTC to Ego < 0.742s

sequence 2.1: actor = $owner$

maneuver 2.1.1: name = perform two lane changes

event 2.1.1.1: condition = upon start of act

action 2.1.1.1: perform lane change to the left within 2s

event 2.1.1.2: condition = longitudinal distance to Ego > 2m

action 2.1.1.2.1: perform lane change to the right within 1.89s
Example 6: „Traffic Jam and Corridor“

- see figure

**storyboard**

**story:** owner = void

**act 1:** condition = pass road pos S (entity = selection A, any)
- sequence 1.1: actor = by condition – triggering entity
  - maneuver 1.1.1: name = full stop
    - event 1.1.1.1: condition = upon start of act
      - action 1.1.1.1.1: acquire $v_{\text{Target}}= 0.0$ with $3\text{m/s}^2$
      - action 1.1.1.1.2: keep lane

**act 2:** condition = upon start of simulation
- sequence 2.1: actor = player P
  - maneuver 2.1.1: name = follow Ego
    - event 2.1.1.1: condition = upon start of act
      - action 2.1.1.1.1: keep 1.5s distance to Ego
      - action 2.1.1.1.2: keep lane

**act 3:** condition = Ego speed < $30\text{m/s}$
- sequence 3.1: actor = player P
  - maneuver 3.1.1: name = use corridor
    - event 3.1.1.1: condition = upon start of act
      - action 3.1.1.1.1: intercept path P1
      - action 3.1.1.1.2: keep lane
      - action 3.1.1.1.3: keep speed
Questions

• **Vehicle parameterization**
  • What are the basic parameters that can be configured by most of the tools?
    • Type (enum)
      • passenger car
      • truck
      • bus
      • etc.
    • Geometry
      • bounding box
      • track
      • wheel base
    • Weight
    • Performance
      • engine power
      • brake power
  • What are the inputs to vehicle dynamics?
    • Physical Inputs
      • throttle
      • brake
      • steering wheel angle / steering torque
      • gear
    • Abstract Inputs
      • desired acceleration
      • desired steering
OpenSCENARIO – Specification

Questions

• **Driver parameterization**
  • What are the basic parameters that can be configured by most of the tools?
    • Interaction with Traffic and Infrastructure
      • look-ahead (time vs. distance)
      • preferred distance to players ahead (time headway)
      • tendency to overtake
      • respect for traffic signs
        • speed from traffic rules
  • Longitudinal Control
    • speed keeping (rate / amplitude of oscillation)
    • preferred longitudinal acceleration
    • preferred longitudinal deceleration
  • Lateral Control
    • look-ahead (time vs. distance)
    • lane keeping (rate / amplitude of oscillation)
    • preferred lateral acceleration
    • duration of lane change
  • Vehicle Control
    • use of indicators
Questions

• Parameters
  • How to represent parameters?
    • Hierarchical names, e.g. object1.driver.desiredSpeed
    • Rule book required
  • How to parameterize a whole scenario
    • Logical scenarios vs. specific scenarios?

• Actions
  • Default behavior of objects without actions:
    • autonomous?
    • deterministic (keep initial velocity vector)?
    • Explicit or implicit transition between deterministic / autonomous modes?
      • what happens if a „speed change“ action is over?
      • what happens if autonomous mode for one axis is disabled and no action is defined to replace the corresponding input?
  • Level of autonomy even in deterministic cases (e.g. change lane while keeping speed)?
    • autonomous mode per control axis?
      • what level of detail?
  • Distinction between endless and finite actions?
  • What shall be part of the action "traffic jam"?
OpenSCENARIO – Specification

Questions

• Conditions
  • Shall "parameter conditions" have access to all state variables of the scenario?
  • Which expressions shall be supported?

• Positioning
  • World position vs. inertial position vs. GPS position?

• Initialization
  • Speed vs. velocity?

• Waypoints
  • Transition of waypoints: to next vs. to current?

• More?
  • sure, but it's up to you now....
OpenSCENARIO in PEGASUS...
Key Requirements

- Compatibility with test specification database
  - Derive logical and specific scenarios
  - Store specific scenarios

- Create parameterizable scenarios
  - Instantiate specific scenarios from these

- Replay of simulation and real test data
  - time-stamped trajectories
What’s next?
Next Steps

• **Step 1: Consolidation (now)**
  - Use MindMap as primary specification document
  - Provide draft E schema to core team
  - Harmonize draft E by the end of August 2016
  - Make schema and mindMap available as rev. 1.0 on website

• **Step 2: Implementation (Aug – Dec 2016)**
  - Identify parties starting with preliminary implementation of draft E
  - Define reporting mechanism for test runs
  - Provide first version of validator
    • VIRES

• **Step 3: Verification (Dec 2016 – Feb 2017)**
  - Define approx. five benchmark scenarios (core team)
  - Provide benchmark scenarios as OpenSCENARIO files (.xosc) on website
  - Collect results from early adopters (mid Nov. 2016)
    • Videos
    • logs as CSV files
  - Make results available via website

• **Step 4: Maintenance Phase (2017+)**
That's it!

Thank you for your attention!

Questions?