OpenScenario Workshop

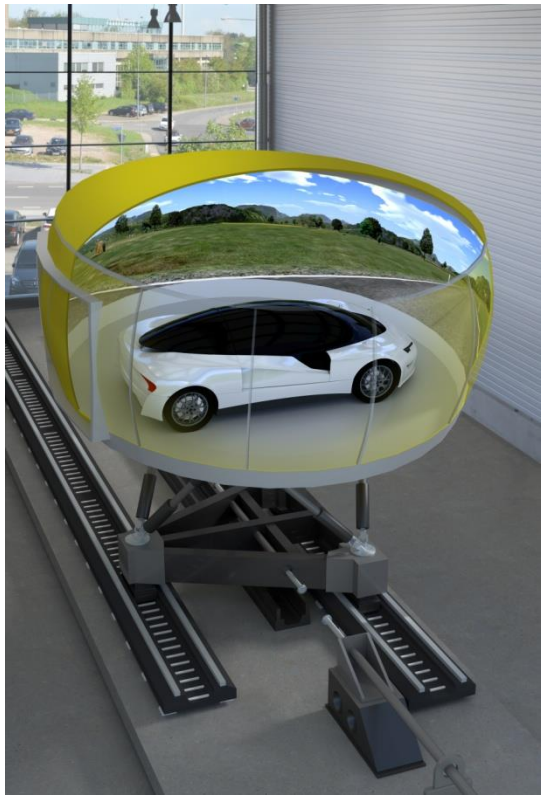
Use Cases und Requirements

Munich, 17. September 2018

Robert Krajewski, M.Sc.

Institute for Automotive Engineering

OpenScenario - Use Cases



Driving Simulator

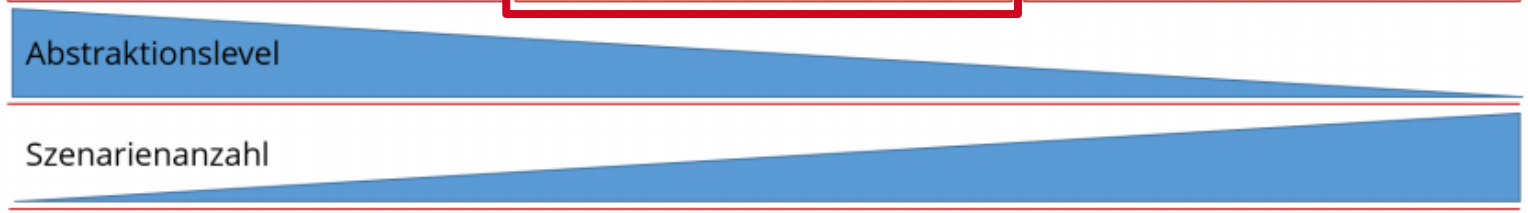
PEGASUS

Scenario Database

Test Track

Definition of Scenario Types

Funktionale Szenarien	Logische Szenarien	Konkrete Szenarien
Basisstrecke:	**Basisstrecke:**	**Basisstrecke:**
3-streifige Autobahn in Kurve Begrenzung auf 100 km/h durch Verkehrszeichen rechts und links	Breite Fahrstreifen [2,3..3,5] m Kurvenradius [0,6..0,9] km Pos_Verkehrszeichen [0..200] m	Breite Fahrstreifen [3,2] m Kurvenradius [0,7] km Pos_Verkehrszeichen [150] m
Stationäre Objekte:	**Stationäre Objekte:**	**Stationäre Objekte:**
-	-	-
Bewegliche Objekte:	**Bewegliche Objekte:**	**Bewegliche Objekte:**
Ego, Stau; Interaktion: Ego in Manöver „Annähern“ auf mittlerem Fahrstreifen, Stau zähfließend	Stauende_Pos [10..200] m Stau_Geschw. [0..30] km/h Ego_Abstand [50..300] m Ego_Geschw. [80..130] km/h	Stauende_Pos 40 m Stau_Geschw. 30 km/h Ego_Abstand 200 m Ego_Geschw. 100 km/h
Umwelt:	**Umwelt:**	**Umwelt:**
Sommer, Regen	Temperatur [10..40] °C Tröpfchengröße [20..100] µm	Temperatur 20 °C Tröpfchengröße 30 µm



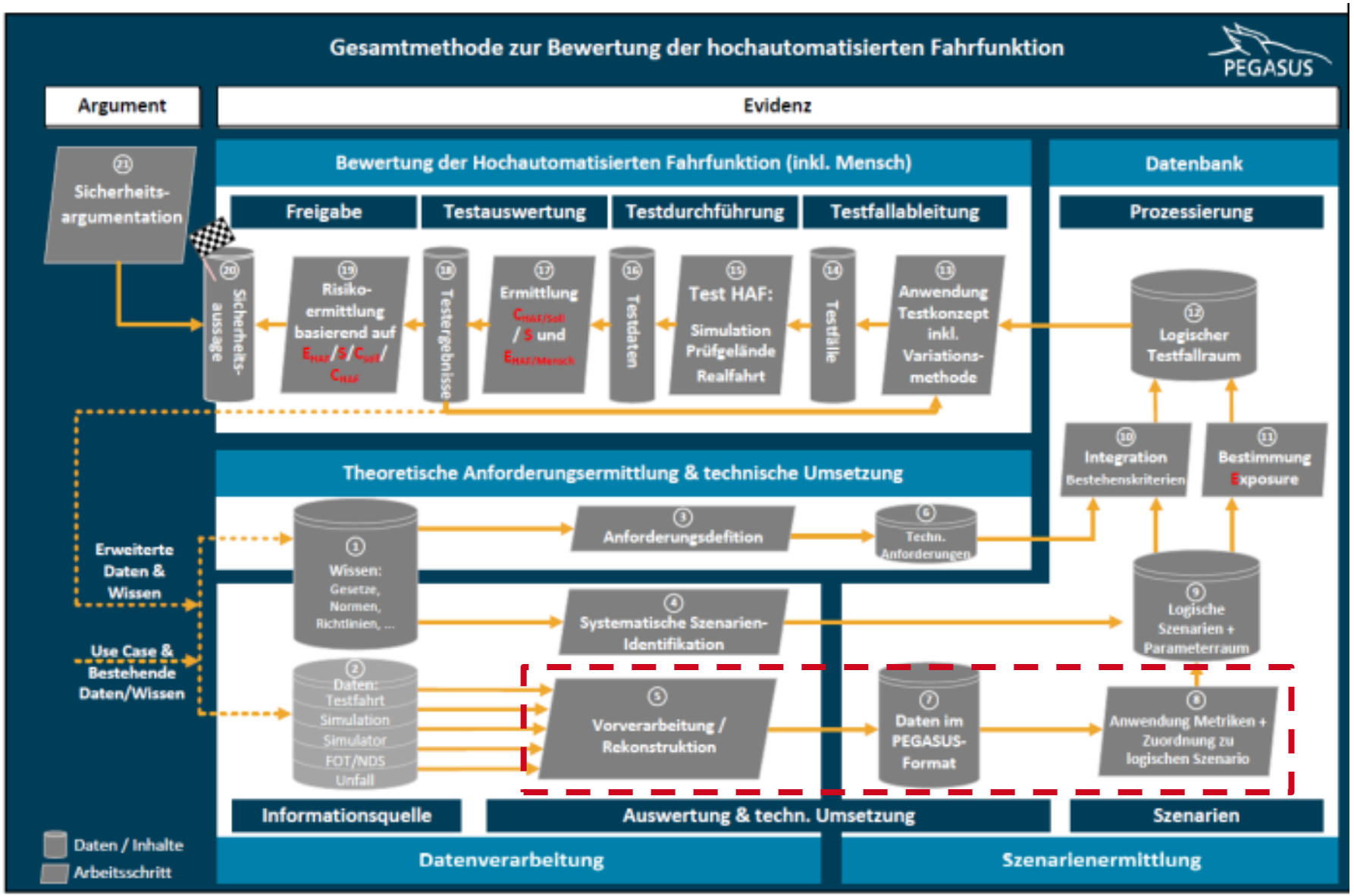
Abstraktionsebenen von Szenarien:
Funktionale Szenarien beinhalten sprachliche Beschreibung.
Logische Szenarien beschreiben Parameterbereiche im Zustandsraum.
Konkrete Szenarien stellen einen Repräsentanten aus einem logischen Szenario dar.

Bagschik (2017)

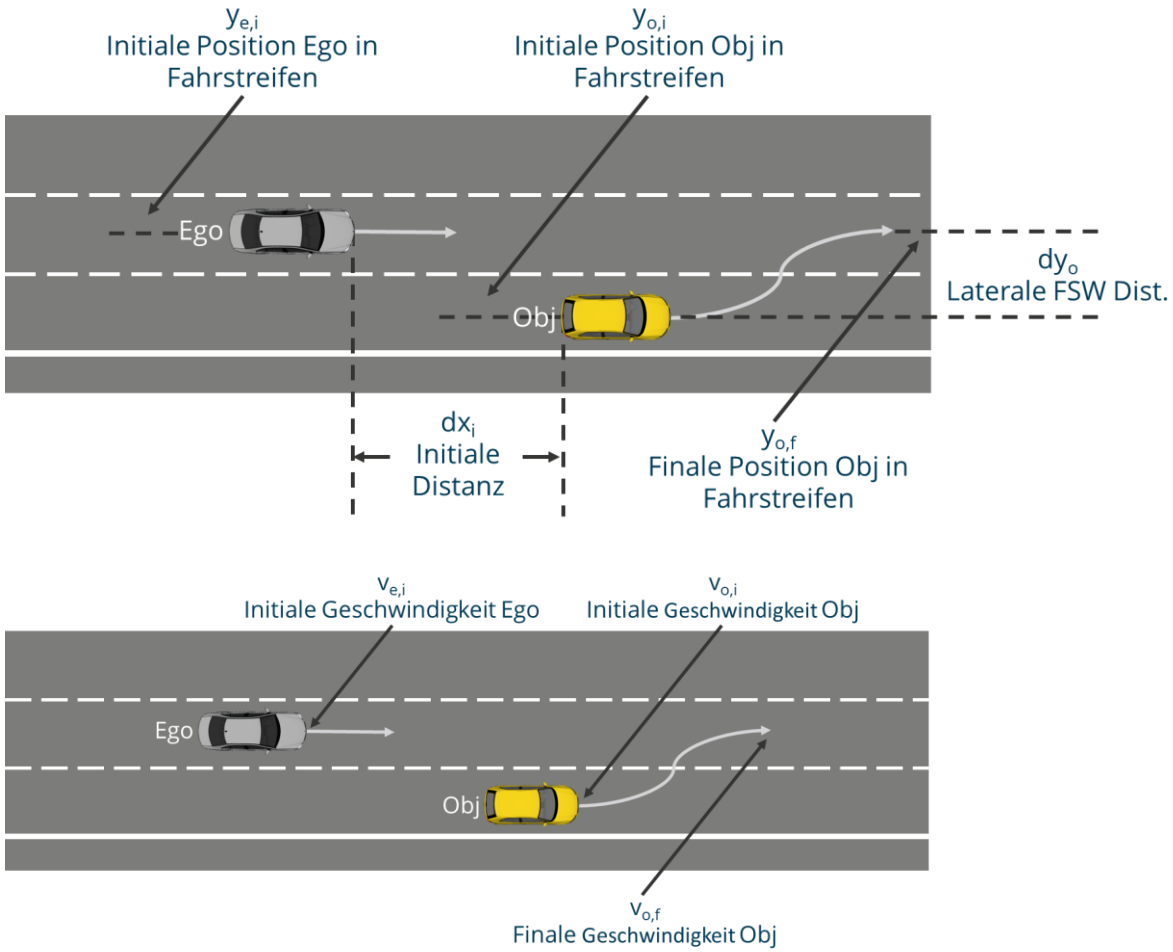
Requirements derived from PEGASUS

- Reproducibility and coverage of relevant scenarios
- Precisely describe frequency distributions of parameters
- Intuitive use of parameters
- Support for custom maneuvers and high-level scenario elements
- Support for open source driver models

PEGASUS - Overview

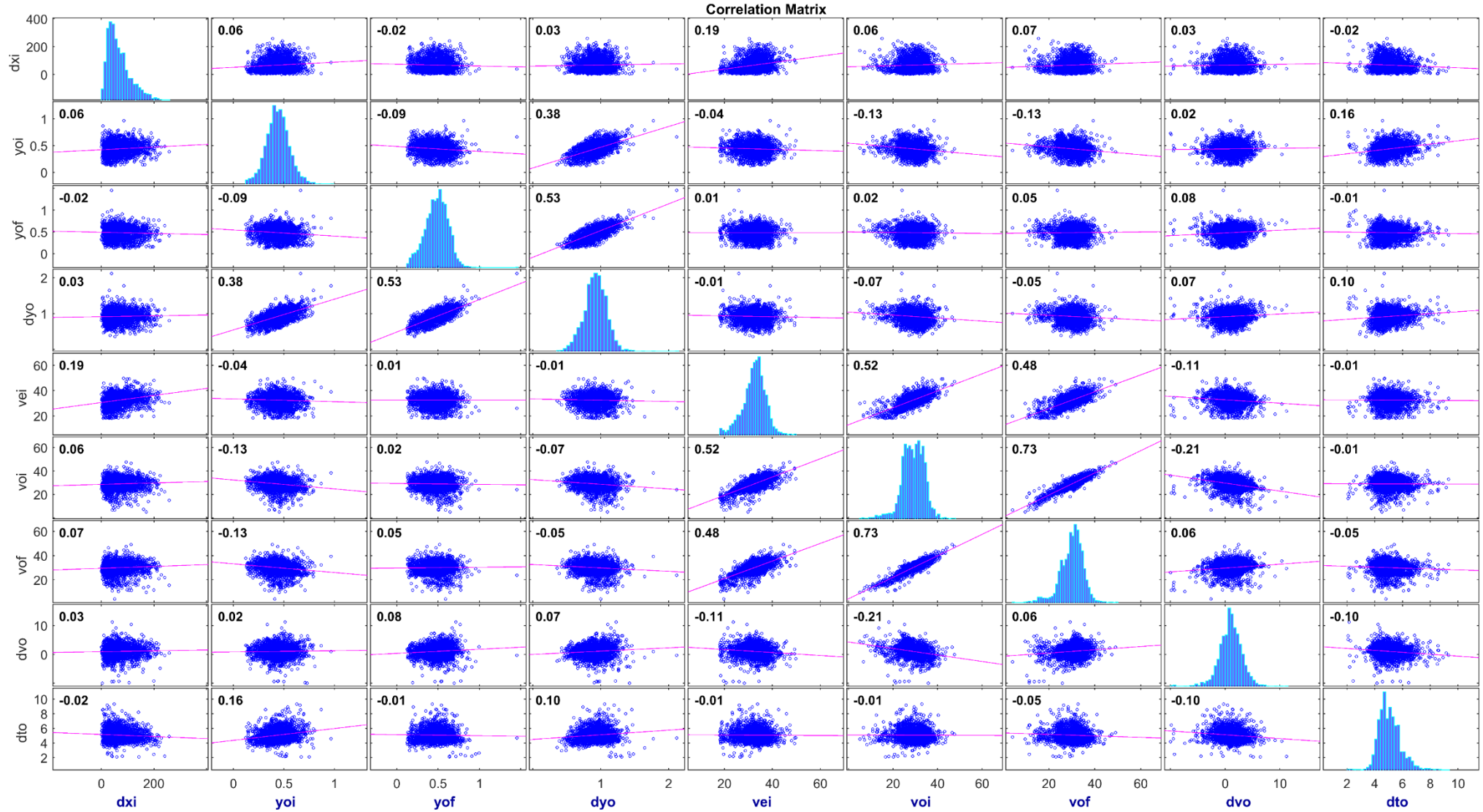


PEGASUS – Example Scenario: Cut-in



	Description
dx_i	Initial Distance
$y_{e,i}$	Initial Position Ego
$y_{o,i}$	Initial Position Obj
$y_{o,f}$	Final Position Obj
$v_{e,i}$	Initial Speed Ego
$v_{o,i}$	Initial Speed Obj
$v_{o,f}$	Final Speed Obj
dt_o	Duration Cutin

PEGASUS – Parameter Frequency Distributions and Correlations



Requirements derived from PEGASUS

- Reproducibility and coverage of relevant scenarios
- Precisely describe frequency distributions of parameters
- Intuitive use of parameters
- Support for custom maneuvers and high-level scenario elements
- Support for open source driver models

Requirements for Describing Parameter Distributions

- In addition to the scenario itself, it must also be possible to specify the distributions of the parameters.
  - Existing format created by Till Menzel (Ifr) for PEGASUS
  - Format implemented by multiple test automation tool vendors
  - Specification of parameter distributions
    - parametric (Gaussian, Uniform)
    - discrete (histograms)

```
<parameter name="$L4_Ego_Lateral_Offset_Initial" unit="m" type="double">
   <distribution>
      <normalDistribution expectedValue="0.5" variance="0.01"/>
   </distribution>
</parameter>
```

  - Specification of simple boundary conditions
  - Specification of pairwise parameter correlations

Requirements derived from PEGASUS

- Reproducibility and coverage of relevant scenarios
- Precisely describe frequency distributions of parameters
- Intuitive use of parameters
- Support for custom maneuvers and high-level scenario elements
- Support for open source driver models

Description of Complex Scenarios

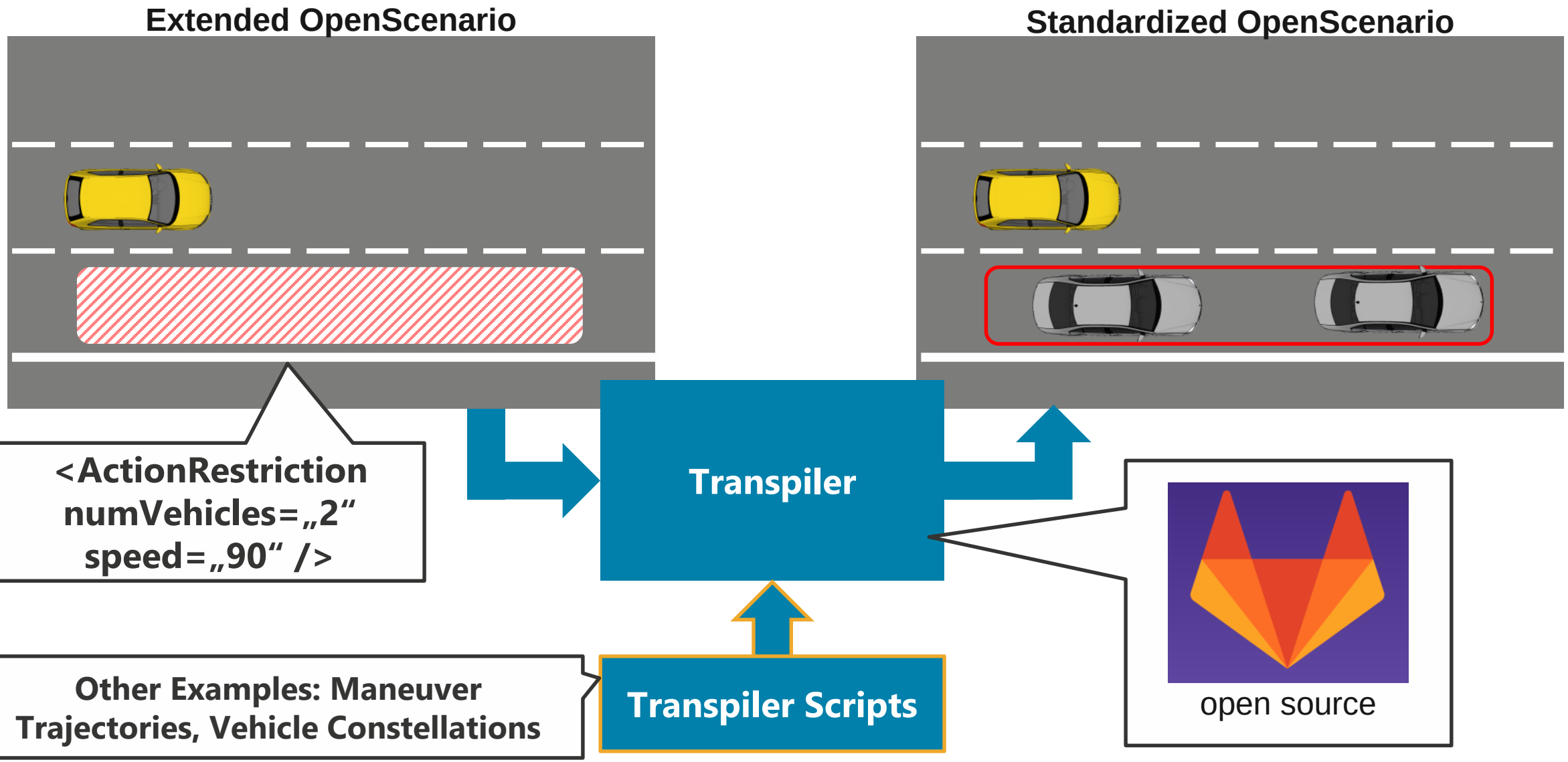
- For the description of logical scenarios,
  - only as many parameters as necessary and
  - parameters as intuitive as possible

  should be used.
- Currently, however, auxiliary parameters must be inserted in non-trivial scenarios.
- Example: An object vehicle should have a 10m/s higher speed at the end of the scenario than the ego vehicle at the beginning of the scenario.
- Current solution: Introduce an additional parameter $FinalObjectSpeed
  - Parameter unnecessary for the users
  - Calculation must be carried out separately
- Proposal: (Re-)Use of existing parameters (Value=„$InitialEgoSpeed + 10“)
  - Intuitive, since only user-relevant parameters are necessary
  - Additional intuitive parameters like $FinalDeltaSpeedObj could be added(Value=„$InitialEgoSpeed + $FinalDeltaSpeedObj“)

Requirements derived from PEGASUS

- Reproducibility and coverage of relevant scenarios
- Precisely describe frequency distributions of parameters
- Intuitive use of parameters
- Support for custom maneuvers and high-level scenario elements
- Support for open source driver models

Transpiler for a simple description of complex scenarios



Overview Transpiler

- Advantages:
  - Freedom to quickly add extensions to OpenScenario such as maneuvers or higher level elements
  - No waiting for an update of OpenScenario, but easily test/share new ideas for the standard
  - No need for implementations of new features in the simulation environments → aligns standard and tool development
  - Reference implementation of extensions → Same results everywhere
  - Extensions can be shared e.g. via Gitlab
  - Own extensions can be used intuitively via XML nodes → one way to get to the same result
  - Own extensions can be varied based on self-defined, intuitive parameters manually or within the stochastic variation.
- Disadvantages:
  - A further process step (use of the transpiler) is necessary.
- Goals/Extension:
  - Improvement of the initial version and adding further examples

Requirements derived from PEGASUS

- Reproducibility and coverage of relevant scenarios
- Precisely describe frequency distributions of parameters
- Intuitive use of parameters
- Support for custom maneuvers and high-level scenario elements
- Support for open source driver models

Contact

Robert Krajewski, M.Sc.

Institute for Automotive Engineering (ika)
RWTH Aachen University
Steinbachstr. 7
52074 Aachen
Germany

Phone +49 241 80 25611
Fax +49 241 80 22147

Email Robert.Krajewski@ika.rwth-aachen.de
Internet www.ika.rwth-aachen.de