

Scenario space exploration for establishing the safety of automated vehicles

Hardi Hungar

Institute of Transportation Systems

German Aerospace Center (DLR)

This research was partially funded by the German Federal Ministry for Economic Affairs and Energy, Grant No. 19A15012F (PEGASUS), Grant No. 19A19004B (SET Level), and 19A19002H (VVMethoden), based on a decision by the Parliament of the Federal Republic of Germany. The responsibility for the content lies with the author.


A large, high-resolution image of the Earth from space occupies the right half of the slide. It shows a curved horizon with a blue atmosphere, white clouds, and green landmasses. The text "Knowledge for Tomorrow" is overlaid in white on the lower right portion of the Earth image.

Knowledge for Tomorrow

Automated Vehicles

SAE Level 3 or higher

ADS ("System") performs the entire DDT (while engaged)						
3	Conditional Driving Automation	The sustained and ODD-specific performance by an ADS of the entire DDT with the expectation that the DDT fallback-ready user is receptive to ADS-issued requests to intervene, as well as to DDT performance-relevant system failures in other vehicle systems, and will respond appropriately.	System	System	Fallback-ready user (becomes the driver during fallback)	Limited



J3016™

Level 3

Automated Car

• Examples

- **Highway Pilot** (Project PEGASUS)
 - Highly automated driving on a highway under regular conditions (with human backup, SAE Level 3)
- **Robot taxi**
 - Automated driving with full machine responsibility (SAE Level 4)

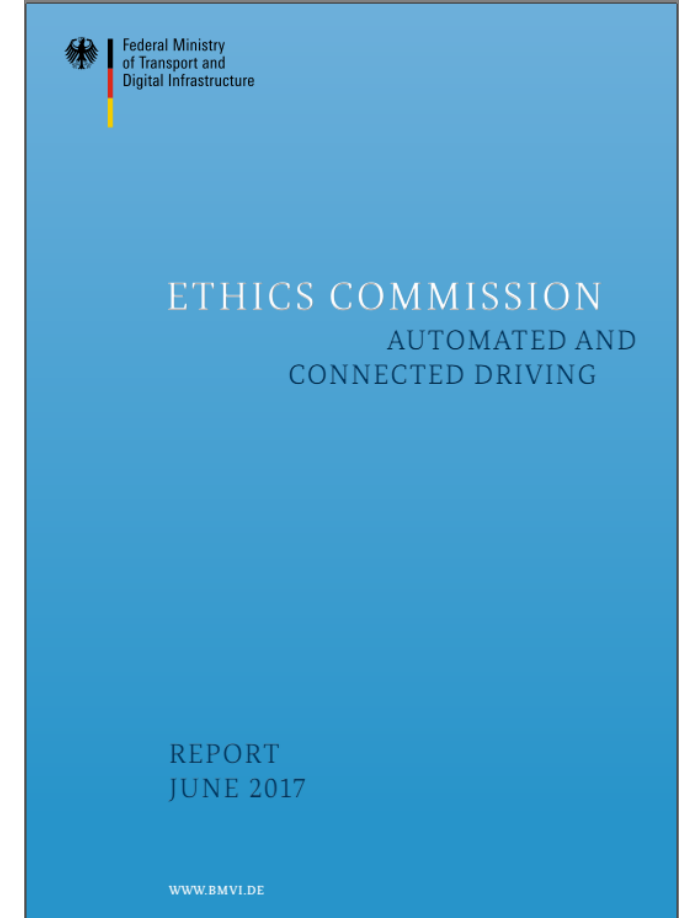


Safety target for automated driving

Ethics Commission on Automated Driving set up by the German Federal Ministry of Transport and Digital Infrastructure (BMVI)

Fully automated driving systems:

1. [...] [Their] primary purpose [...] is to **improve safety** for all road users.
2. [...] produce at least a diminution in harm compared with human driving, in other words a **positive balance of risks**.



Risk

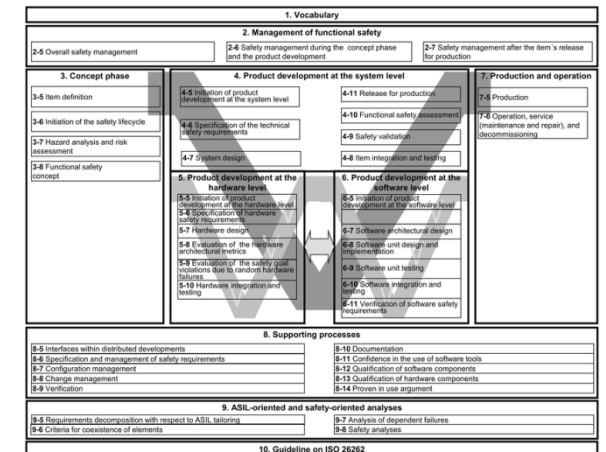
According to the standards – ISO 26262 and others

- **ISO 26262:** Standard „Road Vehicles – Functional Safety“ for developing systems with electronic elements (additional considerations: SOTIF ISO/WD PAS 21448)
 - Risk-based approach to safety

Similar to insurance risk calculation

- **Risk** $\approx \sum_{h \in H} E_h * C_h * S_h$
 - **H: hazards** - set of harmful events h
 - **E: exposure** - probability of occurrence (precisely: expected number per time unit)
 - **C: criticality** – probability of *not* avoiding an accident
 - **S: severity** of event (injuries, fatalities)

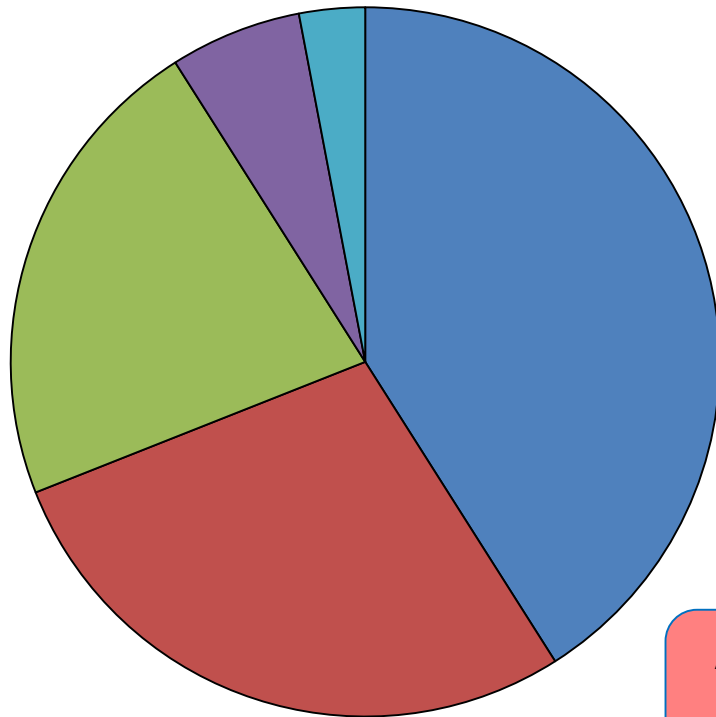
SOTIF: Road vehicles –
Safety of the intended functionality



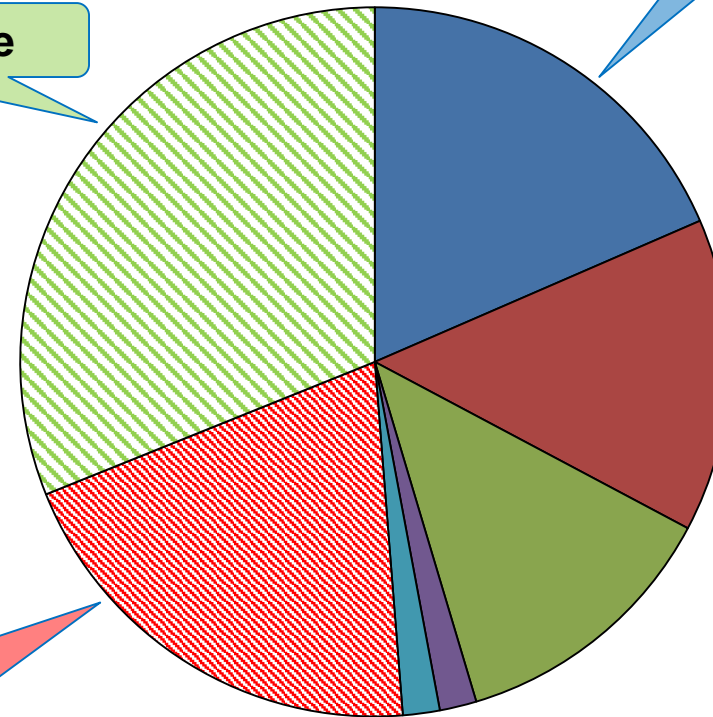
ISO 26262, Overview figure

Safety target (illustration)

Risk chart human driver



Risk chart ADS



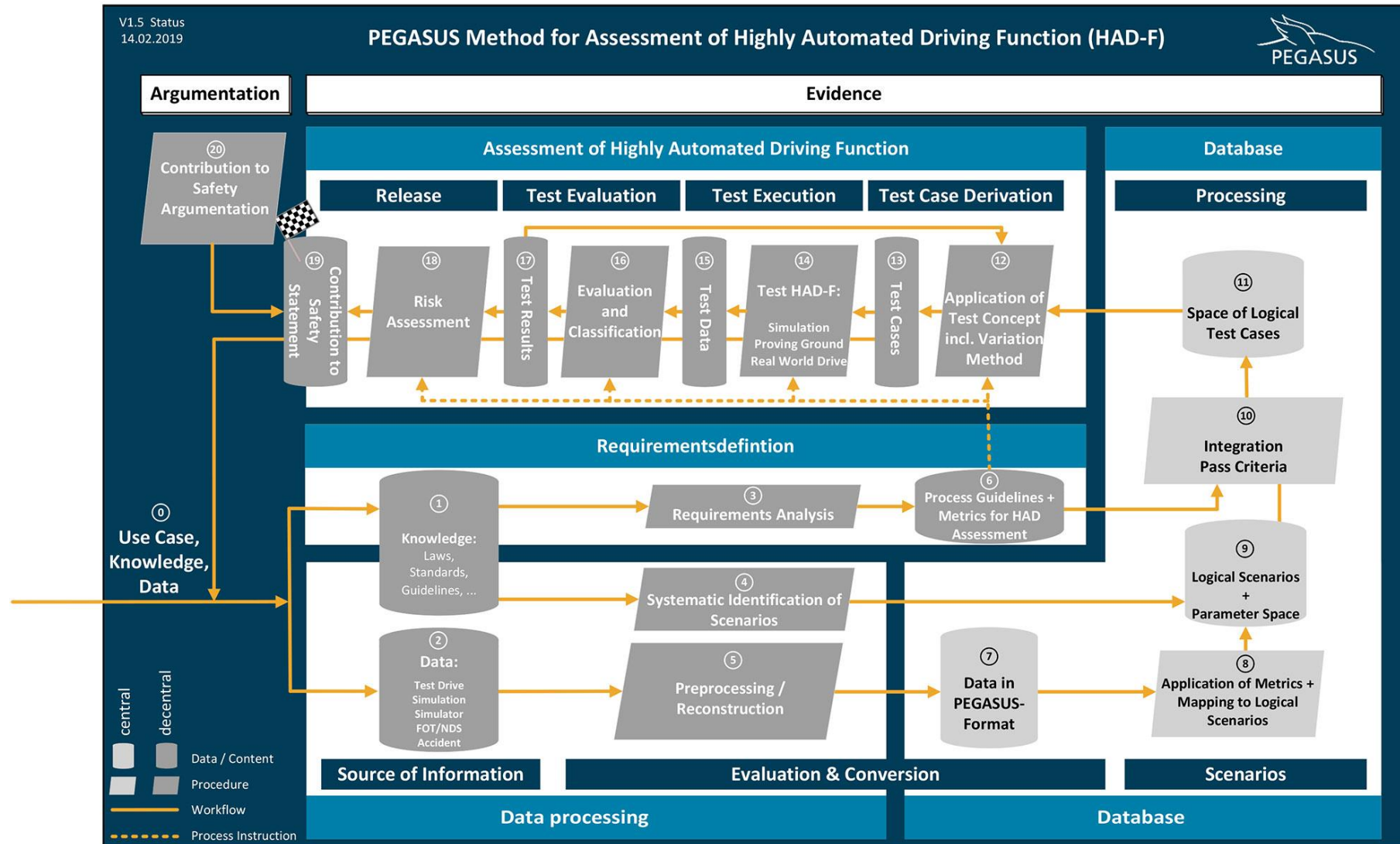
Positive balance

- Obstruction
- Lane change
- Cut in
- Following
- Weather

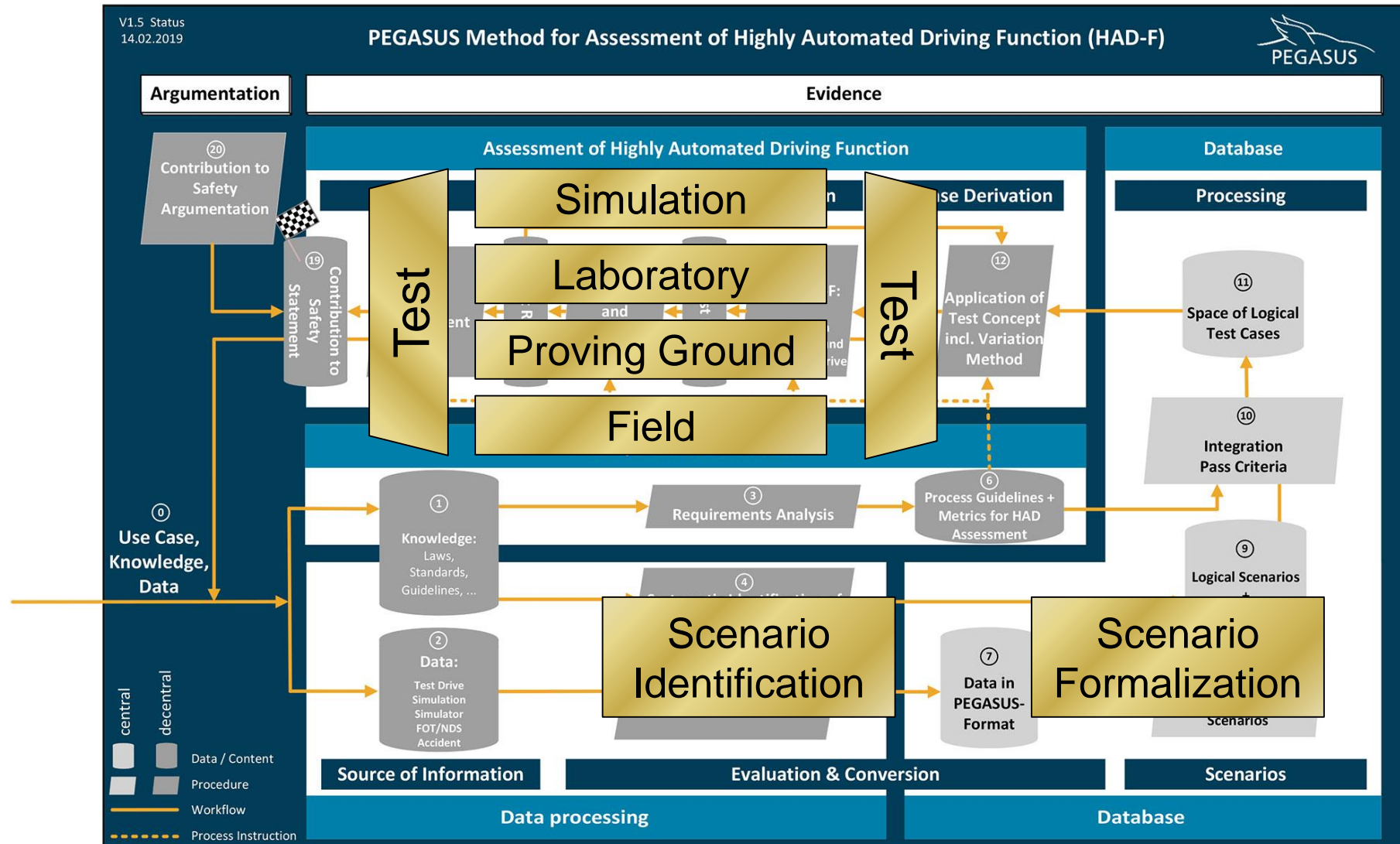
Automation errors:
Sensor error,
misinterpretation etc.

Improvement in
each category

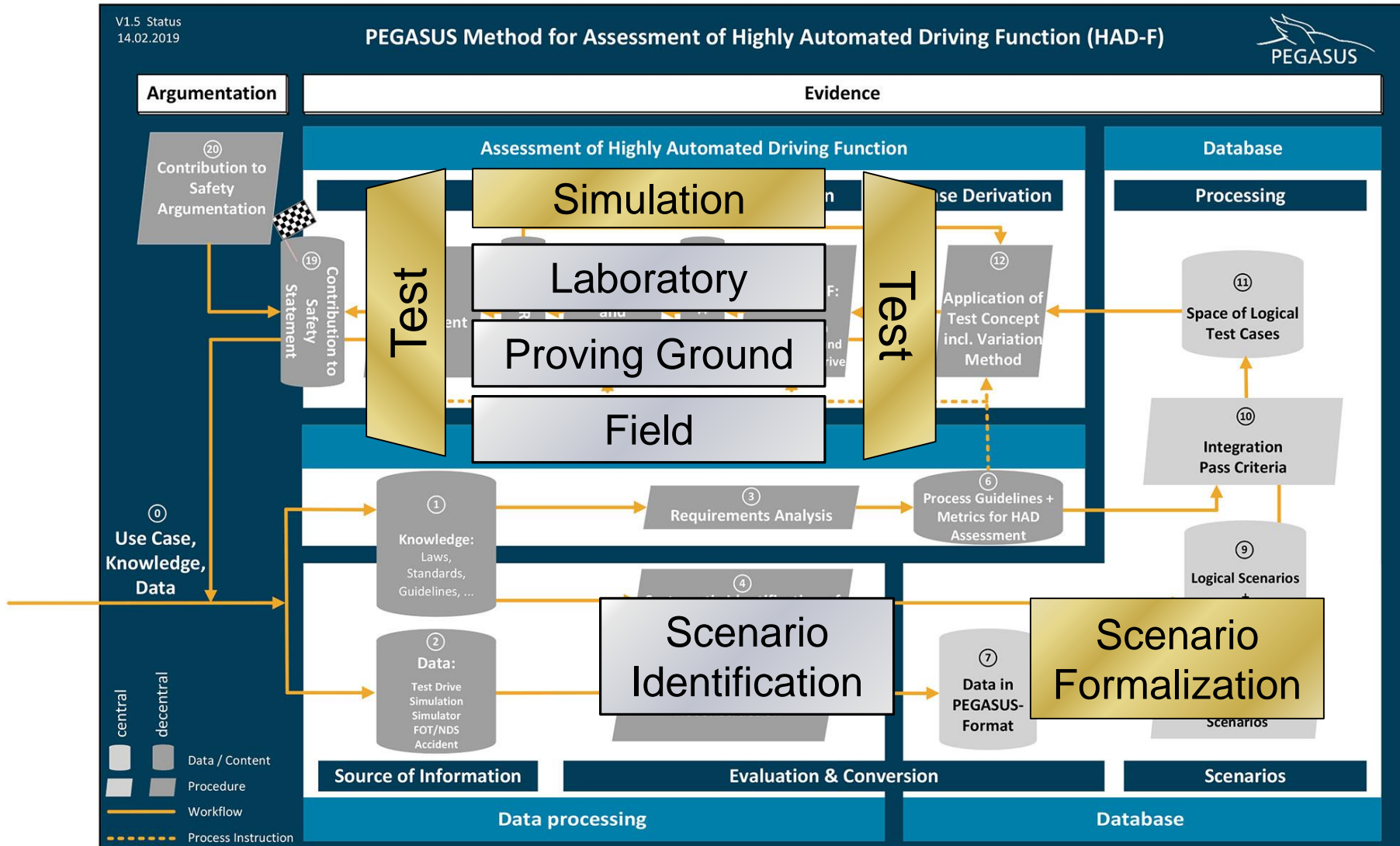
The PEGASUS Method



The PEGASUS Method

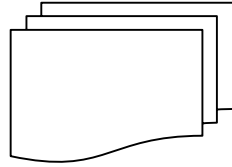


The PEGASUS Method



Risk assessment (commonly applied procedure)

- List all **hazards**
- Determine
 - **Exposure** (how often)
 - **Criticality** (accident probability)
 - **Severity** (damage)



- Sum up for overall risk

Hazard	E	C	S	Risk
Obstruction				
Lane change				
Cut in				
Cut through				
Overtaking				
Lane violation				
...				
...				
Sum				



Systematic computation of risk chart

1. **Capture** all potentially **critical** evolutions
2. **Formalize** the evolutions in precise descriptions of classes of evolutions
3. **Exhaustive testing** of evolution classes
 1. Derive concrete instantiations of a class
 2. Test concrete instances
 3. Identify regions of critical instances
4. **Analyze** the critical instances
 - Detailed evaluation
5. **Aggregate** results in risk chart

Scenario
Identification

Scenario
Formalization

Criticality
Detection

Critical Region
Analysis

Risk
computation

Hazard	E	C	S	Risk
Obstruction				
Lane change				
Cut in				
Cut through				
Overtaking				
Lane violation				
...				
...				
Sum				

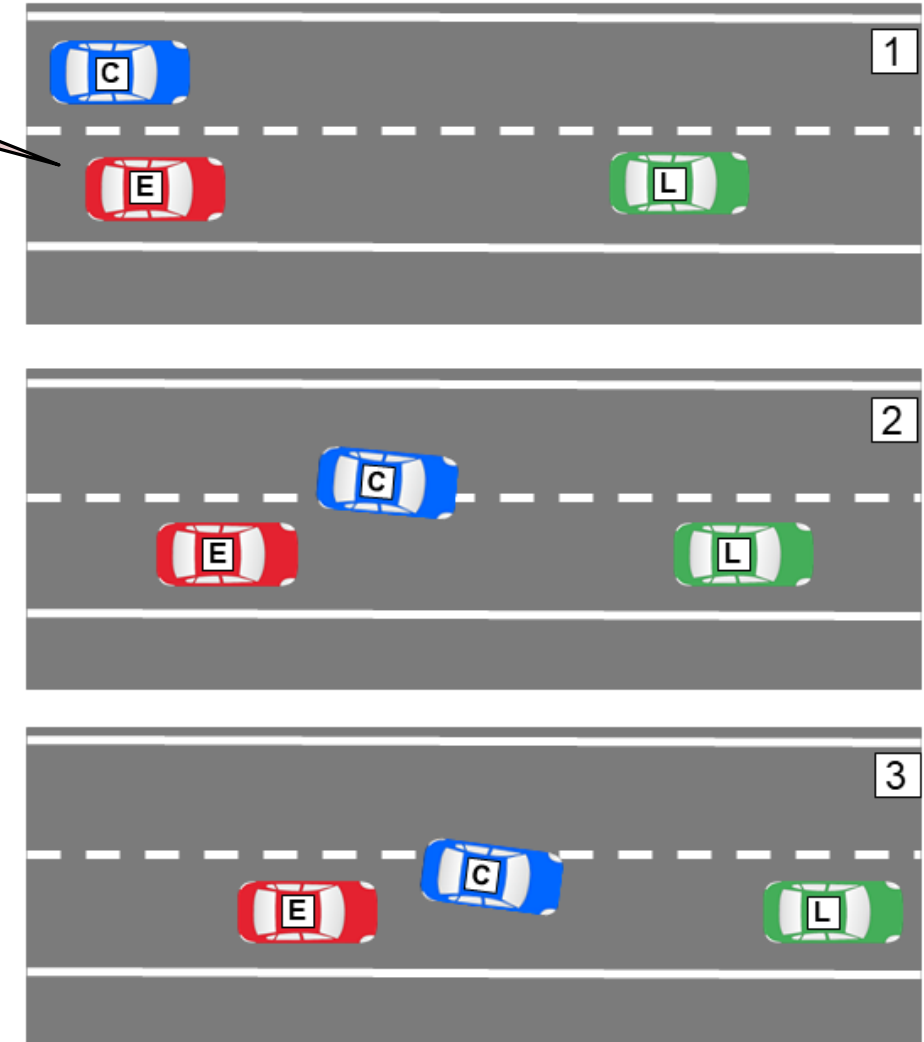


Functional scenario “cut in”

- Rough storyboard of a cut-in evolution
- Sequence of events
 - **C** is approaching on left lane
 - **C** overtakes **E**
 - **C** changes to right lane in front of **E**
- Parametrizing and varying over discrete variants yields the concrete instantiations of a “cut-in”

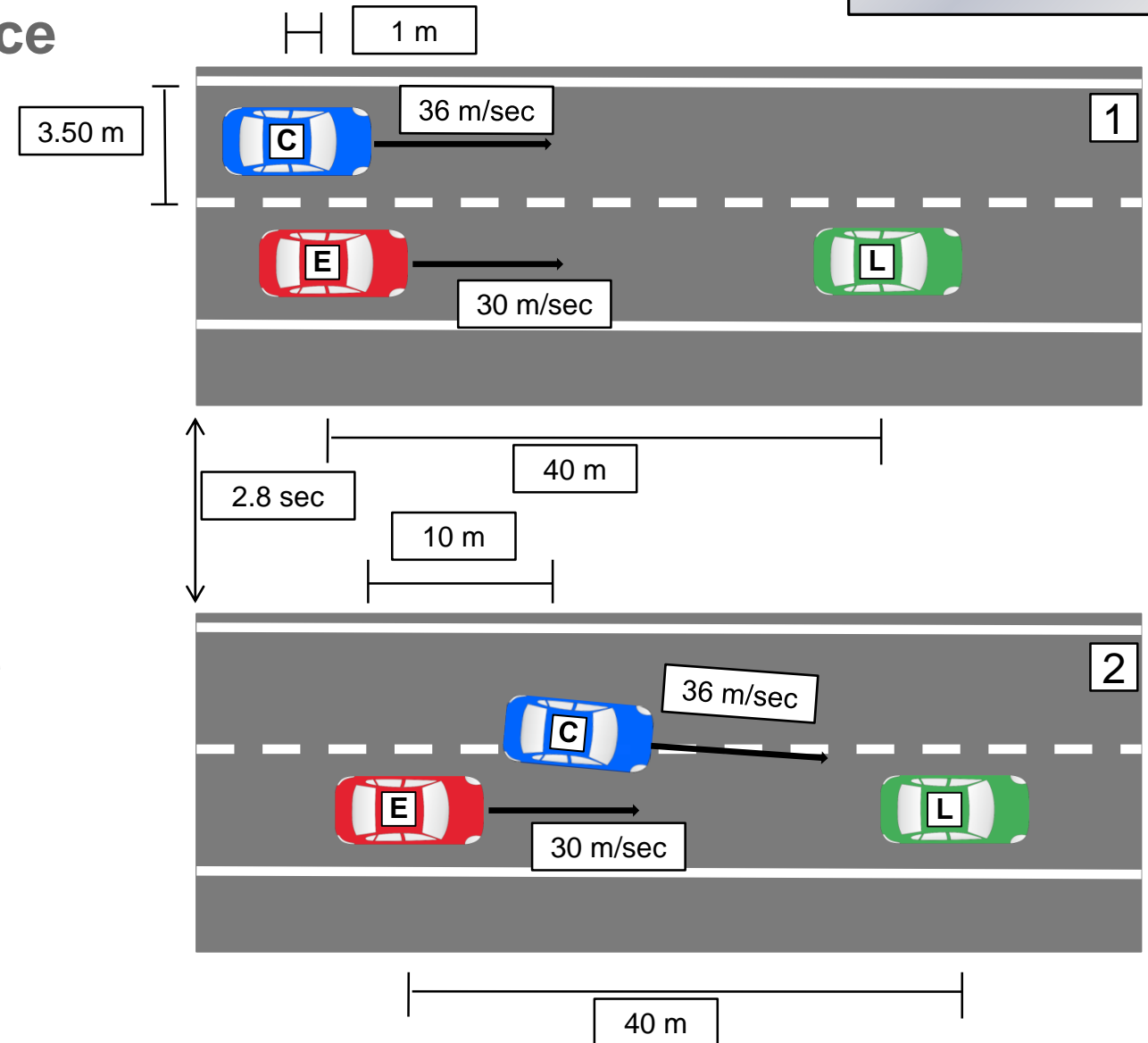
Ego vehicle

E	Ego vehicle
C	Cut-in vehicle
L	Leading vehicle



Cut in: Example of a concrete instance (single evolution)

- Deriving a concrete test scenario
 - Street dimensions
 - Relative positions of vehicles (road and other vehicles)
 - Velocities of vehicles
 - Changes of the dynamic parameters over time
- The derivation process should be systematic
 - This necessitates a formal description of scenarios



Scene: snapshot of evolution

- **Traffic participants**

- **C**, **E**, **L**

- **Positions on the street**

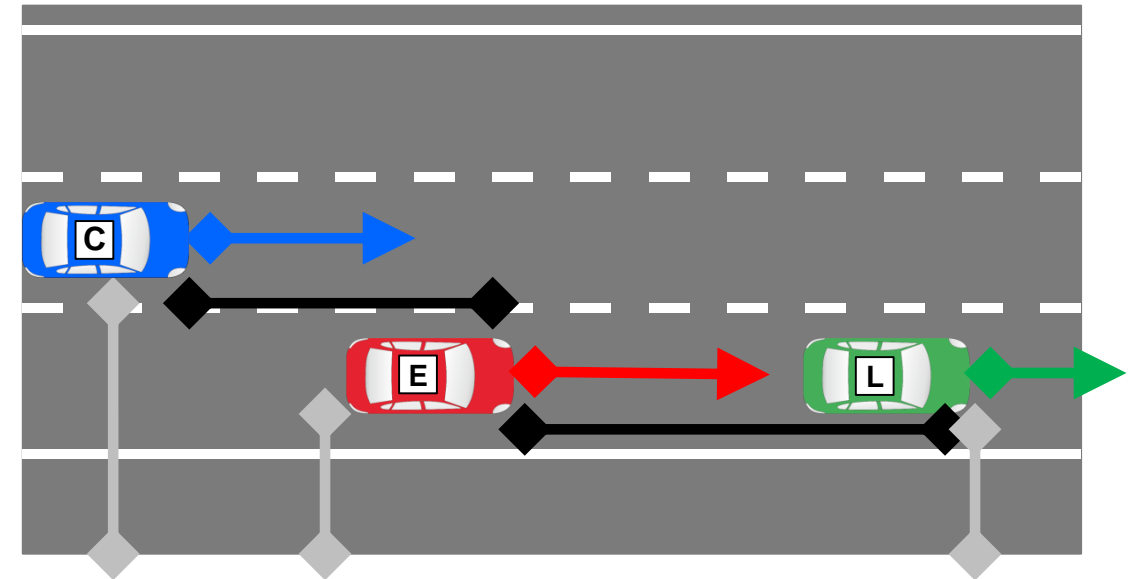
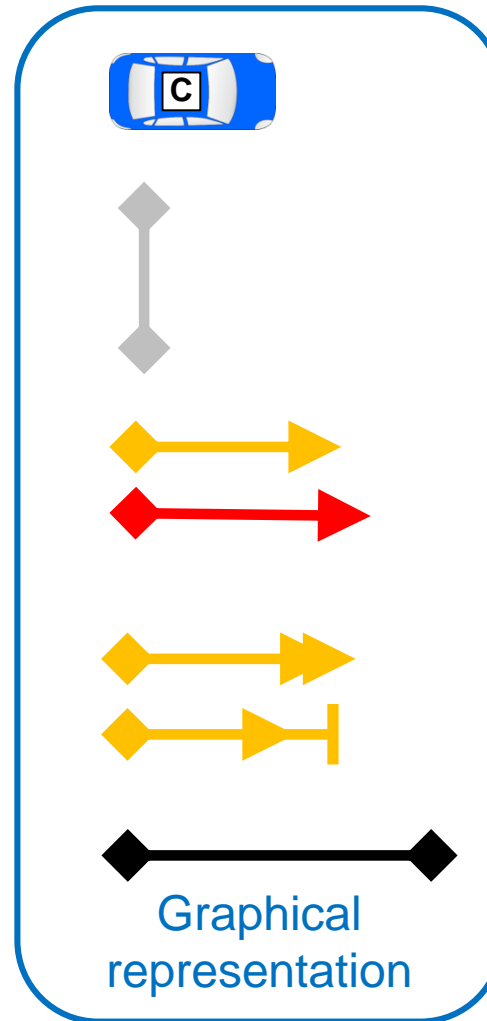
- Distance from road edge

- **Velocities**

- Acceleration
- Deceleration

- **Positions**

- (here: relative to **E**)



More complex: links
between scenes

Maneuver macros: Linking scenes to evolutions

Program-like descriptions of vehicle behavior

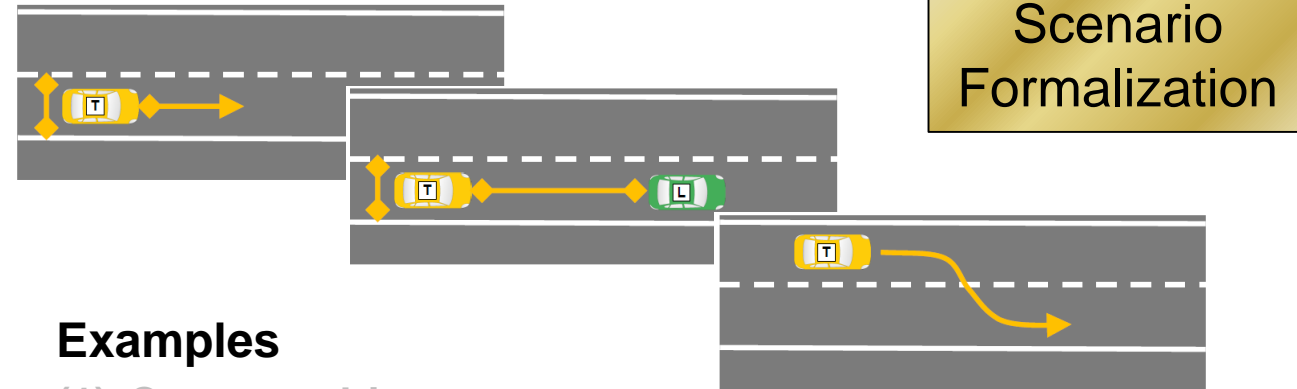
a. Geometry:

- Lateral position
- Discrete shape type: straight, sinusoidal, etc.
- Modifiers: distortions, deviations

b. Execution:

- time profile
- Completion condition (e.g.: time slot, space limitations)
- Absolute or relative to other traffic participants

c. End and exit conditions



Examples

(1) Constant drive

- Lane 1, straight, low lateral deviations
- constant velocity, low deviation
-

(2) Following

- Lane 1, straight, low lateral deviations
- Velocity adjusted on distance to lead vehicle
- Lane change of lead vehicle

(3) Lane change

- Lane 2, sinusoidal negative, low lateral deviations
- constant velocity, low deviation
- Completion of trajectory

discrete parameter

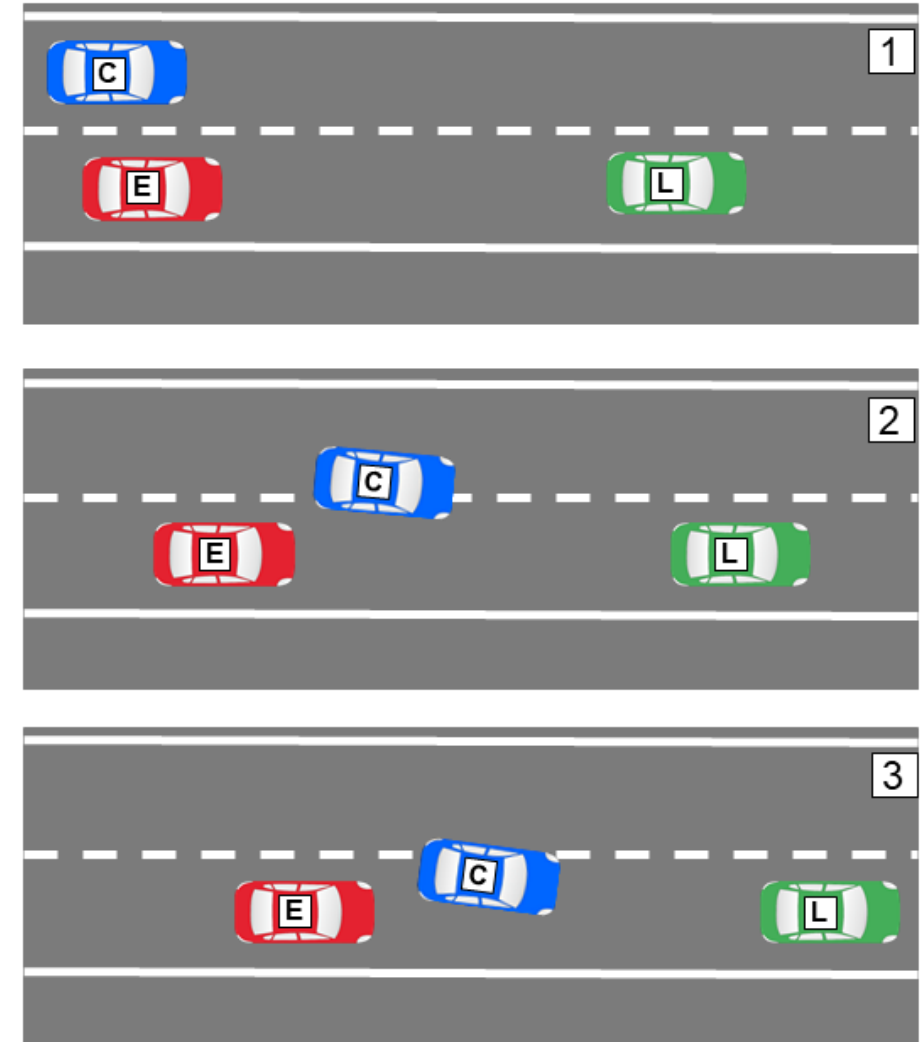
numerical parameter



Example scenario: Cut in (from left lane)

Conceptual description (not formal)

0. The ego vehicle **E** follows **L** on the right lane
C is driving on the left lane, approaching from behind
1. **C** overtakes **E**,
2. **C** changes to the right lane
2. **C** is on right lane and decelerates



Example scenario: Cut in (from left lane) Logical Description = scenario space

0. **L**: constant drive, right lane
C: left lane,
longitudinal behind **E**,
faster than **E**

Formal parameters not
shown

1. **L**: constant drive, right lane
C: left lane, overtaking **E**

Condition: **C**: longitudinal between **E** and **L**

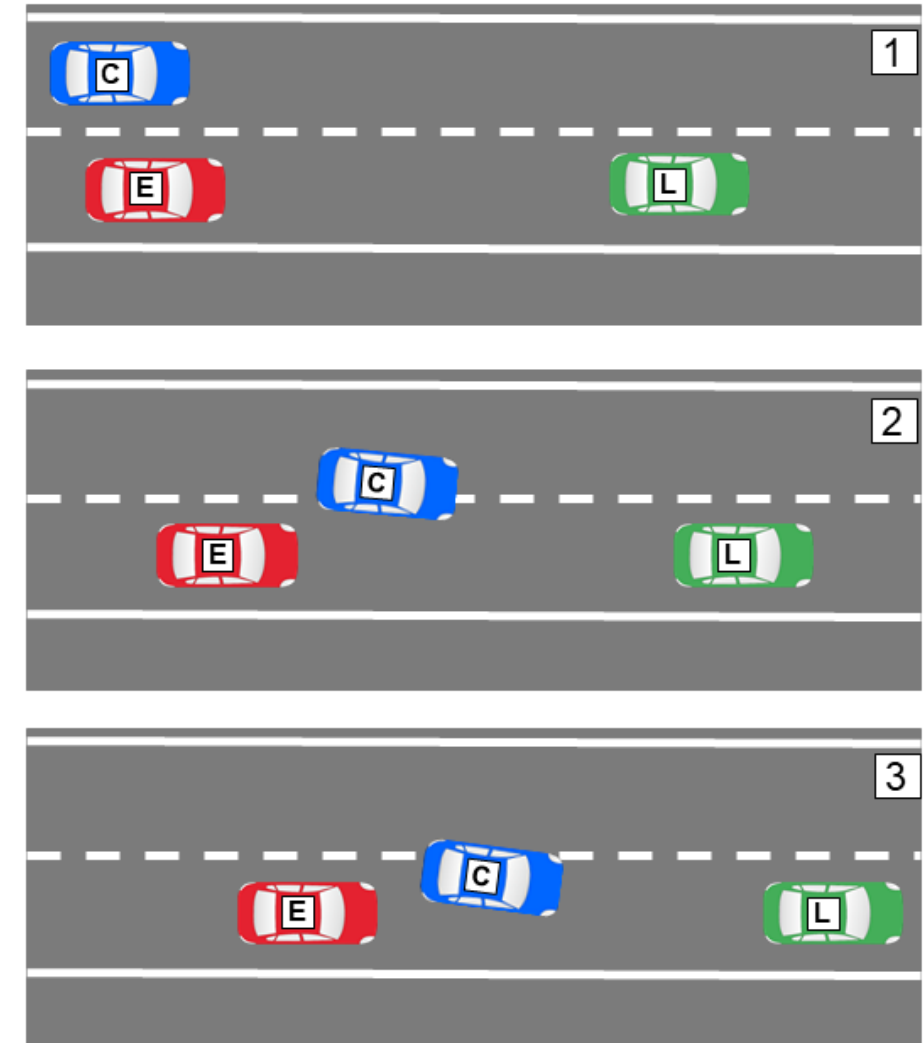
2. **L**: constant drive, right lane
C: lane change to right lane

Condition: **C**: lane change completed

3. **L**: constant drive, right lane
C: right lane, decelerating

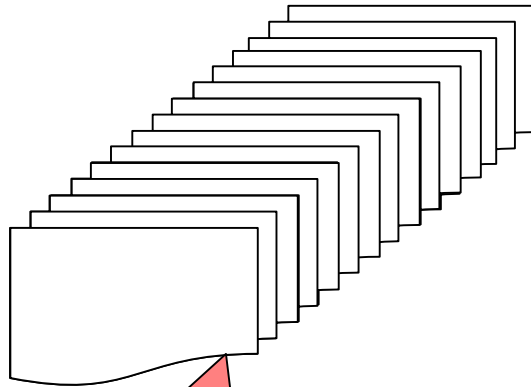
Improvements of

- OpenDRIVE
- OpenSCENARIO
- [additional formats]



Standard risk computation

- List all hazards
- Derive all concrete instances
- Determine
 - **Exposure**
 - **Criticality**
 - **Severity**



**A very long list of
concrete instances!**

- Sum up for overall risk

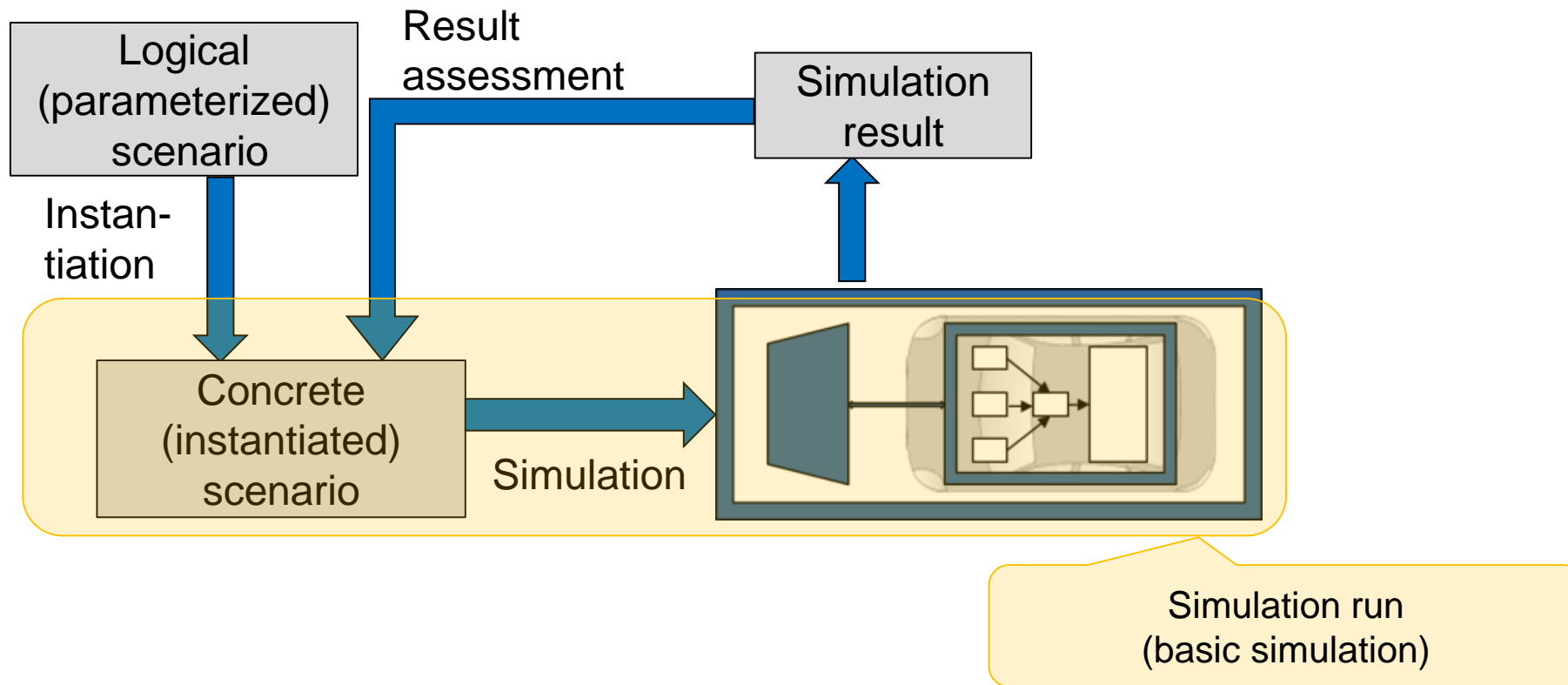
Automation needed

Hazard	E	C	S	Risk
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 100 km/h				
...				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h				
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 130 km/h				
...				
...				
...				
Sum				



Covering a scenario space by simulation

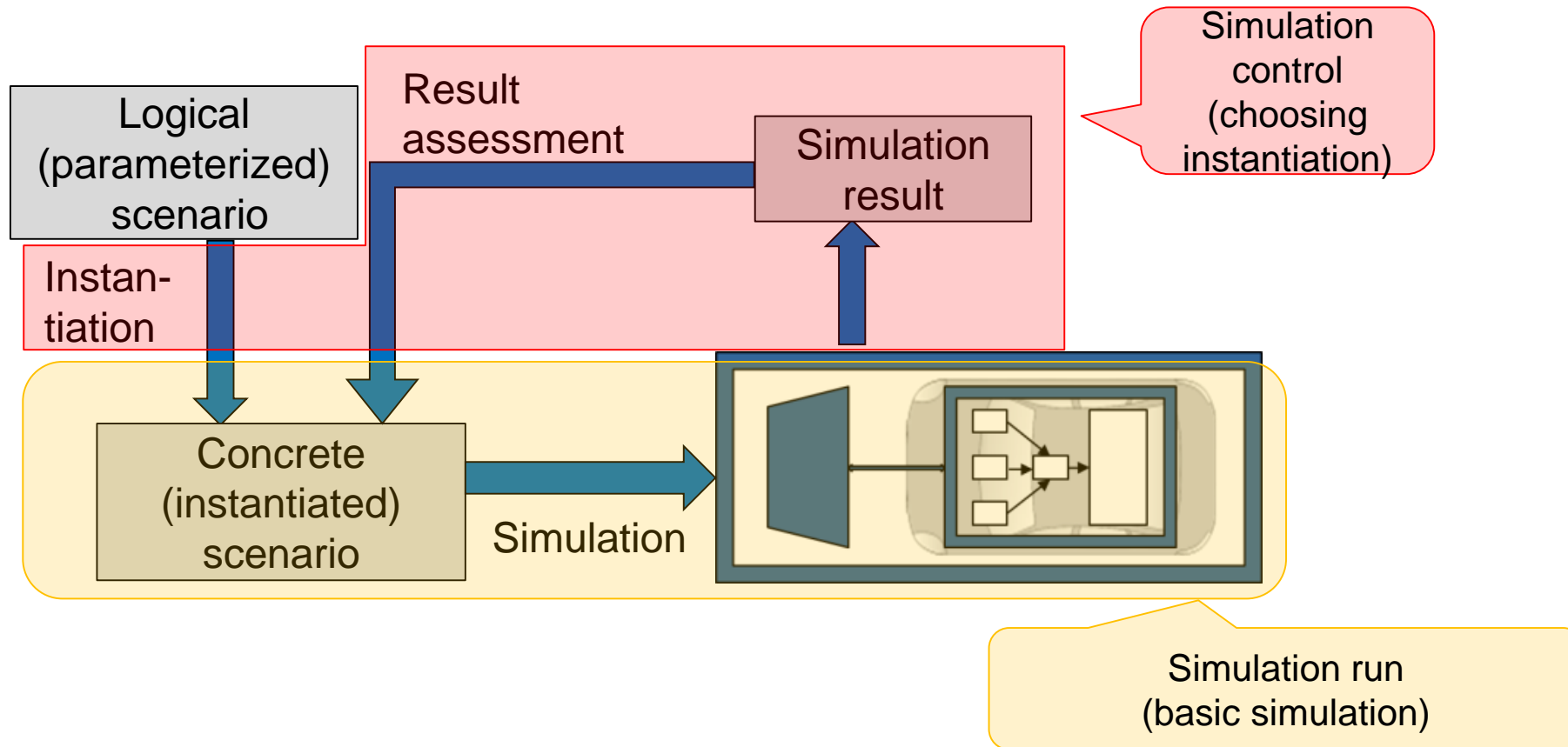
- sample use case for verification and validation -



- **Concrete Scenario**
 - Defines a particular scenario to be simulated
- **Logical Scenario**
 - Parameterized definition of a set of concrete scenarios
- **Result assessment**
 - E.g. criticality indicator

Covering a scenario space by simulation

- sample use case for verification and validation -



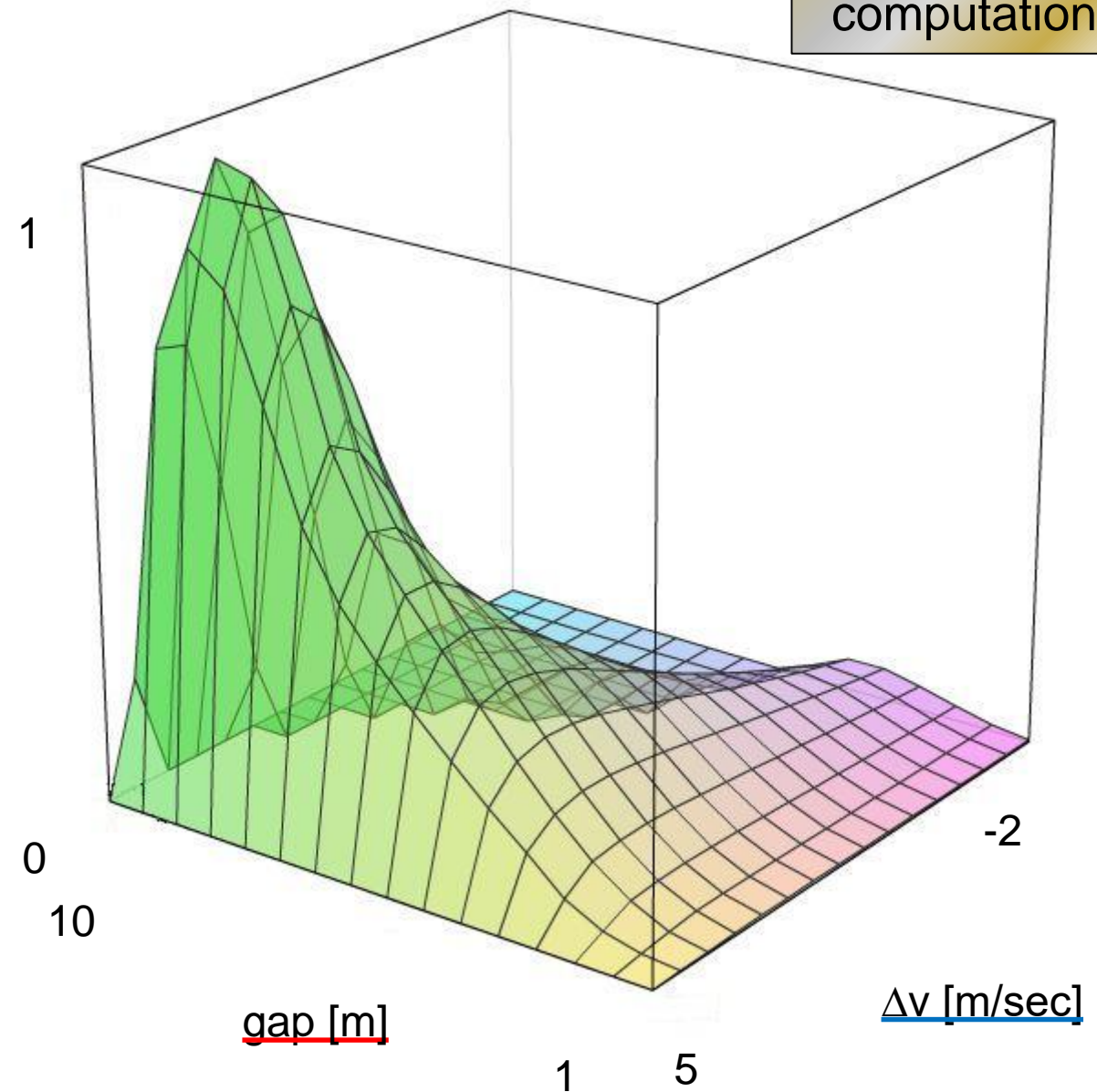
- **Concrete Scenario**
 - Defines a particular scenario to be simulated
- **Logical Scenario**
 - Parameterized definition of a set of concrete scenarios
- **Result assessment**
 - E.g. criticality indicator

Risk computation illustration

Scenario “Cut-in”: Risk

Visualization of risk of cut-in

$R \simeq$
risk



Risk computation illustration

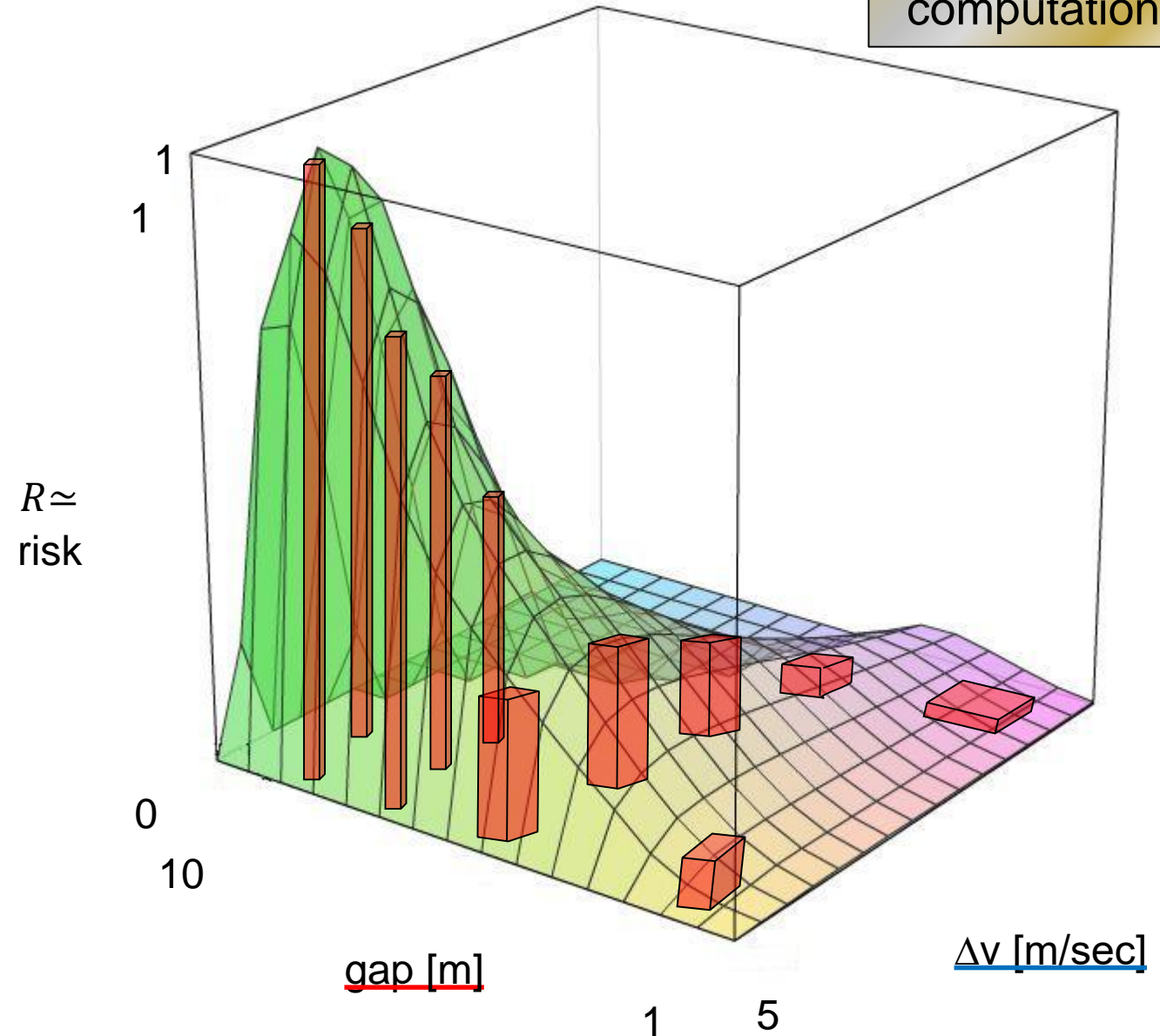
Scenario „Cut-in“:

Risk integration by simulation

Computation by approximate discrete summation

- Like Riemann integral approximation
- Each column represents the result of a test run (simulation / proving ground / field)
- Lower test density in regions with low accident probability

Similar to statistical model checking
with importance sampling



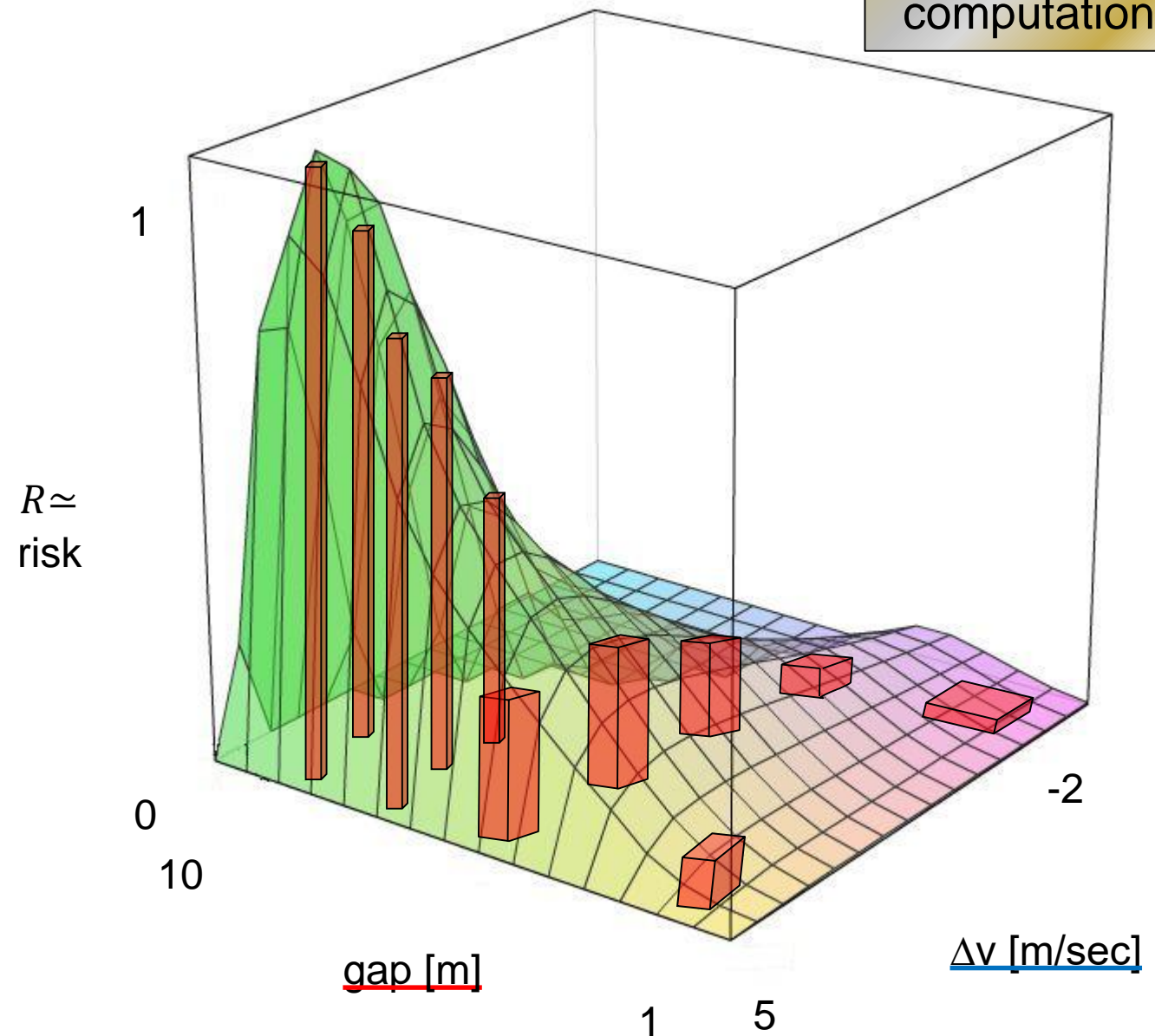
Risk computation illustration

Scenario „Cut-in“:

Risk integration by simulation

This would work, if

- we had a reliable **simulation tool**
- we had a **complete test specification**
- we could estimate the **accident probability** (“C”) of each simulated scenario
- we knew the **frequency** of each scenario (“E”)
- we could judge the accident **severity** (“S”)



Risk computation illustration

Scenario „Cut-in“:

Risk integration by simulation

This would work, if

- we had a reliable **simulation tool**
- we had a **complete test specification**
- we could estimate the **accident probability** (“C”) of each simulated scenario
- we knew the **frequency** of each scenario (“E”)
- we could judge the accident **severity** (“S”)

Assumed to be
available

To be constructed

Can be measured
by testing

Few valid data
available

Only rough
models available

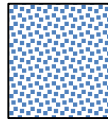


Exploration result illustration

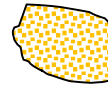
- Criticality detection as an example -

Chart of critical and uncritical parameter regions

low criticality



significant criticality

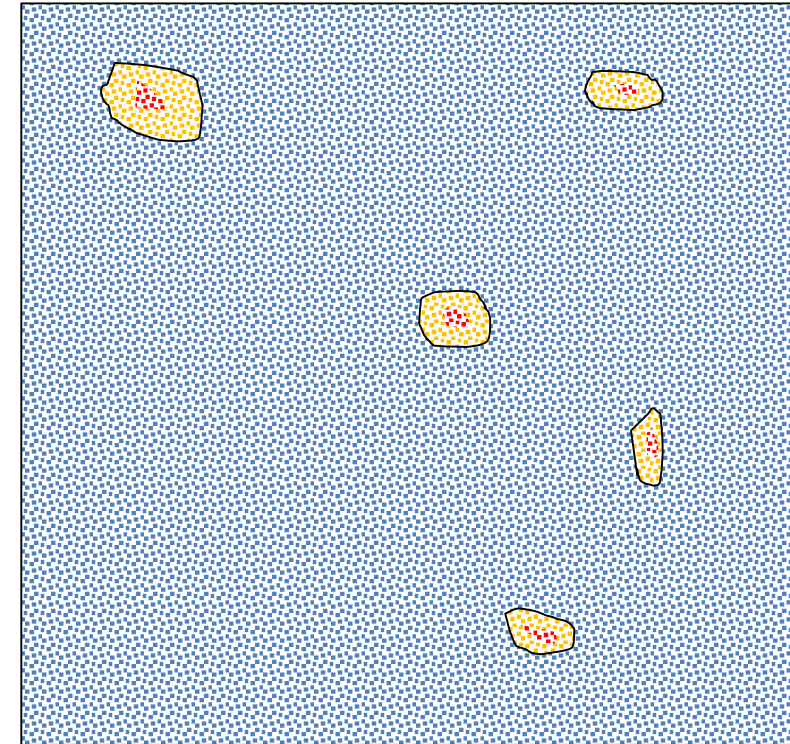


high criticality



Highly desirable:

- Coverage guarantees
- Validated simulation tools



Two-dimensional parameter space
(just for illustration)



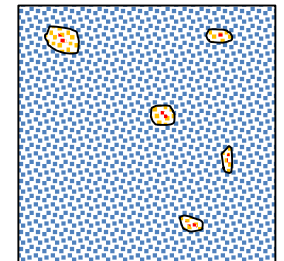
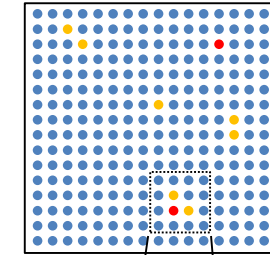
Exploration procedure illustration

- Criticality detection as an example -

Approach: The variation shall be criticality guided

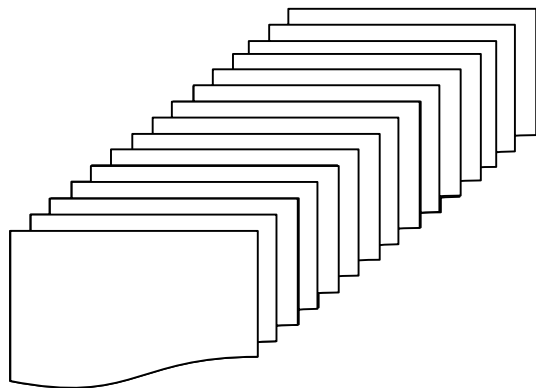
1. Detect regions of potentially significant criticality
 1. Discrete raster to cover the variation parameter space
 2. Criticality indicators to select variation parameter combinations of interest
1. Hill-climbing variation of parameters to measure areas of risk

Result: A landscape of areas of nontrivial risk



Computing the risk

- List all hazards
- Determine
 - Exposure
 - Criticality
 - Severity

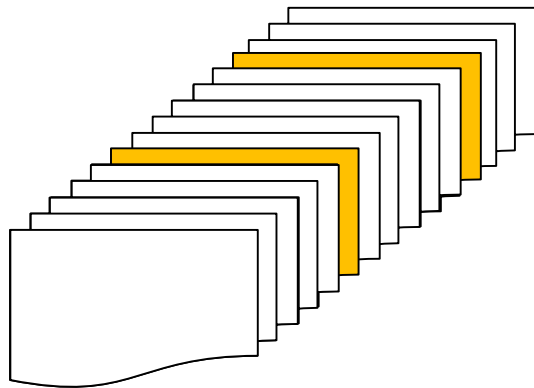


Hazard	E	C	S	Risk
...				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h				
...				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h				
...				
...				
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h				
...				
Sum				



Computing the risk

- List all hazards
- Determine
 - Exposure
 - **Criticality**
 - Severity



Formalized scenario descriptions enable automated test case generation

Determine values by automated simulation



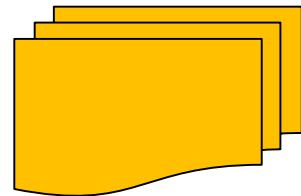
Criticality Detection

Hazard	E	C	S	Risk
...				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h		0.23		
...				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h		0.12		
...				
...				
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h		0.15		
...				
...				
Sum				



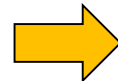
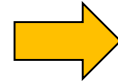
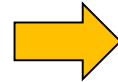
Computing the risk

- List all hazards
- Determine
 - Exposure
 - **Criticality**
 - Severity



- Extract **relevant row sets**

This is what we
detect in the
exploration



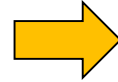
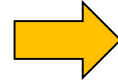
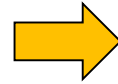
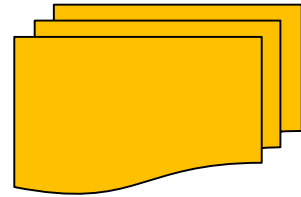
...

Hazard	E	C	S	Risk
...				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h		0.23		
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h		0.12		
...				
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h		0.15		
...				
Sum				



Computing the risk

- List all hazards
- Determine
 - Exposure
 - Criticality
 - Severity
- Extract relevant row sets
- Detailed analysis of risk in critical scenarios



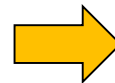
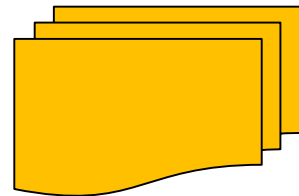
...

Hazard	E	C	S	Risk
...				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h	0.13	0.23	0.8	0.239
...				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h	0.02	0.12	1.3	0.003
...				
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h	0.01	0.15	1.4	0.002
...				
...				
Sum				



Computing the risk

- List all hazards
- Determine
 - Exposure
 - Criticality
 - Severity
- Extract relevant rows
- Detailed analysis of risk in critical scenarios
- Sum up for aggregated risk chart

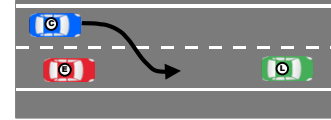


Hazard	E	C	S	Risk
...				
Cut-in by vehicle entering highway Ego: 130 km/h, Cut-in-veh.: 85 km/h	0.13	0.23	0.8	0.239
...				
Cut-in by vehicle concealed by truck Ego: 130 km/h, Cut-in-veh.: 90 km/h	0.02	0.12	1.3	0.003
...				
...				
Cut-in from left lane, decelerating Ego: 110 km/h, Cut-in-veh.: 115 km/h	0.01	0.15	1.4	0.002
...				
Sum				

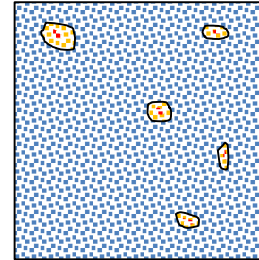


Conclusion

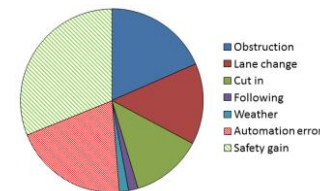
1. Capture all potentially critical evolutions in functional scenarios
2. Formalization of functional scenarios in precisely defined logical scenarios using maneuver macros
3. Identify all regions of critical scenarios by systematic testing
4. Analyze the critical regions
5. Build the risk chart by summing up the analysis results



0. L: constant drive
T: constant drive
C: lane following with goal constellation depending on (C, T, E)



...					
Cut-in by vehicle concealed by truck	0.02	0.12	1.3	0.003	
Ego: 130 km/h, Cut-in-veh.: 90 km/h					
...					
Cut-in from left lane, decelerating	0.01	0.15	1.4	0.002	
Ego: 110 km/h, Cut-in-veh.: 115 km/h					



Scenario
Identification

Scenario
Formalization

Criticality
Detection

Critical Region
Analysis

Risk
computation



Contact info

PD Dr. Hardi Hungar

German Aerospace Center (DLR)
Institute of Transportation Systems
Lilienthalplatz 7
38108 Brunswick
Germany

Hardi.Hungar@dlr.de



Project websites

PEGASUS
www.pegasusprojekt.de

V&V Methods
www.vvm-projekt.de

SET Level (under construction)
setlevel.de



Scenario space exploration for establishing the safety of automated vehicles

3rd China Autonomous Driving Testing Technology Innovation Conference, 2020

Hardi Hungar, DLR

Text notes for slides

01

I will be talking about how to use simulation to explore scenario spaces to establish the safety of automated vehicles.

02

I will be talking specifically about vehicles of SAE level three or higher. These are vehicles where the automation takes driving responsibility and the human is not more than a backup. Examples are a highway pilot - that was the case study on which the PEGASUS project developed its method-, or a robot taxi operating in the urban environment.

03

What we have to show for such applications has not been fully defined yet by the authorities. But at least in Europe, we know that we will have to prove essentially that the automation operates the car more safely than the human driver. Or, in other terms, there will be a positive balance of risk.

04

What risk is needs to be defined, of course. If we look at the available standards, we see something in way an insurance company would define risk. In short, this is probability times cost. For safety, cost means accident severity, that is injuries, casualties, and also damage to the cars.

05

Detailing the safety target, we see the picture on this slide. On the left, we see the illustration of a pie chart of accident types for human operated cars. Different categories of accidents, and the associated risk. The automated vehicle should be better than the human in each category. And even we add the category of accidents by the automation which have no counterpart in the human operated world - sensor misreadings or stupid automation decisions, or whatever, there should remain a sizable diminution of the risk. This is indicated by the sector with light green stripes in the pie chart on the right.

06

The PEGASUS project, running from 2016 to 2019, developed a method for the safety assessment. A graphical overview is shown here. More details can be found on the project's website given below.

07

I will highlight some aspects of this method, before I present how some of the steps of the method may be realized. We start on the bottom left with requirement elicitation. this identifies, among other things, the scenarios which are relevant to prove the safety of the automation. These are the potentially critical scenarios, which have to be tested.

To be able to test them, these scenarios have to be formalized, so that we can write down precise test cases. So this is the next step in the method. And then we are going to test them. By simulation, in the lab, on the proving ground and in the field. And if the outcome of these testing activities is sufficiently positive, we have reached our goal and construct a convincing safety case for the certification authorities.

08

Today, we will look at steps of scenario formalization, and on simulation, and a bit on further assessment steps.

09

The standard approach to risk assessment in the development of a safety-critical system looks as follows. One identifies all hazards, makes a list, estimate or measure the ECS values to get their contribution to the overall risk, and sums it up. Let us see how we must modify this procedure to make applicable to such complex functionalities as they are needed to realize automated vehicles.

10

Essentially, the steps stay the same, as we see here. I will show a bit of the details of the steps in the boxes of somewhat golden apparel, and just indicate what is done in the others.

11

Let us start with scenario identification. There, we write down scenarios in an abstract form, or, as such is sometimes called, as functional scenarios.

12

Such an abstract scenario stands for a lot of concrete instances, indicated here: For the cut-in example, all the ways in which this may occur in real traffic, with many variations of distances, velocities, timing.

13

To capture all these concrete instances formally, we need an appropriate language. I guess all in the audience have heard about OpenSCENARIO and OpenDRIVE - these are current standards which hopefully evolve in the not-too-far future to something which fits our needs. Some indications I will give in the next few slides. Here, we see a graphical representation of what characterizes a particular scene, a snapshot of an evolution. Every simulation will consist of a number of computed scenes.

14

To enable the simulator to perform the computation, we define how the traffic objects around our automated vehicle do behave. This can be done in the form of maneuver macros. A macro describes some action of a traffic participant, like lane following, lane change, or car following. And each macro has a set of parameters controlling the details of how it is performed. And like with macros in a programming language, we can program a scenario with such macros.

15/16

How this may look like is illustrated on the following two slides. The first shows the sequence of maneuvers, the second gets a little bit closer of how a formal description would look like. The parameters are not shown, here. In the end, we have a so-called logical scenario, with formal parameters, which defines a whole space of scenarios: For each set of parameter values, we get a concrete instance which we may use to test the performance of our automated vehicle.

17

Now, that we have the means to describe the test space, let us look at the risk computation again. If we translate our logical scenarios to lines of the risk

assessment list, we notice that this list is very, very long. No chance to fill it out manually. So we need some automation to compute the risk. Simulation is done in the computer, so we can, in principle, automate this. And even speed that up by parallelizing the computation.

18

How this may be done I will show in the following. We add a higher functionality to our simulation tool. The lower part of the picture shows the standard simulation: One concrete scenario is executed in the simulation tool of your choice. But then, to cover the whole scenario space given in the form of a logical scenario on the top left, we automate the call to the basic simulation

19

Depending on the simulation results seen so far - whether there was some criticality or not - new instances of the logical scenario are chosen, and fed to the simulation. And this is repeated until a sufficient coverage is reached.

20

Let us illustrate this process on a simple example. For ease of exposition, let us assume there are only two parameters to the cut-in.

The gap between the automated vehicle and the one cutting in, and the velocity difference.

The function shows the risk associated with the concrete scenario, that is, the combination of exposure, criticality and severity. Normed to the interval from 0 to 1. 0 means no risk, 1 means maximal risk.

This is the function would we like to measure with our simulation.

On the bottom corner, the gap is just 1 m, and the automated vehicle is 5 m/s faster than the one cutting in. Accident unavoidable, but risk is low because this will happen very rarely. On the other hand, such cut-ins might happen with a larger gap, and so we see a high risk more to the left.

21

Then we explore this parameter space, and the concrete instance are chosen depending on the risk we have measured for a particular parameter combination. When it gets interesting, we look closer, this means with a tight spacing. Where in the vast majority of the scenario space, where there is no risk, we can be sloppier.

And we compute the risk approximately, similar as one does in the Riemann definition of integration. Or, in more contemporary terms, like in statistical model checking with importance sampling.

22

That would solve our problem nicely, but ... we would need.

- a reliable simulation
- a complete test specification
- a way to estimate
 - the accident probability
 - the exposure
 - and the severity

23

We don't have that all

But let us assume we have

- the simulation, and

- the means to write a complete test specification (a collection of logical scenarios plus evaluation function)

Our simulation computes the accident probability - well, that is what our simulation shall be able to do

But we certainly do not have (yet) the two other ingredients. Enough traffic data, and good estimators of medical accident damages.

So even if we are generous about our tools, we lack essential ingredients.

24

So-what do we do instead? More modest, we just aim for identifying critical and uncritical parameter regions. That is, computing a criticality chart. On the right, you see an illustration, again for a logical scenario with just two parameters. All dots are concrete instances. Blue dots are uncritical ones. The red ones are accidents or near accidents. the yellow ones are in between, with some significant criticality but not yet an accident. And all of these yellow and red areas should be detected reliably.

25

How such a procedure might proceed I would like to indicate in the following. In a first steps, it does a rough scan of the parameter space. And wherever there is some indication that something bad might result by modifying the current parameters a little, this is done systematically, by refining the resolution. And by that, we improve our chart to the desired precision. This may sound pretty simple and obvious. But the trick is, to do it in a way that it indeed guarantees that nothing has been overlooked. And that is precisely what we are working on, and one of the persons working on that is me.

26/27

The result of this computation is then used for risk assessment. We can fill out the C-column of our risk computation table. This identifies the rows of the table where we have to look closer and estimate the contribution to the overall risk.

28/29/30

This estimation, the "critical region analysis" as I call it here, will certainly involve some manual work. But now we have to cope with only a limited set of criticality clusters, blocks of lines, and not with nearly infinitely many single lines. This gives us a chance to compute an estimation of the risk. which is hopefully than lower than the risk incurred by a human-operated car, as required for a positive risk balance.

31

Let me summarize with an overview of the method I have shown. Starting after scenario identification, I have indicated how one may be able to formalize the scenarios for a test specification, and how a chart of critical region can be drawn. And this is an important towards being able to in the end estimate the risk coming from the automated vehicles operating in its intended environment. Be this the highway, as in the PEGASUS example, or an urban area in the case for a robot taxi, or whatever.

32

On the very last slide, you see my contact details, and also the addresses of the web sites of the projects about which I have been talking today.

--

(EOF)