TRUSWORTHY EMBEDDED AI

RISK ANALYSIS AND CERTIFICATION FRAMEWORKS FOR CRITICAL TRUSTED AI APPLICATIONS

Supporting Safety Assessment of Autonomous Systems with *Papyrus for Robotics*

Morayo Adedjouma

- with contributions from Matteo MORELLI, Ansgar RADERMACHER, Fabio ARNEZ, Guillaume OLLIER, Diana RAZAFINDRABE (CEA-LIST/DILS/LSEA); EL JIHAD Hasnaa; Huascar ESPINOZA (KDT JU)

Safety and robustness — Key factors in autonomous systems innovation

- Safety of robotics applications must be guaranteed
- Legal directives and standards compliance must be fulfilled!
- Avoid emergency stops and ensure system stability

“Safety”, “Functional Safety” and “Safety of the Intended Functionality”

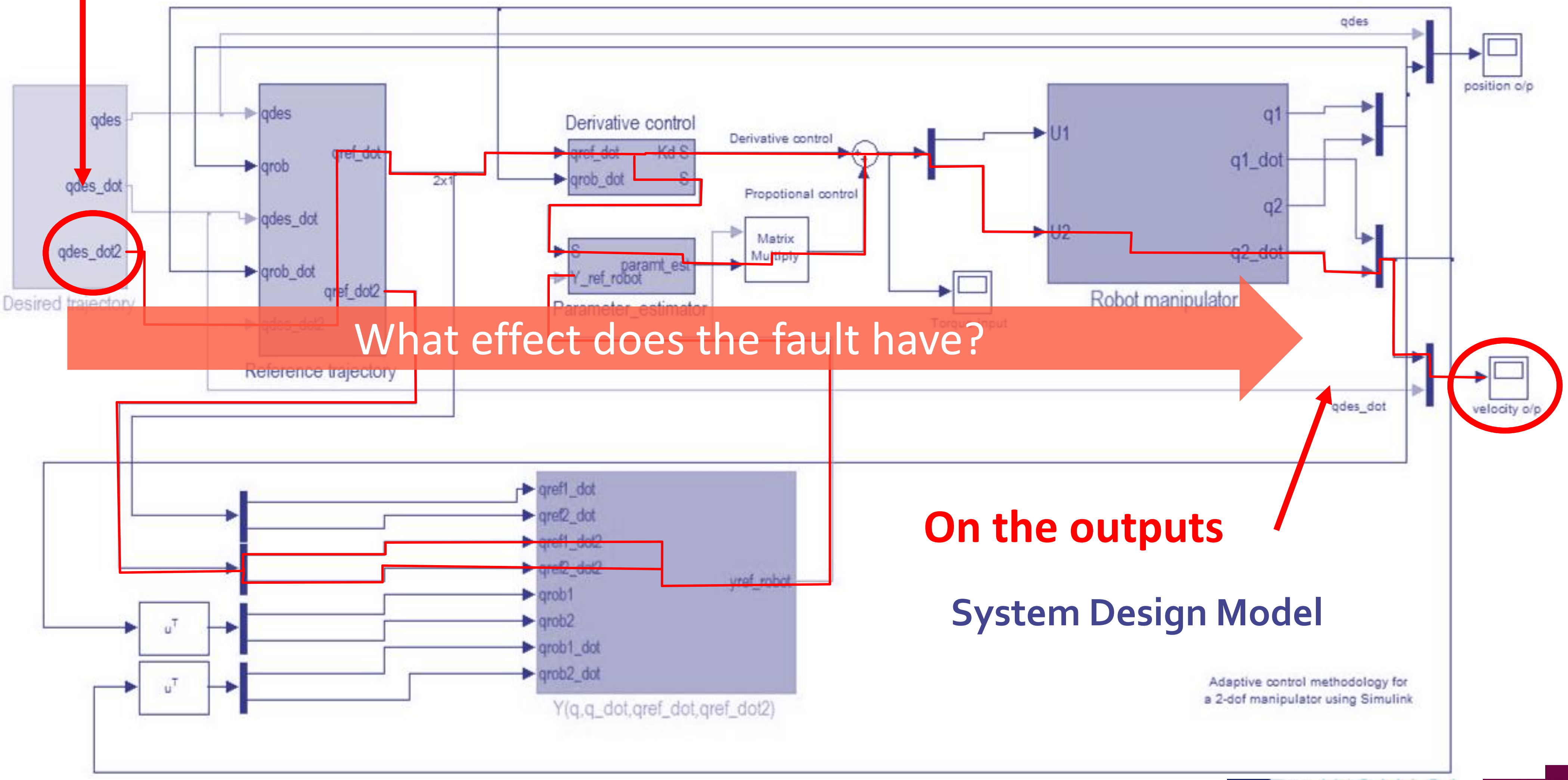
Safety is the condition of being protected from harm or other non-desirable outcomes. It can also refer to risk management.

Functional safety is the part of the overall safety of a system or piece of equipment that depends on automatic protection operating correctly in response to its inputs or failure in a predictable manner.

Safety of the Intended Functionality (SOTIF) concerns with guaranteeing the safety of a functionality that can have safety risks in the absence of a fault.

How to anticipate the fault impact on the system to ensure safety?

If a fault develops here



On the outputs

System Design Model

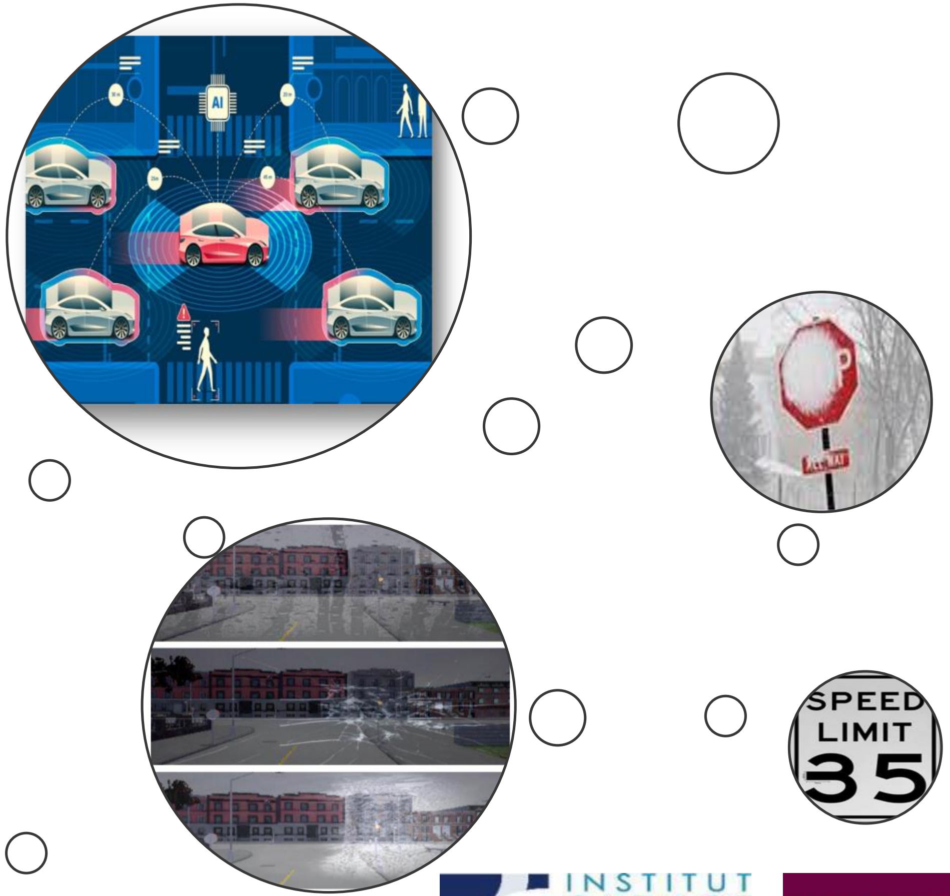
Credits: Yiannis Papadopoulos, University of Hull, U.K

How to support design, verification and validation to achieve and maintain the SOTIF?

Guidance on measures to ensure the absence of unreasonable risk due to a hazard caused by insufficiencies of functionalities where proper situational awareness is essential to safety and where such situational awareness is derived from complex sensors and processing algorithms, including AI

- **SOTIF is crucial to achieve trustworthy AI-based systems**
  e.g., autonomous shuttles for passenger transportation near activity zones, living areas open to pedestrians, etc.

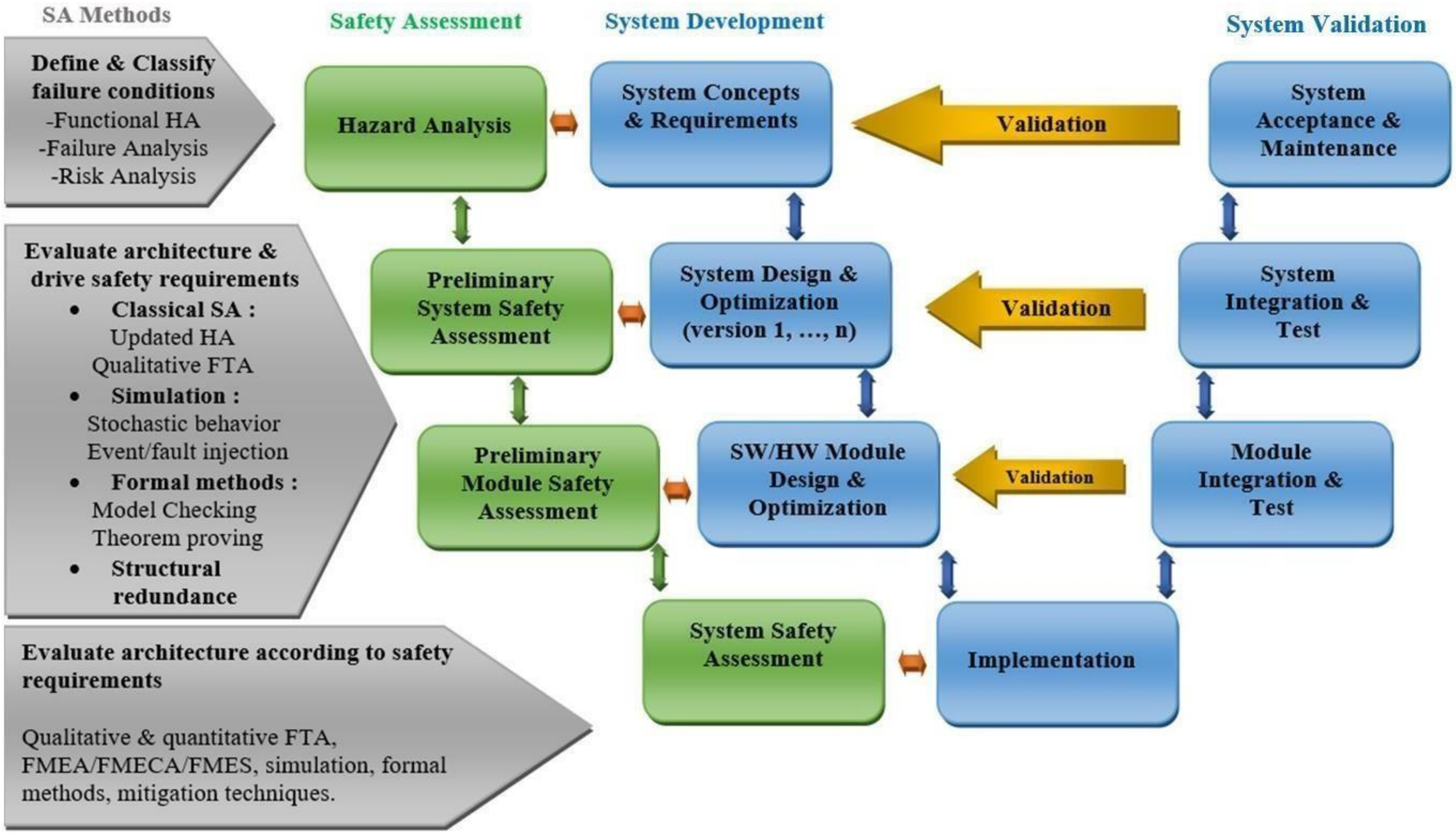
- **Challenges:**
  complex/changing operational contexts;
  data noise, ambiguous scenarios;
  degraded sensor quality and sensor failures.



Model-driven engineering as a key enabler for design and V&V of safe autonomous systems

formal process,
based on models

formal process,
based on models



SA Methods

Define & Classify failure conditions
-Functional HA
-Failure Analysis
-Risk Analysis

Evaluate architecture & drive safety requirements
- **Classical SA :**
  Updated HA
  Qualitative FTA
- **Simulation :**
  Stochastic behavior
  Event/fault injection
- **Formal methods :**
  Model Checking
  Theorem proving
- **Structural redundance**

Evaluate architecture according to safety requirements

Qualitative & quantitative FTA, FMEA/FMECA/FMES, simulation, formal methods, mitigation techniques.

Exemplification Agenda, based on Papyrus 4 Robotics framework

- **Definition of the operational domain of AI system functions**

  ODD specification

Ollier, G., Razafindrabe, D., Adedjouma, M., Gerasimou, S., Mraidha, C., 2022. « Using Operational Design Domain in Hazard Identification for Automated Systems », In proceedings of 18th *European Dependable Computing Conference.*

- **Identification of critical system functions based on safety standards**

  Papyrus for Robotics support for HARA, FMEA, FTA

Radermacher, A., Morelli, M., Hussein, M. and Nouacer, R., 2021. "Designing Drone Systems with Papyrus for Robotics". In Proceedings of the 2021 Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings.

Jihad, H.E., Adedjouma, M. and Morelli, M., 2021. "Automated Fault Tree generation in Open-PSA from UML Models". In 2021 28th Asia-Pacific Software Engineering Conference. IEEE.

- **Functional safety through anticipation of faults' impacts on the system**

  Papyrus for Robotics support for simulation-based FI

Uriagereka, G.J., et al., 2019. "Design-time safety assessment of robotic systems using fault injection simulation in a model-driven approach". In 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE.

- **Guidance on measures to ensure the safety of the intended functionality (SOTIF)**

  Combined process based on knowledge engineering and simulation for the identification and evaluation of unsafe scenarios in autonomous driving systems

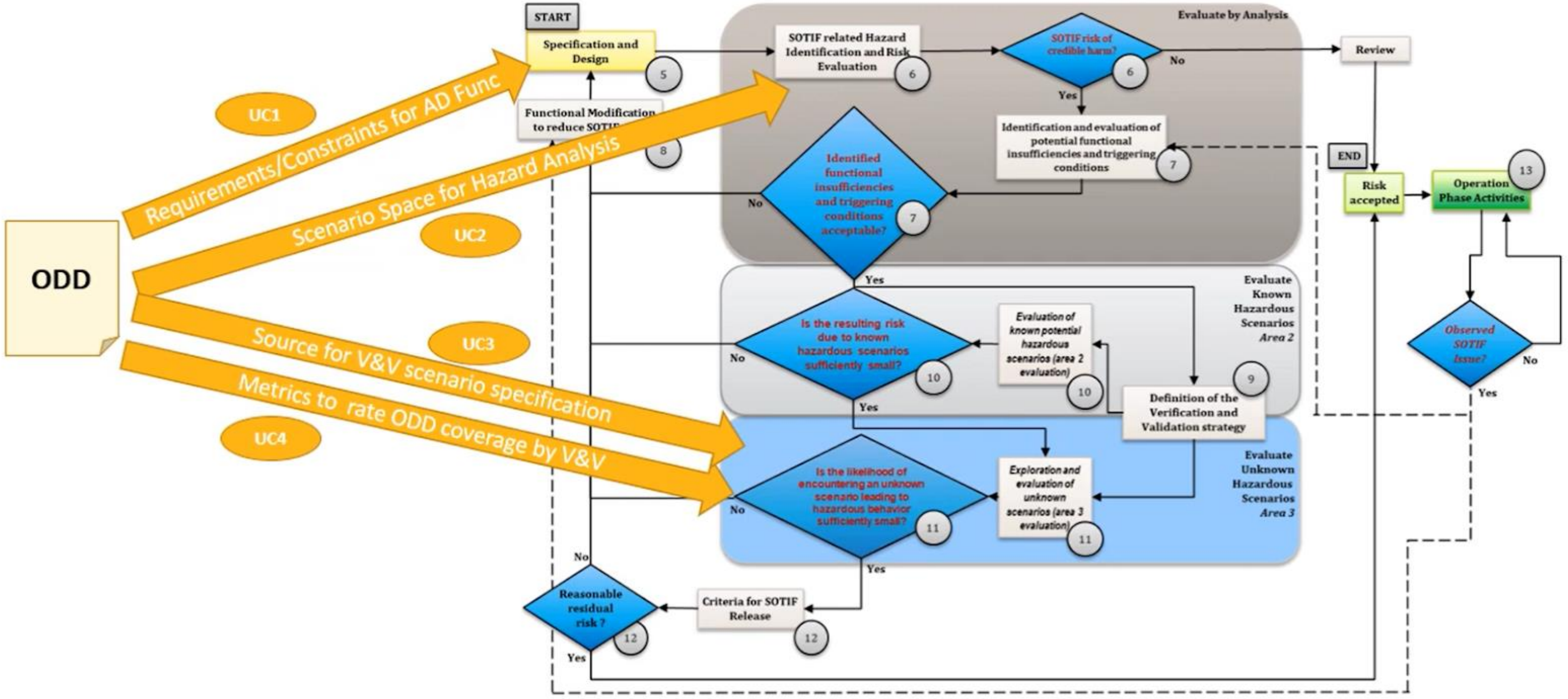
Arnez, F., Ollier, G., et al., 2022. "Skeptical Dynamic Dependability Management For Automated systems". In Euromicro DSD 2022.

Operational Design Domain (ODD) definition

Context: In practice, the number of possible scenarios which have to be managed by an automated tends to be infinite. Because the NNs learned from data, it is impossible to ensure that these data capture the infinite number of scenarios in which automated systems must operate, which makes their safety evaluation challenging.

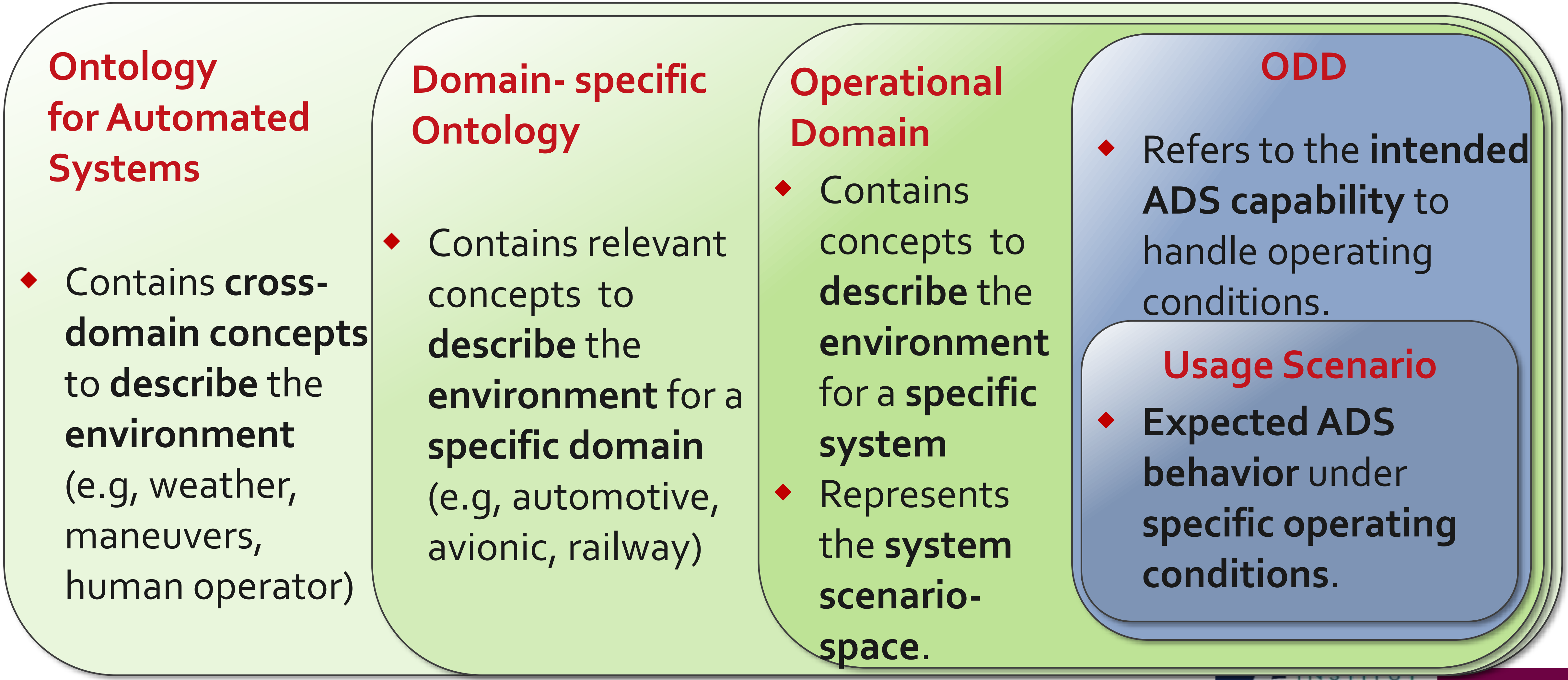
Goal: We need a mean to define the scenario-space in which the automated system must operate safely without having to enumerate the different scenarios individually. The scenario-space is specified through the operational design domain.

Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, **environmental**, **geographical**, and **time-of-day** restrictions, and/or the requisite presence or absence of certain **traffic** or **roadway characteristic**.



*Definition from SAE J3016

Operational Design Domain : From Operational Domain to ODD and scenarios



Ontology for Automated Systems

- Contains **cross-domain concepts** to **describe** the **environment** (e.g, weather, maneuvers, human operator)

Domain- specific Ontology

- Contains relevant concepts to **describe** the **environment** for a **specific domain** (e.g, automotive, avionic, railway)

Operational Domain

- Contains concepts to **describe** the **environment** for a **specific system**
- Represents the **system scenario-space**.

ODD

- Refers to the **intended ADS capability** to handle operating conditions.

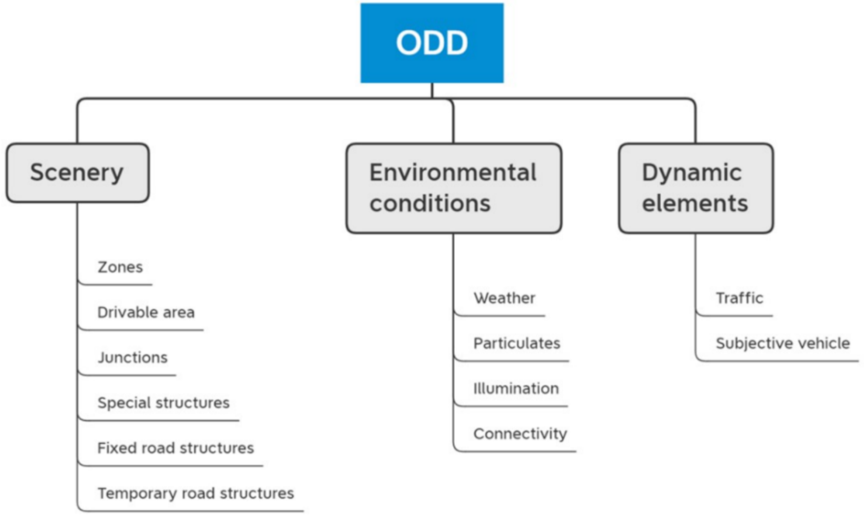
Usage Scenario

- **Expected ADS behavior** under **specific operating conditions**.

From Operational Domain to ODD and scenarios

The structuring of scenarios can be achieved following a number of approaches, e.g.:

✓ *descriptions from the outside of the ADS (e.g. 6-layer approach, ISO/DIS 34503, PAS 1883)*



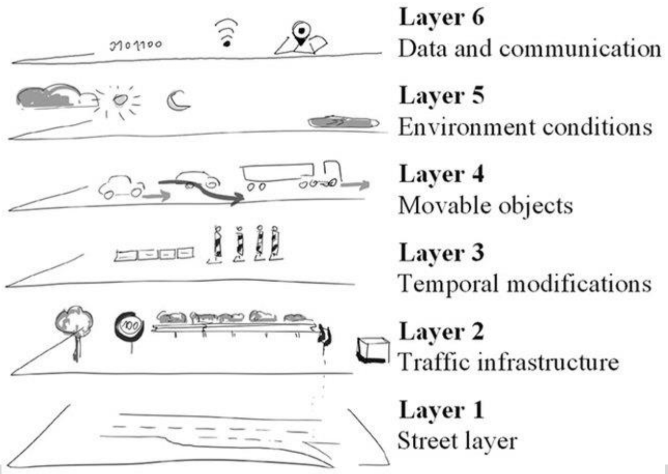
ODD

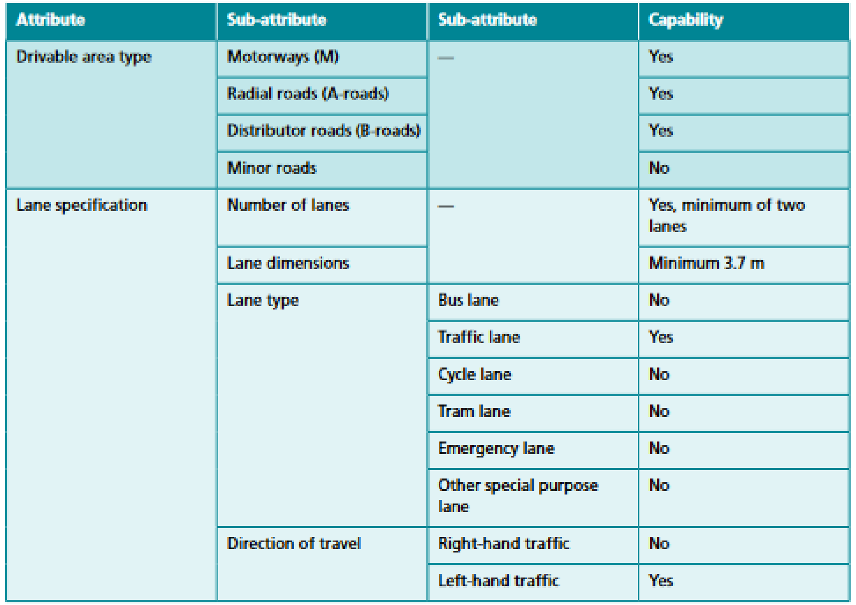
Scenery
- Zones
- Drivable area
- Junctions
- Special structures
- Fixed road structures
- Temporary road structures

Environmental conditions
- Weather
- Particulates
- Illumination
- Connectivity

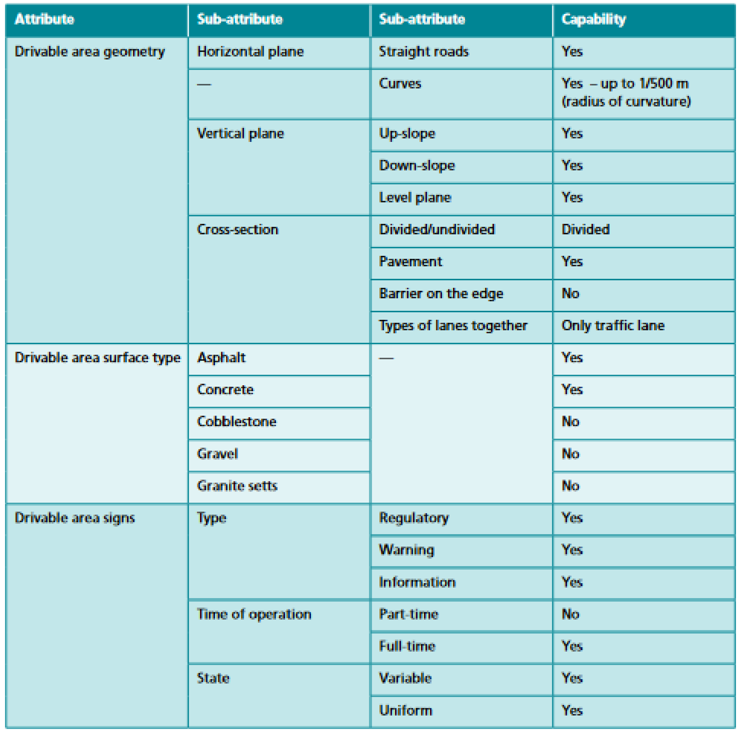
Dynamic elements
- Traffic
- Subjective vehicle

Top-level taxonomy with ODD attributes





Attribute	Sub-attribute	Sub-attribute	Capability
Drivable area type	Motorways (M)	—	Yes
	Radial roads (A-roads)		Yes
	Distributor roads (B-roads)		Yes
	Minor roads		No
Lane specification	Number of lanes	—	Yes, minimum of two lanes
	Lane dimensions		Minimum 3.7 m
	Lane type	Bus lane	No
		Traffic lane	Yes
		Cycle lane	No
		Tram lane	No
		Emergency lane	No
		Other special purpose lane	No
	Direction of travel	Right-hand traffic	No
		Left-hand traffic	Yes



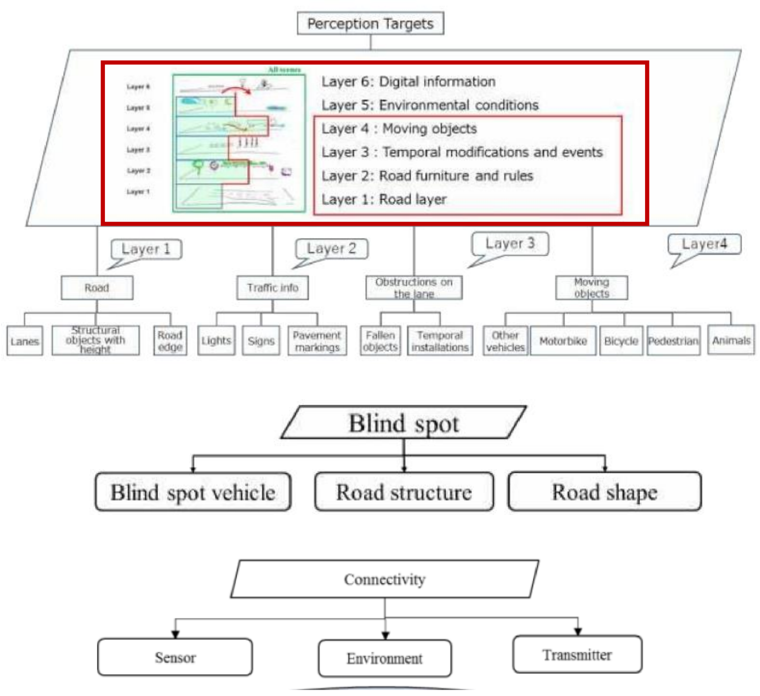
Attribute	Sub-attribute	Sub-attribute	Capability
Drivable area geometry	Horizontal plane	Straight roads	Yes
	—	Curves	Yes – up to 1/500 m (radius of curvature)
	Vertical plane	Up-slope	Yes
		Down-slope	Yes
		Level plane	Yes
	Cross-section	Divided/undivided	Divided
		Pavement	Yes
		Barrier on the edge	No
		Types of lanes together	Only traffic lane
Drivable area surface type	Asphalt	—	Yes
	Concrete		Yes
	Cobblestone		No
	Gravel		No
	Granite setts		No
Drivable area signs	Type	Regulatory	Yes
		Warning	Yes
		Information	Yes
	Time of operation	Part-time	No
		Full-time	Yes
	State	Variable	Yes
		Uniform	Yes

*Source: PAS 1883

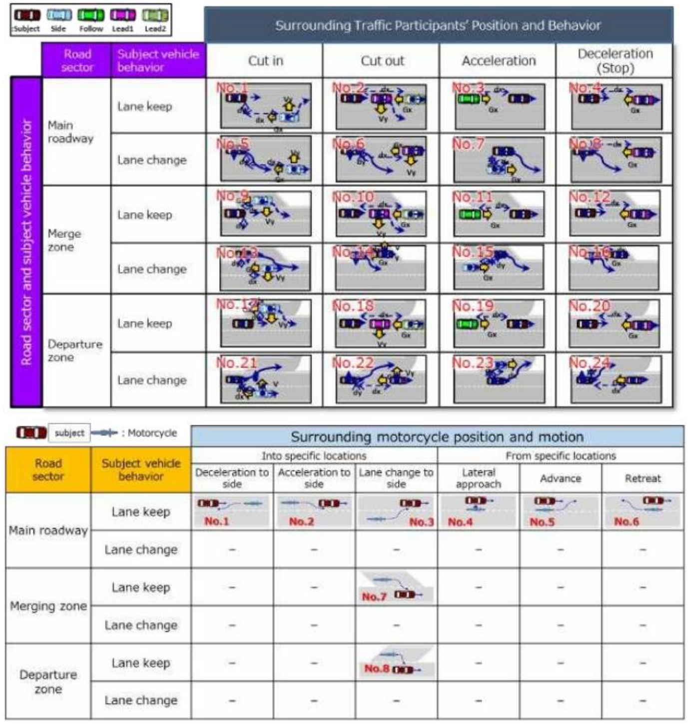
From Operational Domain to ODD and scenarios

The structuring of scenarios can be achieved following a number of approaches, e.g.:

✓ *descriptions from the inside the ADS (e.g. 3-categories approach, ISO/DIS 34502 approach)*



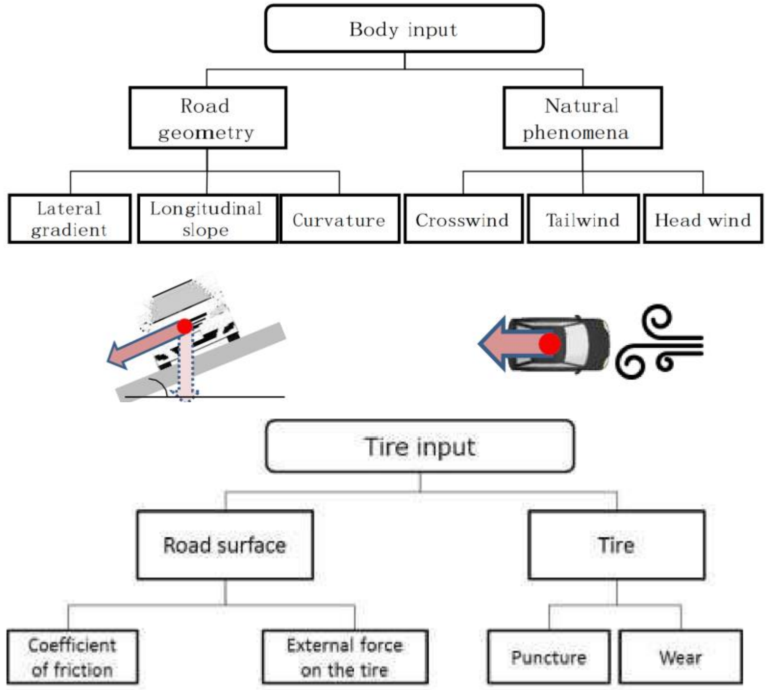
Perception related critical scenarios



Road sector	Subject vehicle behavior	Cut in	Cut out	Acceleration	Deceleration (Stop)
		Surrounding Traffic Participants' Position and Behavior			
Main roadway	Lane keep	No.1	No.2	No.3	No.4
	Lane change	No.5	No.6	No.7	No.8
Merge zone	Lane keep	No.9	No.10	No.11	No.12
	Lane change	No.13	No.14	No.15	No.16
Departure zone	Lane keep	No.17	No.18	No.19	No.20
	Lane change	No.21	No.22	No.23	No.24

Road sector	Subject vehicle behavior	Surrounding motorcycle position and motion					
		Into specific locations			From specific locations		
		Deceleration to side	Acceleration to side	Lane change to side	Lateral approach	Advance	Retreat
Main roadway	Lane keep	No.1	No.2	No.3	No.4	No.5	No.6
	Lane change	–	–	–	–	–	–
Merging zone	Lane keep	–	–	No.7	–	–	–
	Lane change	–	–	–	–	–	–
Departure zone	Lane keep	–	–	No.8	–	–	–
	Lane change	–	–	–	–	–	–

Traffic related critical scenarios



Vehicle control related critical scenarios

*Source: ISO 34502-#:####(X)-DIS draft 210908

From Operational Domain to ODD and scenarios

✓ *ODD definition and formalization using OpenODD language*

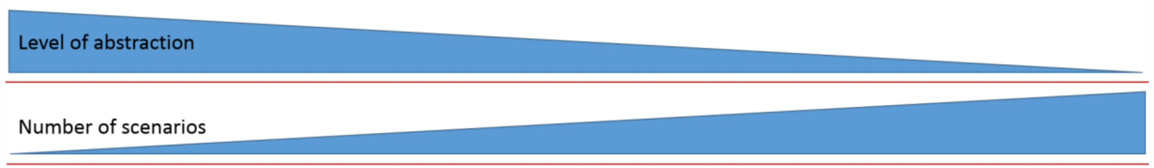
```
#Composition statements
Suitable geofenced areas is [predefined route]
Suitable regions or states is [Ottawa Canada]
Suitable zones are [regions or states, geofenced areas]
Cond_1 Conditional drivable area type are [minor roads, parking, shared space]
Cond_2 Conditional horizontal plane is [curved roads]
Unsuitable transverse plane is [divided]
Suitable types of lane together is [traffic lane]
Suitable lane dimension is [3.7,∞]
Suitable lane marking is [2,∞]
Suitable lane type is [traffic lane]
Suitable number of lanes is [2,∞]
Suitable direction of travel is [right hand travel]
Unsuitable drivable area signs is [variable]
Suitable drivable area edge is [line markers, solid barriers]
Unsuitable induced drivable area surface conditions are [flooded roadways, mirage]
Suitable drivable area surface type is [uniform]
Suitable roundabout is [normal]
Suitable normal is [non signalised]
Unsuitable intersection is [staggered, grade separated]
Suitable special structures is [pedestrian crossing]
Unsuitable temporary road structures is [construction site detours]
Unsuitable wind are [near gale, gale, strong gale, storm, violent storm, hurricane-
force]
Unsuitable rainfall is [violent rain, cloudburst]
Suitable temperature is [-30,40 C]
Suitable particulates is [non precipitating water droplets]
Suitable vehicle to infrastructure is [cellular]
Suitable positioning is [global positioning]
Suitable subject vehicle speed is [0,15 km/h]

#Conditional statements
Cond_1 Suitable speed of subject vehicle for [minor roads] is [0,15 km/h]
Cond_1 Suitable speed of subject vehicle for [parking, shared space] is [0,10 km/h]
Cond_2 Unsuitable radius of curved road is [0,5 m]
```

*Source: ISO 34503-#:####(X)-WD 34503 – r11.0

From Operational Domain to ODD and scenarios

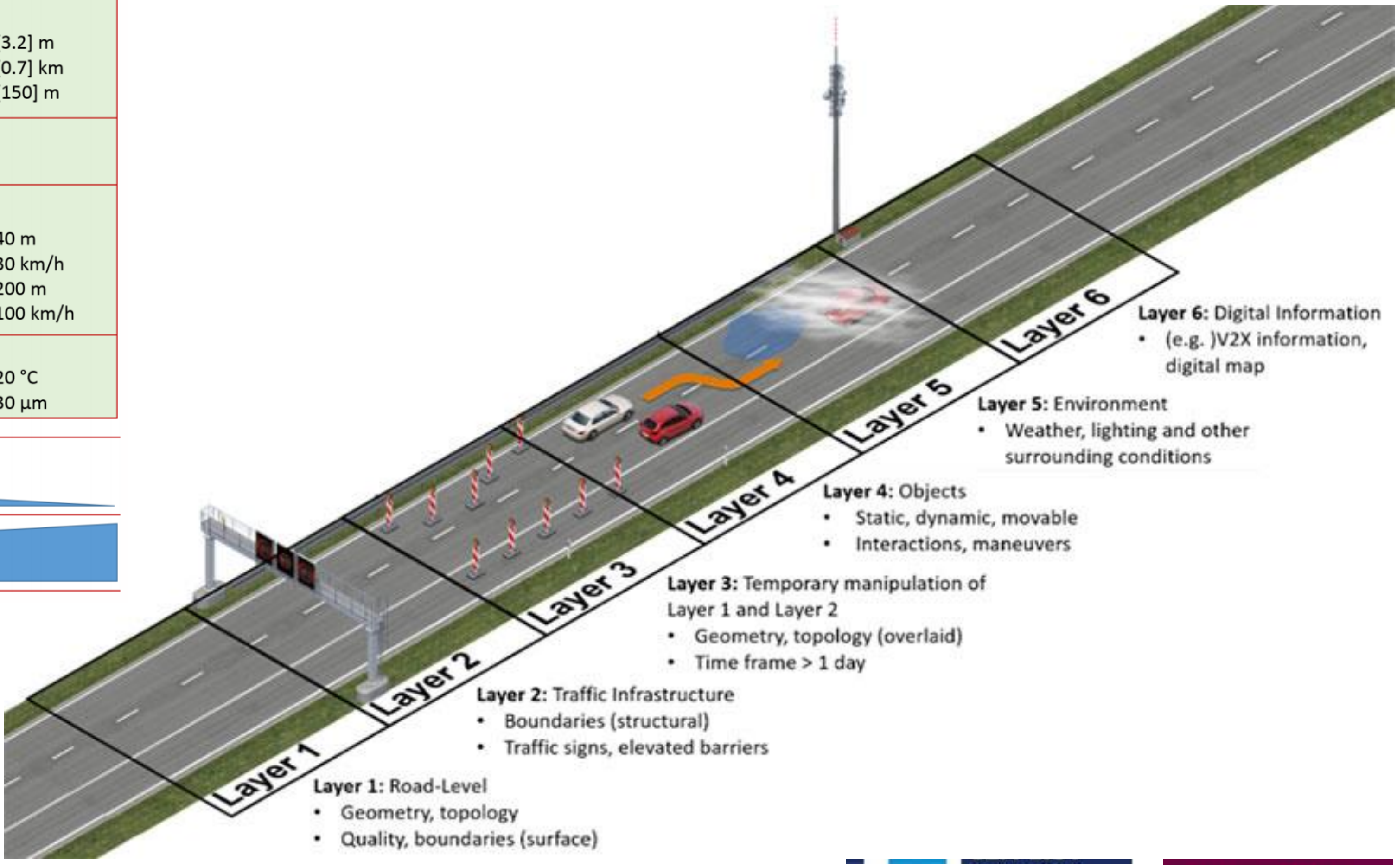
Scenario description can be done at functional, logical, concrete levels

Functional scenarios	**Logical scenarios**	**Concrete scenarios**
Base road network: three-lane motorway in a curve, 100 km/h speed limit indicated by traffic signs	Base road network: Lane width [2.3..3.5] m Curve radius [0.6..0.9] km Position traffic sign [0..200] m	Base road network: Lane width [3.2] m Curve radius [0.7] km Position traffic sign [150] m
Stationary objects: -	Stationary objects: -	Stationary objects: -
Moveable objects: Ego vehicle, traffic jam; Interaction: Ego in maneuver „approaching“ on the middle lane, traffic jam moves slowly	Moveable objects: End of traffic jam [10..200] m Traffic jam speed [0..30] km/h Ego distance [50..300] m Ego speed [80..130] km/h	Moveable objects: End of traffic jam 40 m Traffic jam speed 30 km/h Ego distance 200 m Ego speed 100 km/h
Environment: Summer, rain	Environment: Temperature [10..40] °C Droplet size [20..100] µm	Environment: Temperature 20 °C Droplet size 30 µm



- ✓ *Do we need to include occupants and vehicle status?*

*Source: https://www.pegasusprojekt.de/de/about-PEGASUS



Layer 6: Digital Information
- (e.g.)V2X information, digital map

Layer 5: Environment
- Weather, lighting and other surrounding conditions

Layer 4: Objects
- Static, dynamic, movable
- Interactions, maneuvers

Layer 3: Temporary manipulation of Layer 1 and Layer 2
- Geometry, topology (overlaid)
- Time frame > 1 day

Layer 2: Traffic Infrastructure
- Boundaries (structural)
- Traffic signs, elevated barriers

Layer 1: Road-Level
- Geometry, topology
- Quality, boundaries (surface)

Exemplification Agenda

- **Definition of the operational domain of AI system functions**

ODD specification

Ollier, G., Razafindrabe, D., Adedjouma, M., Gerasimou, S., Mraidha, C., 2022. « Using Operational Design Domain in Hazard Identification for Automated Systems », In proceedings of 18th *European Dependable Computing Conference.*

- **Identification of critical system functions based on safety standards**

Papyrus for Robotics support for HARA, FMEA, FTA

Radermacher, A., Morelli, M., Hussein, M. and Nouacer, R., 2021. "Designing Drone Systems with Papyrus for Robotics". In Proceedings of the 2021 Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings.

Jihad, H.E., Adedjouma, M. and Morelli, M., 2021. "Automated Fault Tree generation in Open-PSA from UML Models". In 2021 28th Asia-Pacific Software Engineering Conference. IEEE.

- **Functional safety through anticipation of faults' impacts on the system**

Papyrus for Robotics support for simulation-based FI

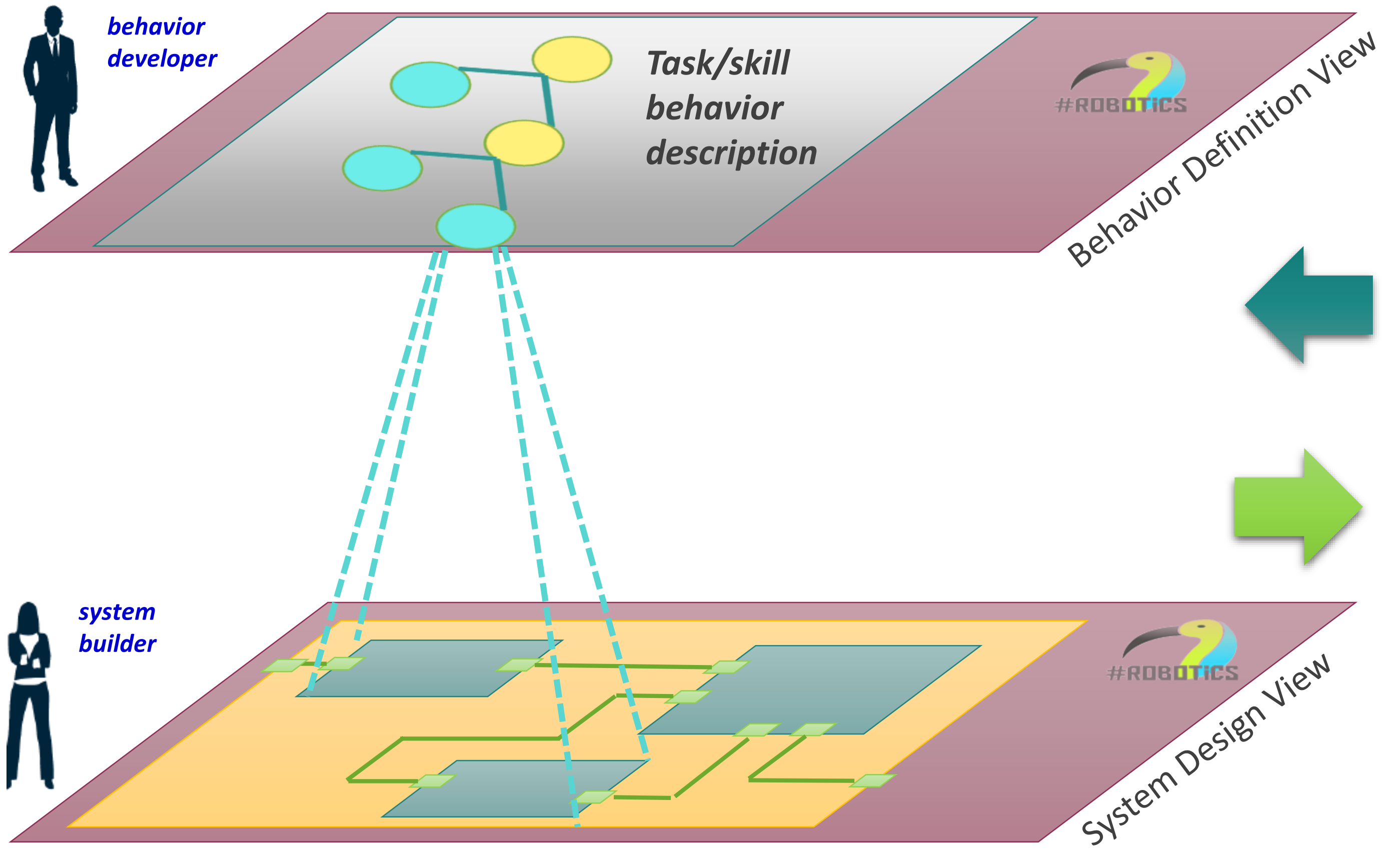
Uriagereka, G.J., et al., 2019. "Design-time safety assessment of robotic systems using fault injection simulation in a model-driven approach". In 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE.

- **Guidance on measures to ensure the safety of the intended functionality (SOTIF)**

Combined process based on knowledge engineering and simulation for the identification and evaluation of unsafe scenarios in autonomous driving systems

Arnez, F., Ollier, G., et al., 2022. "Skeptical Dynamic Dependability Management For Automated systems". In Euromicro DSD 2022.

Standard-based hazard analysis and risk assessment



HAZARD vs **RISK**

A HAZARD is something that has the potential to harm you

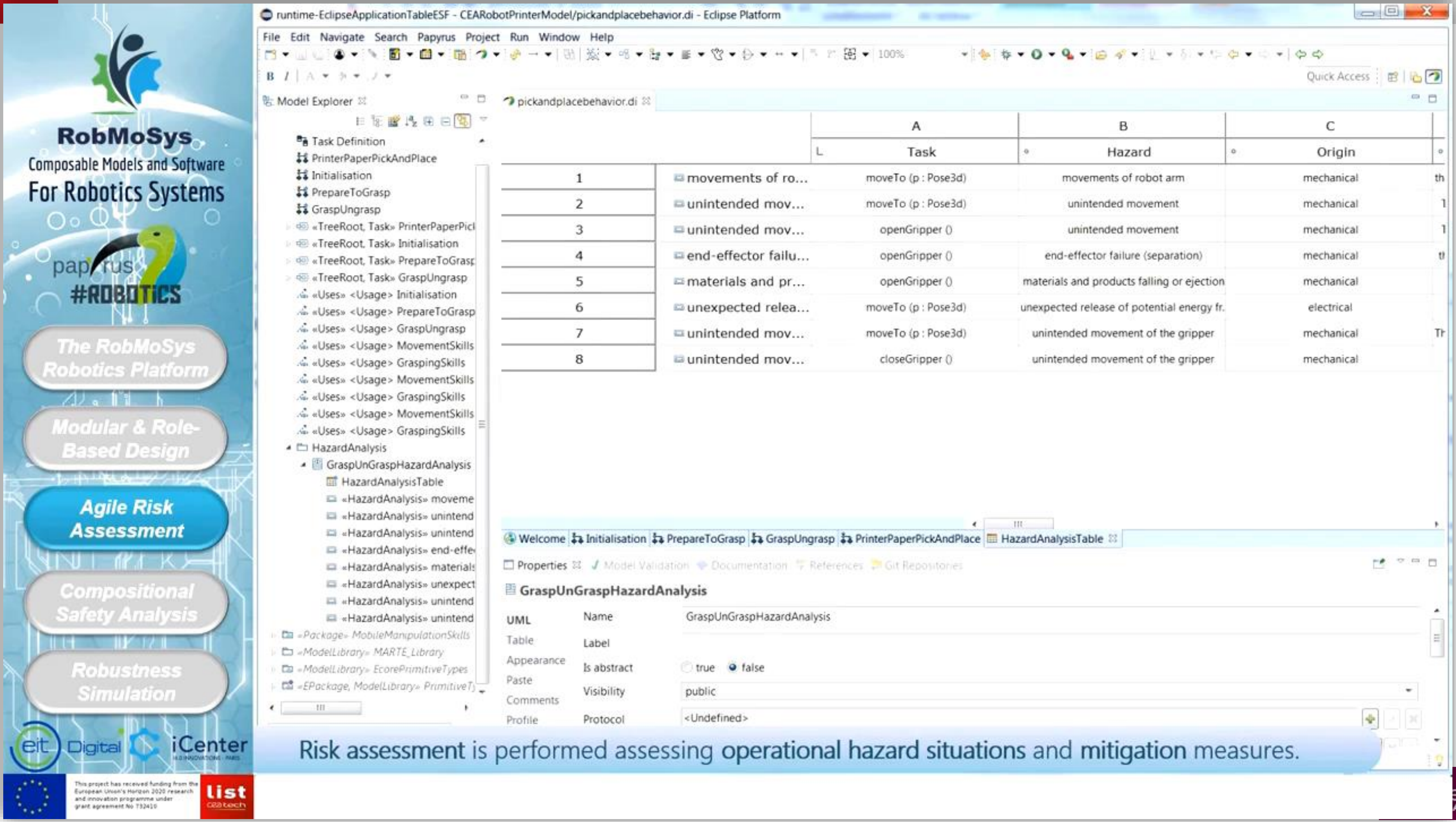
RISK is the likelihood of a hazard causing harm

risk = hazard x exposure

Likelihood	Criticality 4	3	2R	2	1R	1
High	M	M	H	H	H	H
Medium	L	M	M	M	H	H
Low	L	L	M	M	M	M

Hazard Analysis and Risk Assessment view

- **HARA is performed following ISO 10218-2:2011.**
  list all the relevant hazards at system and behavior level and compute their risk index.
  The risk analysis table structure is extracted from **ISO/TR 14121-2:2007**.



		A	B	C	
		Task	Hazard	Origin	
1	movements of ro…	moveTo (p : Pose3d)	movements of robot arm	mechanical	th
2	unintended mov…	moveTo (p : Pose3d)	unintended movement	mechanical	1
3	unintended mov…	openGripper ()	unintended movement	mechanical	1
4	end-effector failu…	openGripper ()	end-effector failure (separation)	mechanical	tl
5	materials and pr…	openGripper ()	materials and products falling or ejection	mechanical	
6	unexpected relea…	moveTo (p : Pose3d)	unexpected release of potential energy fr.	electrical	
7	unintended mov…	moveTo (p : Pose3d)	unintended movement of the gripper	mechanical	Th
8	unintended mov…	closeGripper ()	unintended movement of the gripper	mechanical	

GraspUnGraspHazardAnalysis

Name: GraspUnGraspHazardAnalysis

Label:

Is abstract: ○ true ● false

Visibility: public

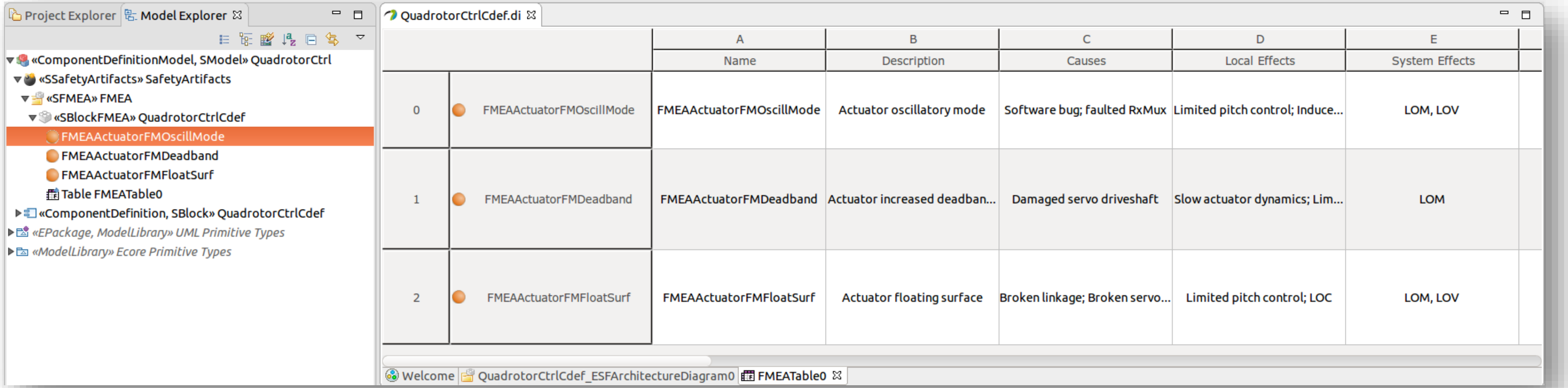
Protocol: <Undefined>

Risk assessment is performed assessing operational hazard situations and mitigation measures.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732410

Complementing HARA with FMEA

- **HARA is a preliminary analysis step, needs to be completed with FMEA**
  from hazardous situations to failure modes, causes and effects → FM criticality is automatically computed

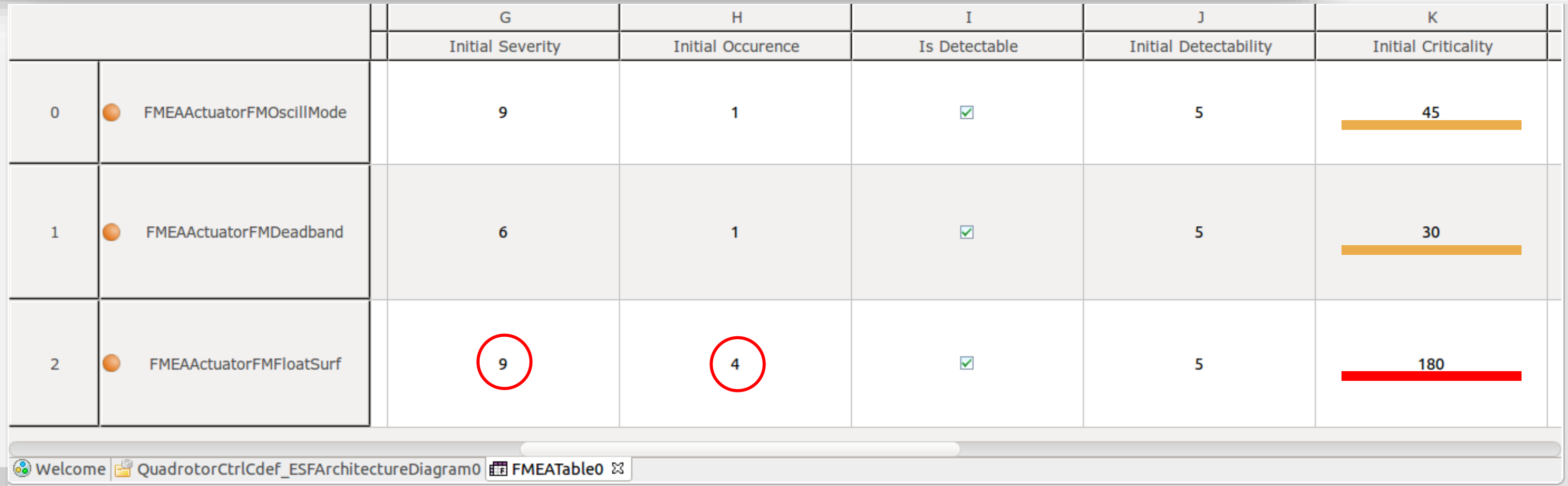


- «ComponentDefinitionModel, SModel» QuadrotorCtrl
  - «SSafetyArtifacts» SafetyArtifacts
    - «SFMEA» FMEA
      - «SBlockFMEA» QuadrotorCtrlCdef
        - FMEAActuatorFMOscillMode
        - FMEAActuatorFMDeadband
        - FMEAActuatorFMFloatSurf
        - Table FMEATable0
  - «ComponentDefinition, SBlock» QuadrotorCtrlCdef
  - *«EPackage, ModelLibrary» UML Primitive Types*
  - *«ModelLibrary» Ecore Primitive Types*

		A	B	C	D	E
		Name	Description	Causes	Local Effects	System Effects
0	● FMEAActuatorFMOscillMode	FMEAActuatorFMOscillMode	Actuator oscillatory mode	Software bug; faulted RxMux	Limited pitch control; Induce...	LOM, LOV
1	● FMEAActuatorFMDeadband	FMEAActuatorFMDeadband	Actuator increased deadban...	Damaged servo driveshaft	Slow actuator dynamics; Lim...	LOM
2	● FMEAActuatorFMFloatSurf	FMEAActuatorFMFloatSurf	Actuator floating surface	Broken linkage; Broken servo...	Limited pitch control; LOC	LOM, LOV

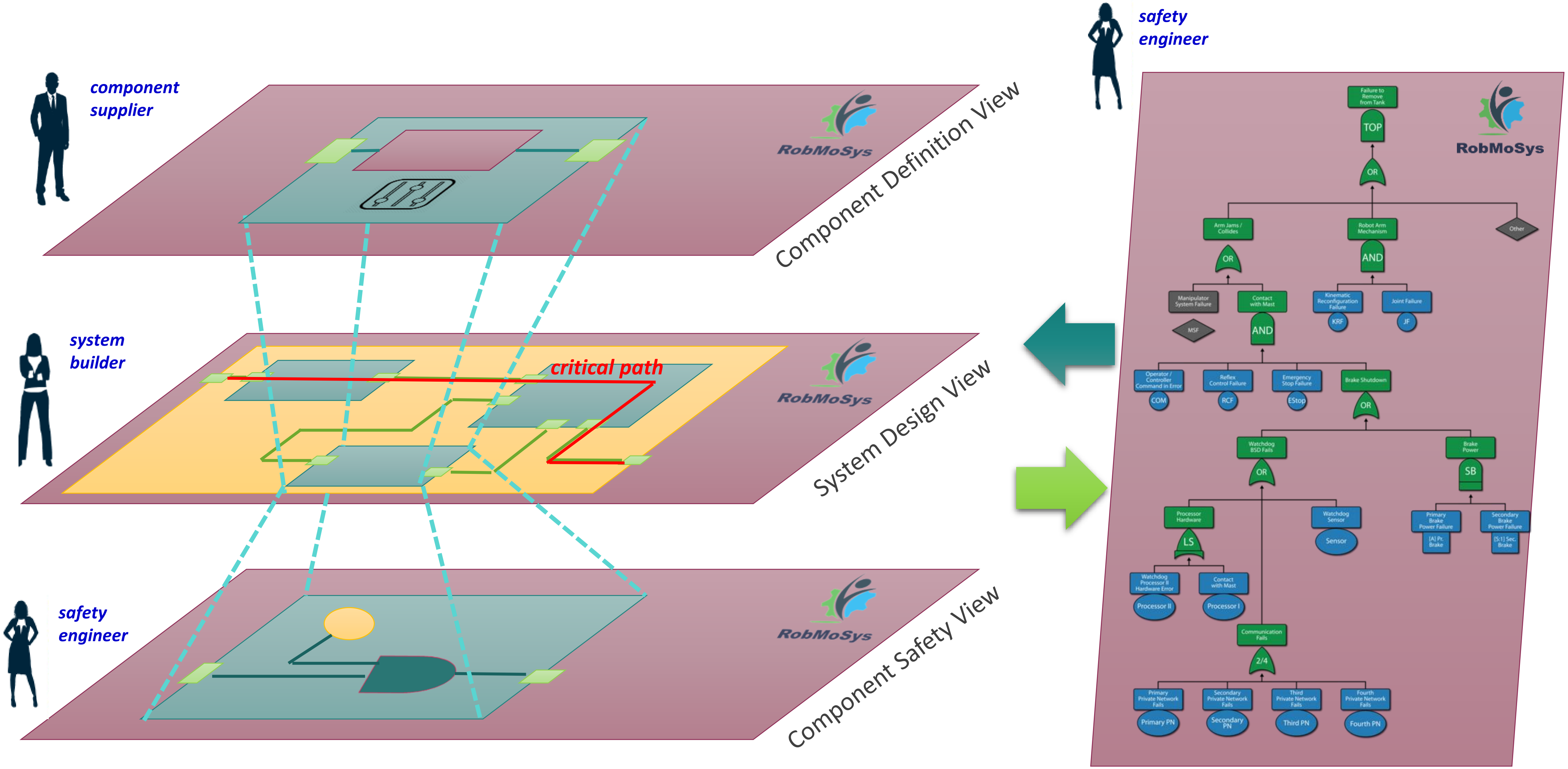
...

...



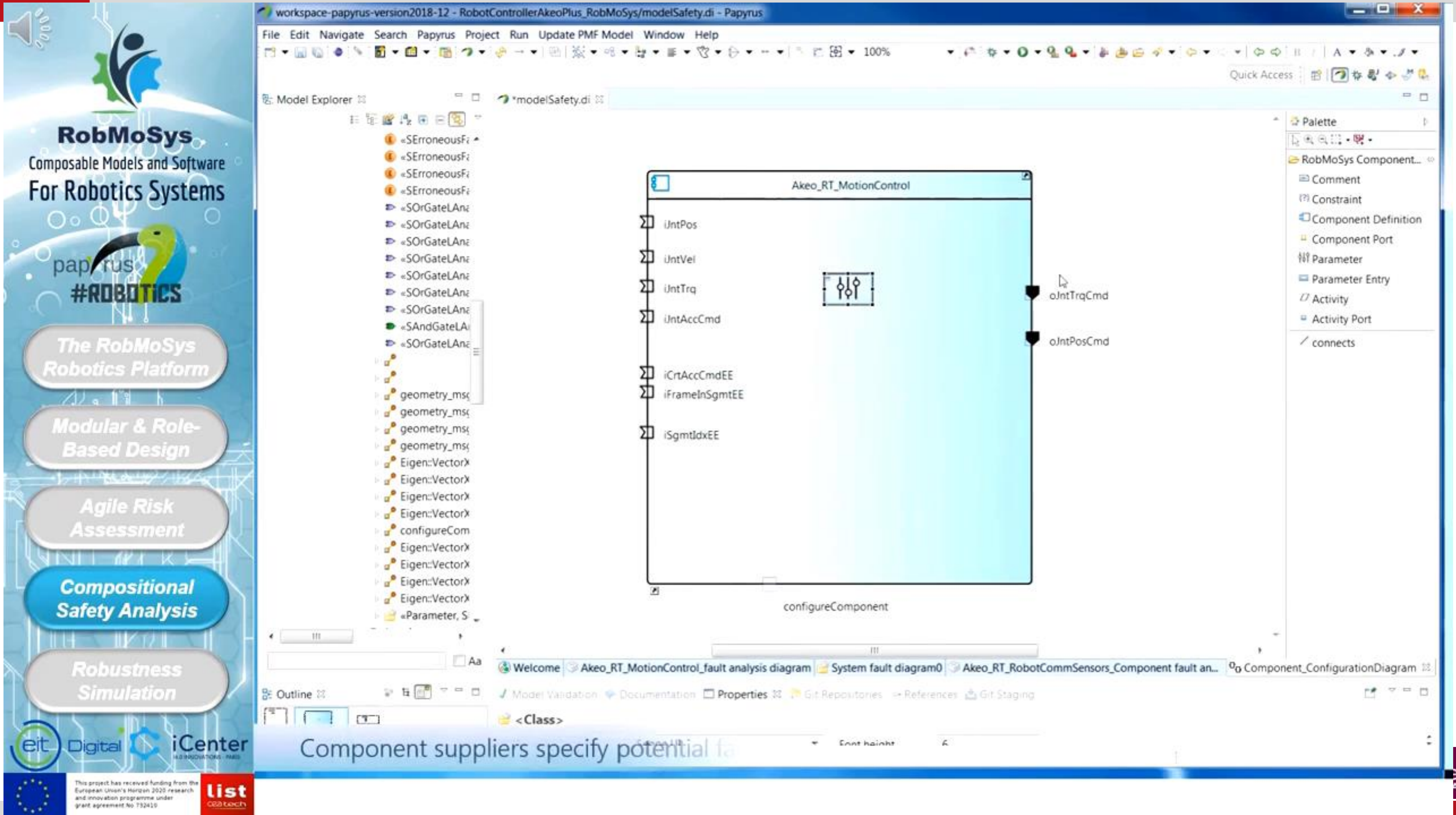
		G	H	I	J	K
		Initial Severity	Initial Occurence	Is Detectable	Initial Detectability	Initial Criticality
0	● FMEAActuatorFMOscillMode	9	1	☑	5	45
1	● FMEAActuatorFMDeadband	6	1	☑	5	30
2	● FMEAActuatorFMFloatSurf	9	4	☑	5	180

Complementing with compositional safety analysis



Fault Tree Analysis (FTA) View

- From failure modes (causes/effects) to feared events
- Combination/Propagation of failures on the architecture, and cut-sets



Component suppliers specify potential fa

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732410

Exemplification Agenda

- **Definition of the operational domain of AI system functions**

  ODD specification

Ollier, G., Razafindrabe, D., Adedjouma, M., Gerasimou, S., Mraidha, C., 2022. « Using Operational Design Domain in Hazard Identification for Automated Systems », In proceedings of 18th *European Dependable Computing Conference.*

- **Identification of critical system functions based on safety standards**

  Papyrus for Robotics support for HARA, FMEA, FTA

Radermacher, A., Morelli, M., Hussein, M. and Nouacer, R., 2021. "Designing Drone Systems with Papyrus for Robotics". In Proceedings of the 2021 Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings.

Jihad, H.E., Adedjouma, M. and Morelli, M., 2021. "Automated Fault Tree generation in Open-PSA from UML Models". In 2021 28th Asia-Pacific Software Engineering Conference. IEEE.

- **Functional safety through anticipation of faults' impacts on the system**

  Papyrus for Robotics support for simulation-based FI

Uriagereka, G.J., et al., 2019. "Design-time safety assessment of robotic systems using fault injection simulation in a model-driven approach". In 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE.

- **Guidance on measures to ensure the safety of the intended functionality (SOTIF)**

  Combined process based on knowledge engineering and simulation for the identification and evaluation of unsafe scenarios in autonomous driving systems

Arnez, F., Ollier, G., et al., 2022. "Skeptical Dynamic Dependability Management For Automated systems". In Euromicro DSD 2022.

Design-time safety assessment using fault injection simulation

Faults

data arrives too late (communication delay)

data are corrupted, etc.

How does the system react to faults?

faults can even jeopardize the system stability

Process

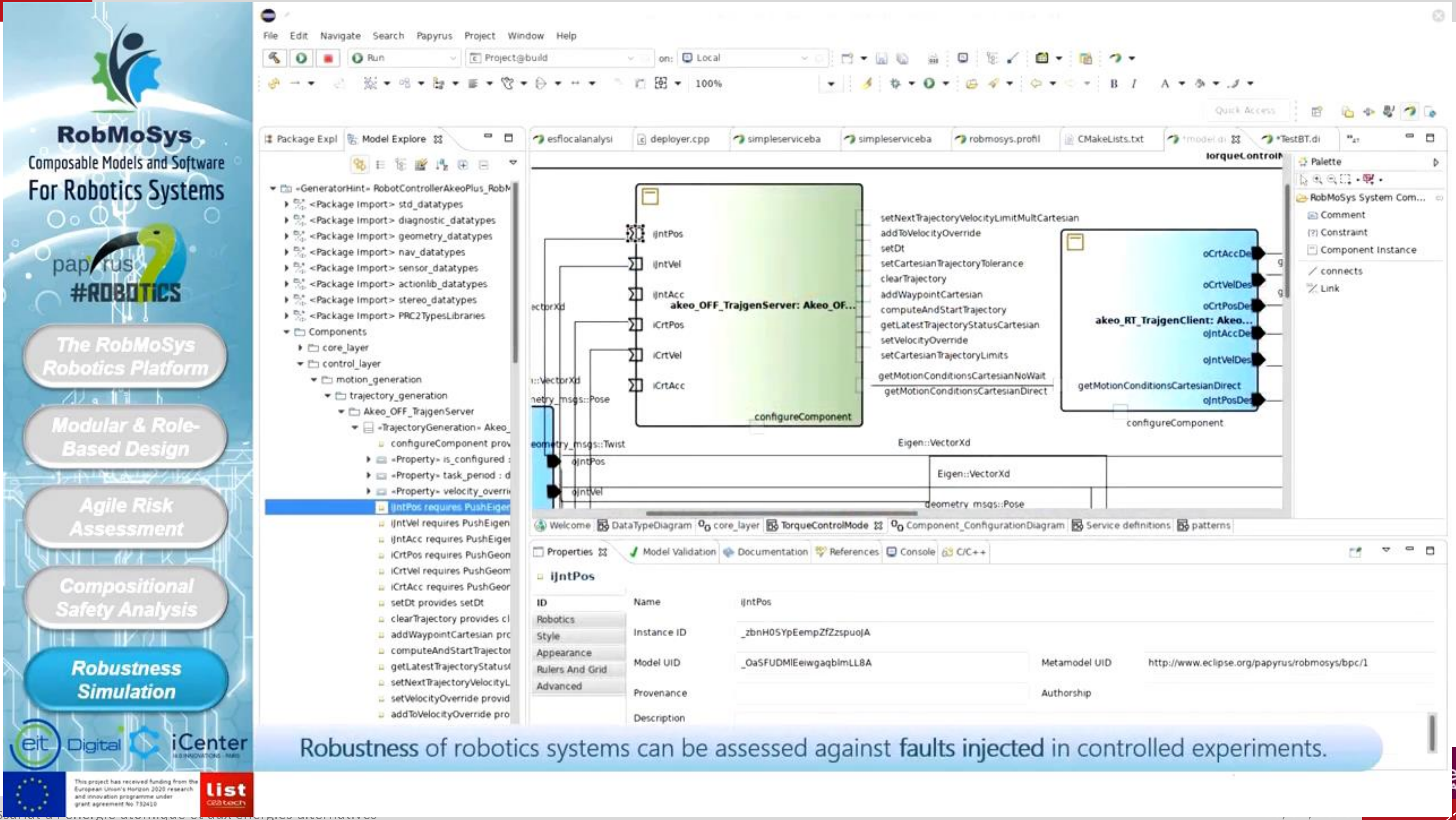
annotate system model with a fault specification

generate “Saboteur” component from specification, inject it into architecture

simulate, simulate, simulate

observe run-time behavior and refine the design:

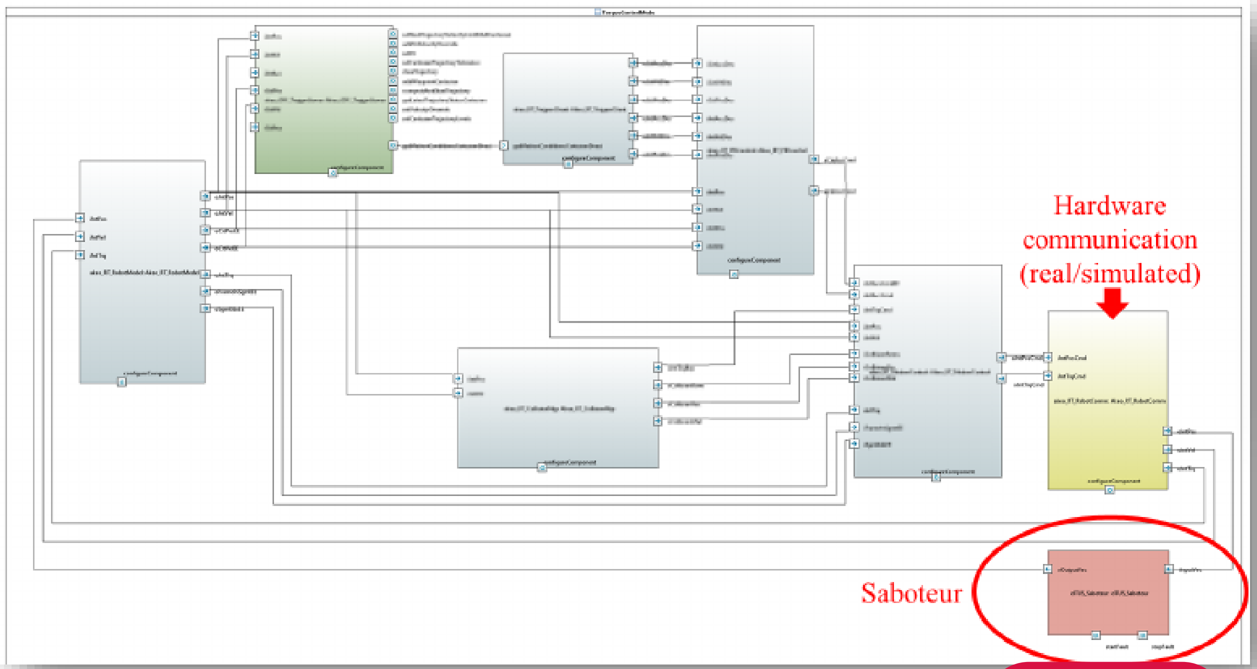
- under which conditions the system stability is jeopardized?
- which are appropriate strategies to add to the architecture design and ensure the mitigation of fault effects?
- which is the lowest response time that a monitor must have to trigger mitigation measures?
- …

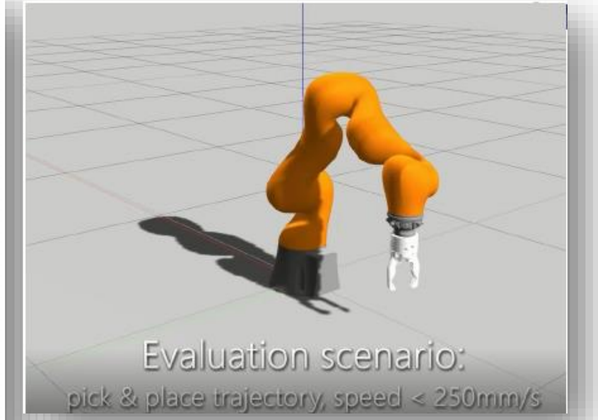


Name: iJntPos
Instance ID: _zbnH05YpEempZfZzspuoJA
Model UID: _OaSFUDMlEeiwgaqblmLL8A
Metamodel UID: http://www.eclipse.org/papyrus/robmosys/bpc/1

Robustness of robotics systems can be assessed against **faults injected** in controlled experiments.

Recap on design-time safety assessment using fault injection simulation




Evaluation scenario:
pick & place trajectory, speed < 250mm/s

thanks

@ tecnalia Inspiring Business

Function	Component Fault Target	Fault Model	Potential Effect	System Maximun Response Time (ms)
Trajectory Controller	*Sensor*	*Stuck-at 0*	*The velocity increase exceeding the maximum robot velocity*	27
Trajectory Controller	*Sensor*	*Stuck-at Last Value*	*The velocity increase exceeding the maximum robot velocity*	222

MDE-based simulated fault injection enables :

- quantitative assessment of safety properties of interest
- refinement cycles of design until reaching the required level of safety
- exploration of mitigation strategies to potential hazards in early development phases

Exemplification Agenda

- **Definition of the operational domain of AI system functions**

  ODD specification

Ollier, G., Razafindrabe, D., Adedjouma, M., Gerasimou, S., Mraidha, C., 2022. « Using Operational Design Domain in Hazard Identification for Automated Systems », In proceedings of 18th *European Dependable Computing Conference.*

- **Identification of critical system functions based on safety standards**

  Papyrus for Robotics support for HARA, FMEA, FTA

Radermacher, A., Morelli, M., Hussein, M. and Nouacer, R., 2021. "Designing Drone Systems with Papyrus for Robotics". In Proceedings of the 2021 Drone Systems Engineering and Rapid Simulation and Performance Evaluation: Methods and Tools Proceedings.

Jihad, H.E., Adedjouma, M. and Morelli, M., 2021. "Automated Fault Tree generation in Open-PSA from UML Models". In 2021 28th Asia-Pacific Software Engineering Conference. IEEE.

- **Functional safety through anticipation of faults' impacts on the system**

  Papyrus for Robotics support for simulation-based FI

Uriagereka, G.J., et al., 2019. "Design-time safety assessment of robotic systems using fault injection simulation in a model-driven approach". In 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE.

- **Guidance on measures to ensure the safety of the intended functionality (SOTIF)**

  Combined process based on knowledge engineering and simulation for the identification and evaluation of unsafe scenarios in autonomous driving systems

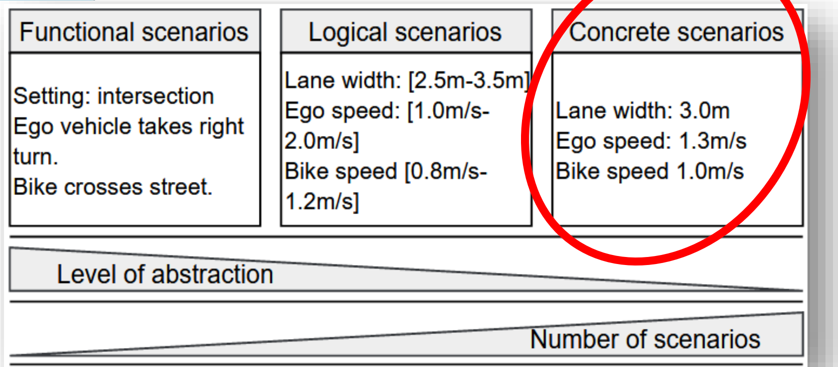
Arnez, F., Ollier, G., et al., 2022. "Skeptical Dynamic Dependability Management For Automated systems". In Euromicro DSD 2022.

Limitations of current solutions when dealing with SOTIF

Current solutions ***rarely enable the assessment of the effects of functional insufficiencies of learning-enabled components***

- Validation of the driving logics by simulation
  Solving ODE/physics simulation but
  with limited rendering realism and simplistic sensor models
  → unable to run realistic AI-based perception pipelines

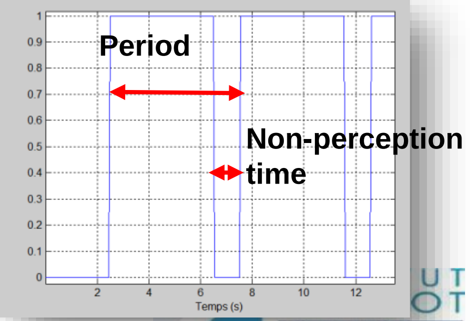
- Parameterizable elements in the operational scenarios
  Vehicle, pedestrian, road, sign, traffic light status/attributes parameterizable
  at the level of concrete scenarios
  → large number of low-level scenario descriptions needed
  → no support for intelligent generation of scenarios from higher-level specs
  → simulator-specific (migration to other technologies may require big effort)



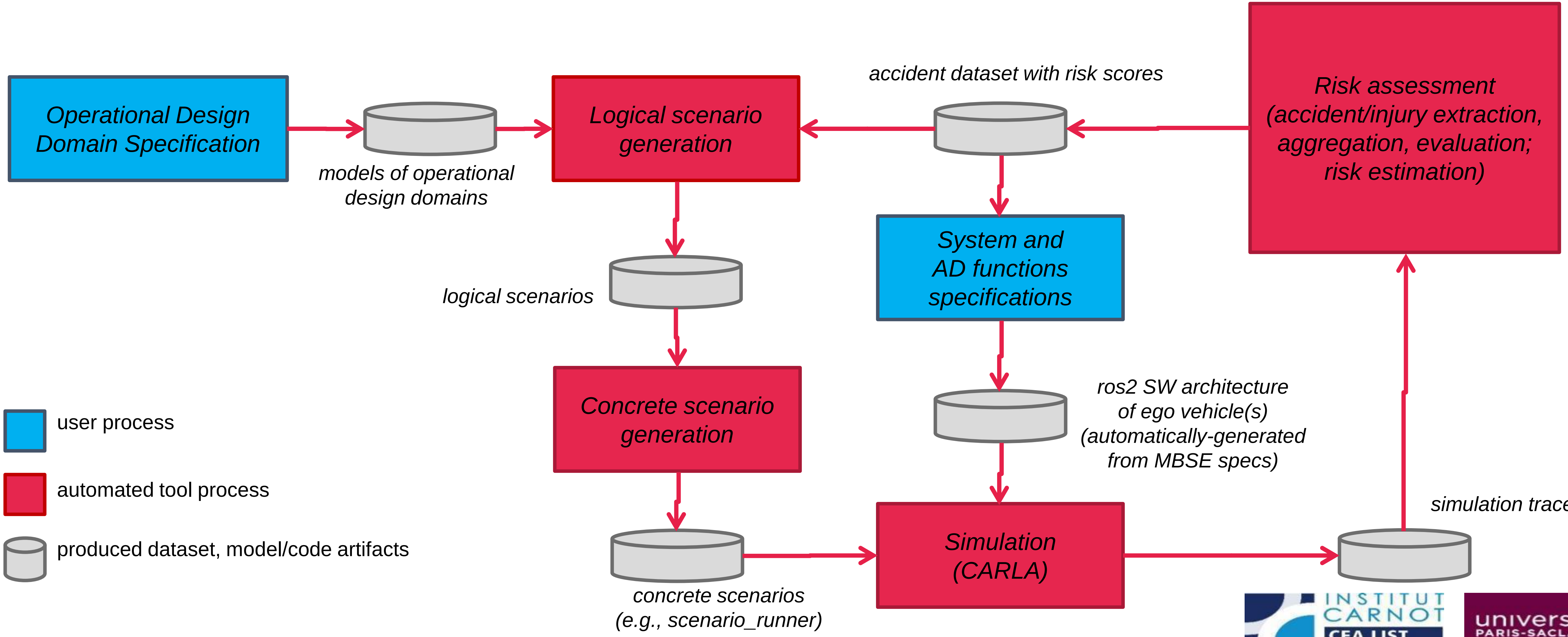
Functional scenarios	Logical scenarios	Concrete scenarios
Setting: intersection Ego vehicle takes right turn. Bike crosses street.	Lane width: [2.5m-3.5m] Ego speed: [1.0m/s-2.0m/s] Bike speed [0.8m/s-1.2m/s]	Lane width: 3.0m Ego speed: 1.3m/s Bike speed 1.0m/s

Menzel et al. [2018]

- Parameterizable failure models
  Failures of perception and localization systems can be simulated
  only in a simplistic way (non-perception time over an acquisition perdiod)
  → complicates the design of mitigation policies in ambiguous situations (e.g., wrong information perceived) or of policies aware of perception uncertainty



Combined process based on knowledge engineering and simulation for the identification and evaluation of unsafe scenarios in autonomous driving systems



Operational Design Domain Specification

models of operational design domains

Logical scenario generation

accident dataset with risk scores

Risk assessment (accident/injury extraction, aggregation, evaluation; risk estimation)

logical scenarios

System and AD functions specifications

Concrete scenario generation

ros2 SW architecture of ego vehicle(s) (automatically-generated from MBSE specs)

simulation traces

Simulation (CARLA)

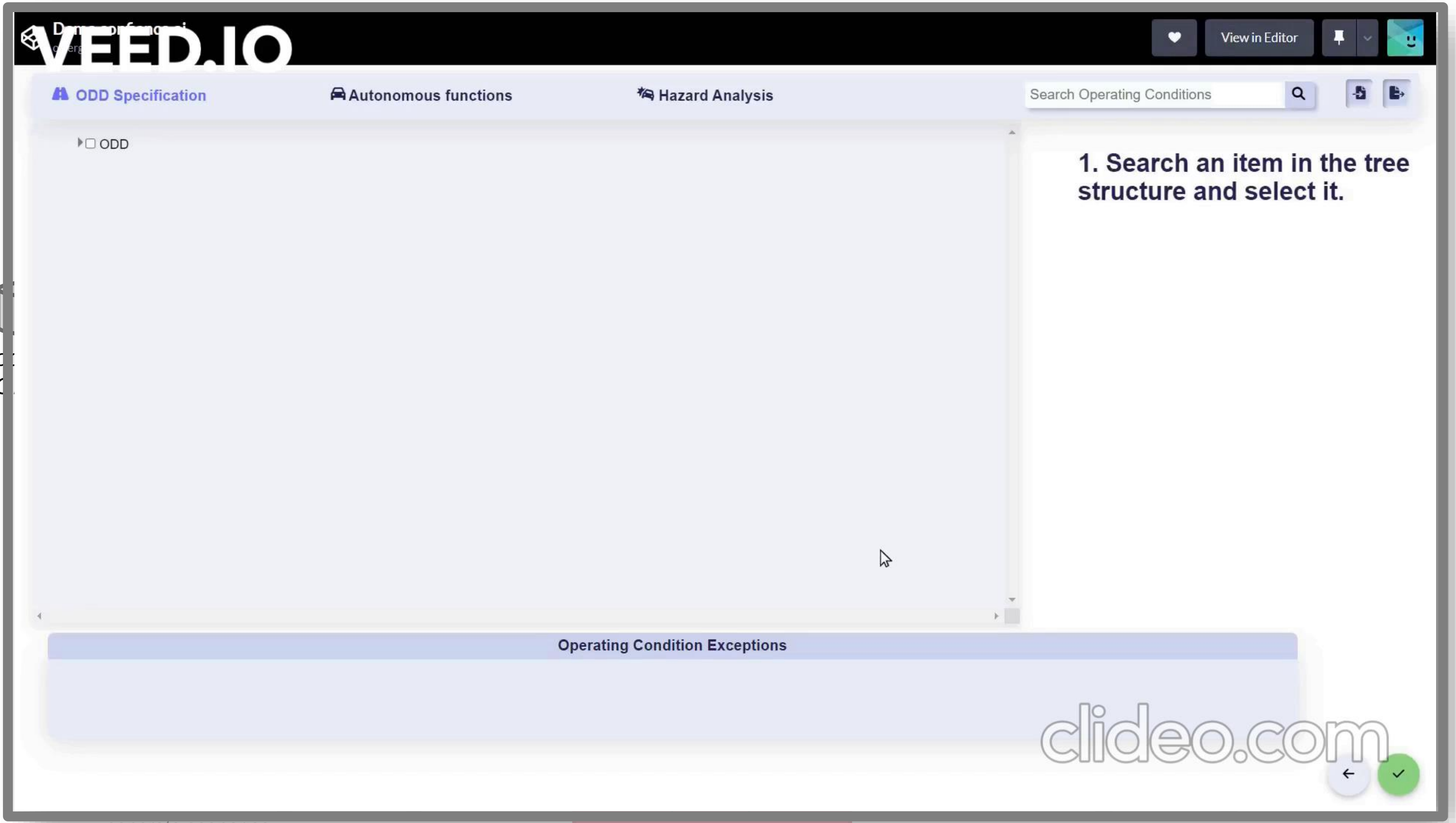
concrete scenarios (e.g., scenario_runner)

🟦 user process

🟥 automated tool process

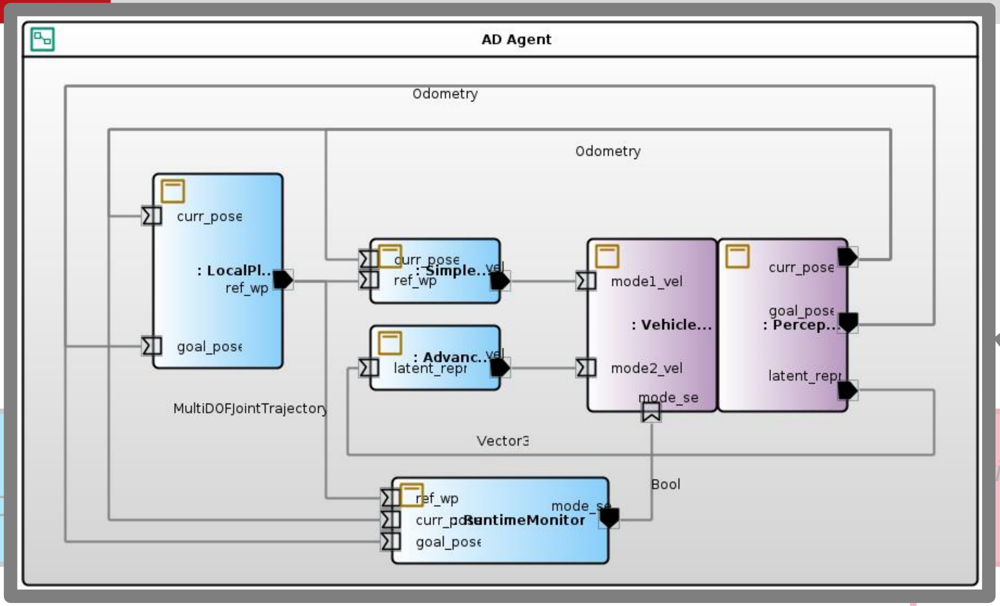
⛁ produced dataset, model/code artifacts

Operational Design Domain Specification

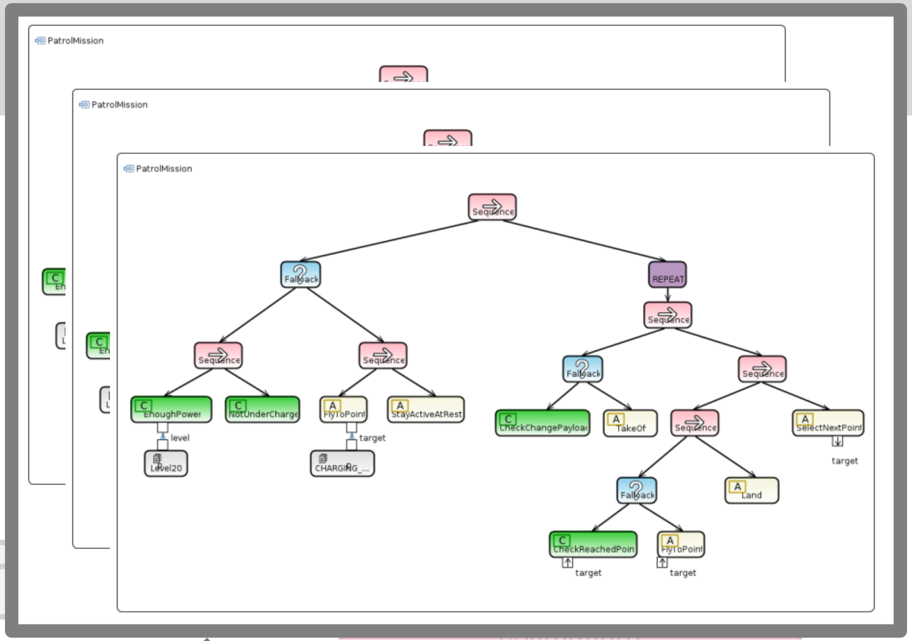


1. Search an item in the tree structure and select it.

concrete scenarios
(e.g., scenario_runner)



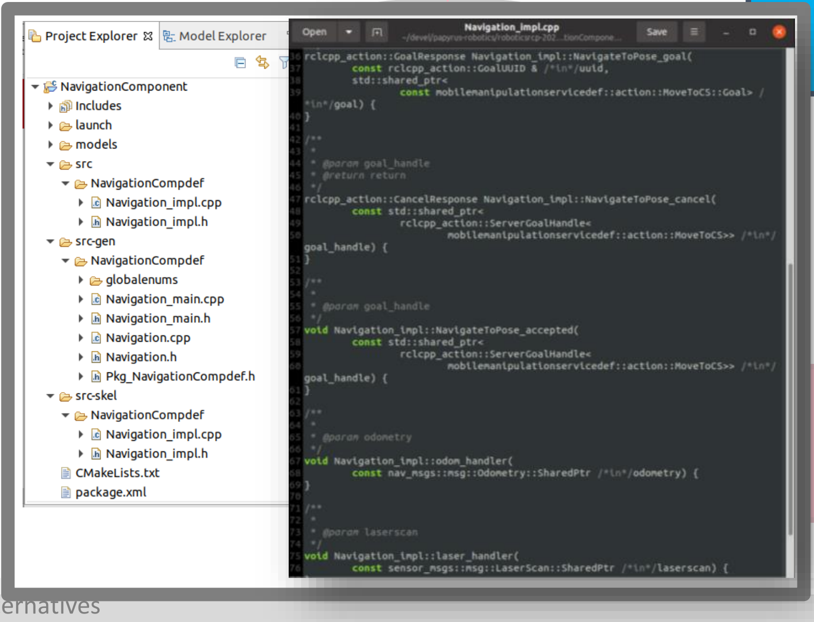
Autonomous Driving Agent's SW Architecture



Autonomous Driving Agent's Behavior Logics

accident dataset

System and AD functions specifications

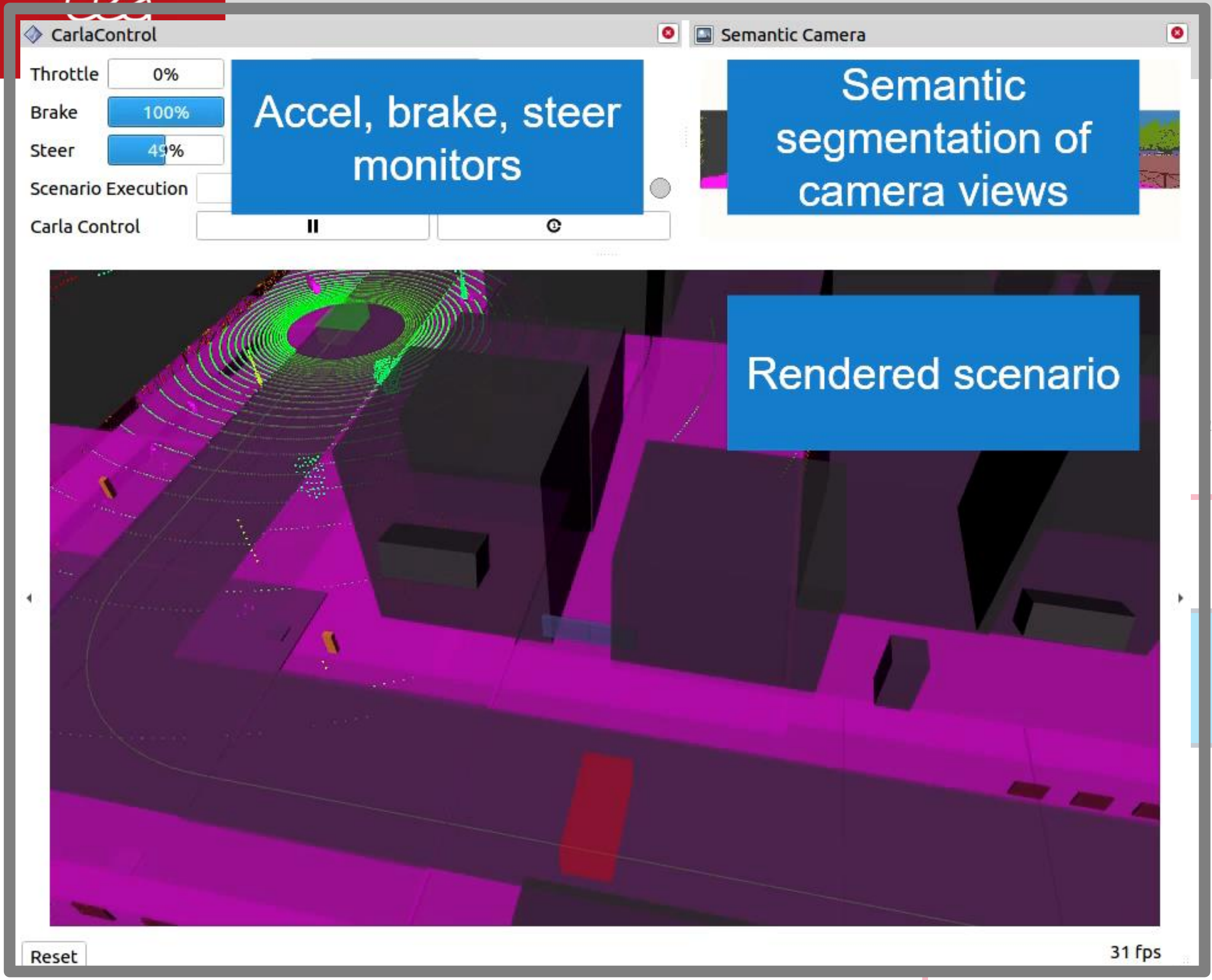


```
rclcpp_action::GoalResponse Navigation_impl::NavigateToPose_goal(
        const rclcpp_action::GoalUUID & /*in*/uuid,
        std::shared_ptr<
                const mobilemanipulationservicedef::action::MoveToCS::Goal> /
*in*/goal) {
}

/**
 *
 * @param goal_handle
 * @return return
 */
rclcpp_action::CancelResponse Navigation_impl::NavigateToPose_cancel(
        const std::shared_ptr<
                rclcpp_action::ServerGoalHandle<
                        mobilemanipulationservicedef::action::MoveToCS>> /*in*/
goal_handle) {
}

/**
 *
 * @param goal_handle
 */
void Navigation_impl::NavigateToPose_accepted(
        const std::shared_ptr<
                rclcpp_action::ServerGoalHandle<
                        mobilemanipulationservicedef::action::MoveToCS>> /*in*/
goal_handle) {
}

/**
 *
 * @param odometry
 */
void Navigation_impl::odom_handler(
        const nav_msgs::msg::Odometry::SharedPtr /*in*/odometry) {
}

/**
 *
 * @param laserscan
 */
void Navigation_impl::laser_handler(
        const sensor_msgs::msg::LaserScan::SharedPtr /*in*/laserscan) {
```

ros2 SW architecture of ego vehicle(s) (automatically-generated from MBSE specs)

simulation traces

Simulation (CARLA)



Throttle 0%
Brake 100%
Steer 49%
Scenario Execution
Carla Control

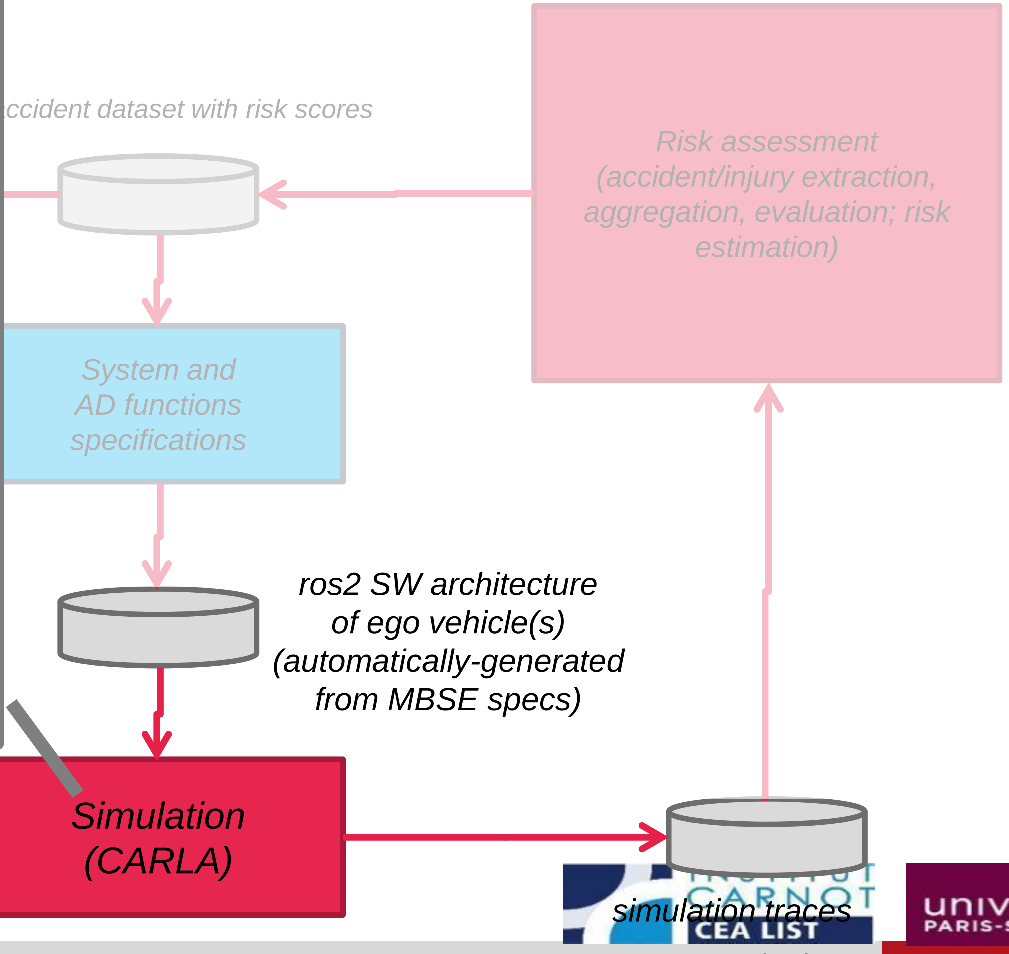
Accel, brake, steer monitors

Semantic segmentation of camera views

Rendered scenario

Reset

31 fps



ccident dataset with risk scores

Risk assessment (accident/injury extraction, aggregation, evaluation; risk estimation)

System and AD functions specifications

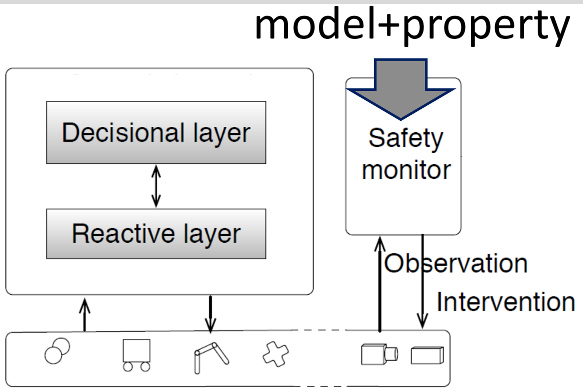
ros2 SW architecture of ego vehicle(s) (automatically-generated from MBSE specs)

Simulation (CARLA)

concrete scenarios (e.g., scenario_runner)

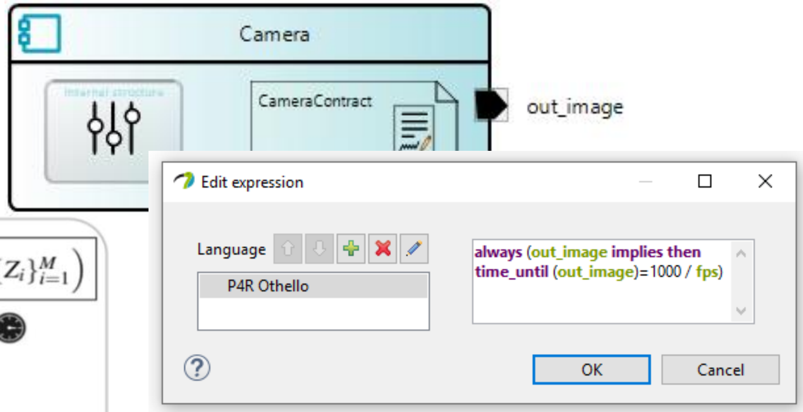
simulation traces

On-line verification of safety properties



- No complex system can be considered as fault-free
  - Unspecified situations may also induce a hazardous behavior
  - Safety monitors observe the system and its environment, and trigger interventions to keep the system in a safe state

- Approaches
  - Automated generation of run-time monitors from property specifications (in models)

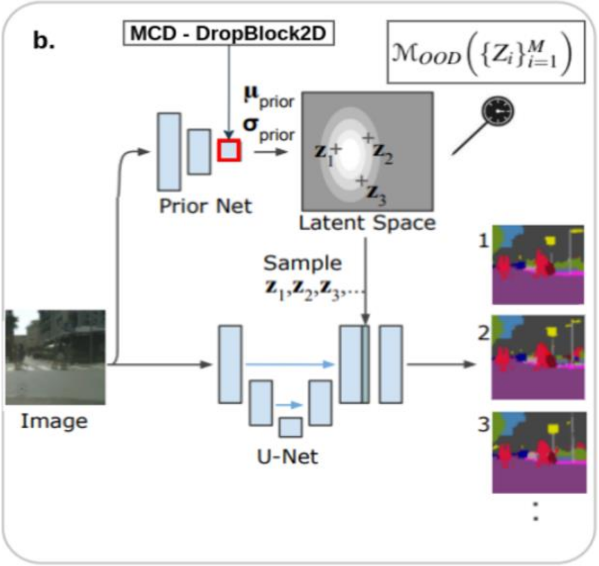


P4R Othello

always (out_image implies then time_until (out_image)=1000 / fps)

  - Use the uncertainty from intermediate latent features for OoD detection in a semantic segmentation tasks

    CEA built a (data-driven) monitoring function for OoD detection using latent-feature uncertainty



Conclusions

- ***Papyrus for Robotics***
  - “umbrella framework that collects a set of Papyrus-based DSLs and tools and supports the design of robotic systems in conformance with the RobMoSys approach”
  - Support code generation to ROS2 with roundtrip engineering capabilities
  - Provides plugins and bridges to external technologies to support safety assessment of autonomous systems

- **Identification of critical system functions based on safety standards**
  - Papyrus for Robotics supports HARA, FMEA, FTA

- **Functional safety through anticipation of faults’ impacts on the system**
  - Papyrus for Robotics supports simulation-based FI

- **Guidance on measures to ensure the safety of the intended functionality (SOTIF)**
  - Combined process based on knowledge engineering and simulation for the identification and evaluation of unsafe scenarios in autonomous driving systems
  - Run-time monitoring of safety properties
    - Automated generation of run-time monitors from property specifications (in models)
    - data-driven monitoring for OoD detection in a semantic segmentation tasks using latent-feature uncertainty