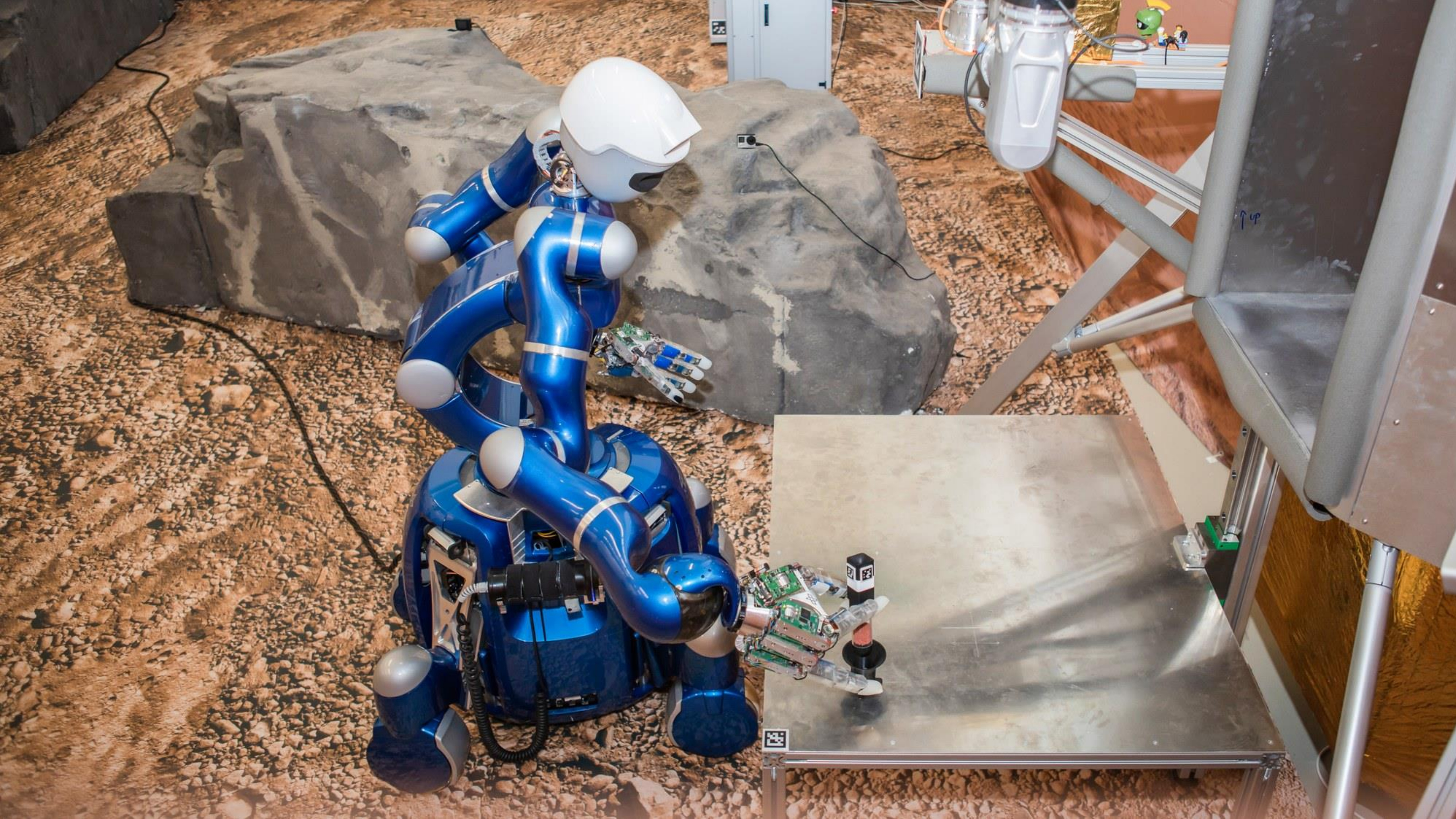
RISK-DRIVEN TESTING AND CERTIFICATION IN SIMULATED ENVIRONMENTS

Prof. Dr. Michael Felderer
Institute of Software Technology
German Aerospace Center (DLR)

universität innsbruck

UNIVERSITY OF COLOGNE

DLR


UP


SOFIA
STRATOSPHERIC OBSERVATORY
FOR INFRARED ASTRONOMY
NASA
DLR
N747NA


699M
H2
HYDROGEN
ZERO E
renfe
adif
CAF
TOYOTA

DLR
S
U
P
Deutsches Zentrum
für Luft- und Raumfahrt
DLR

Testing and Certification are Key in Safety-Critical Domains Like Aerospace

DLR in Numbers

10,000 Employees

20% develop software

55 Institutes and Facilities

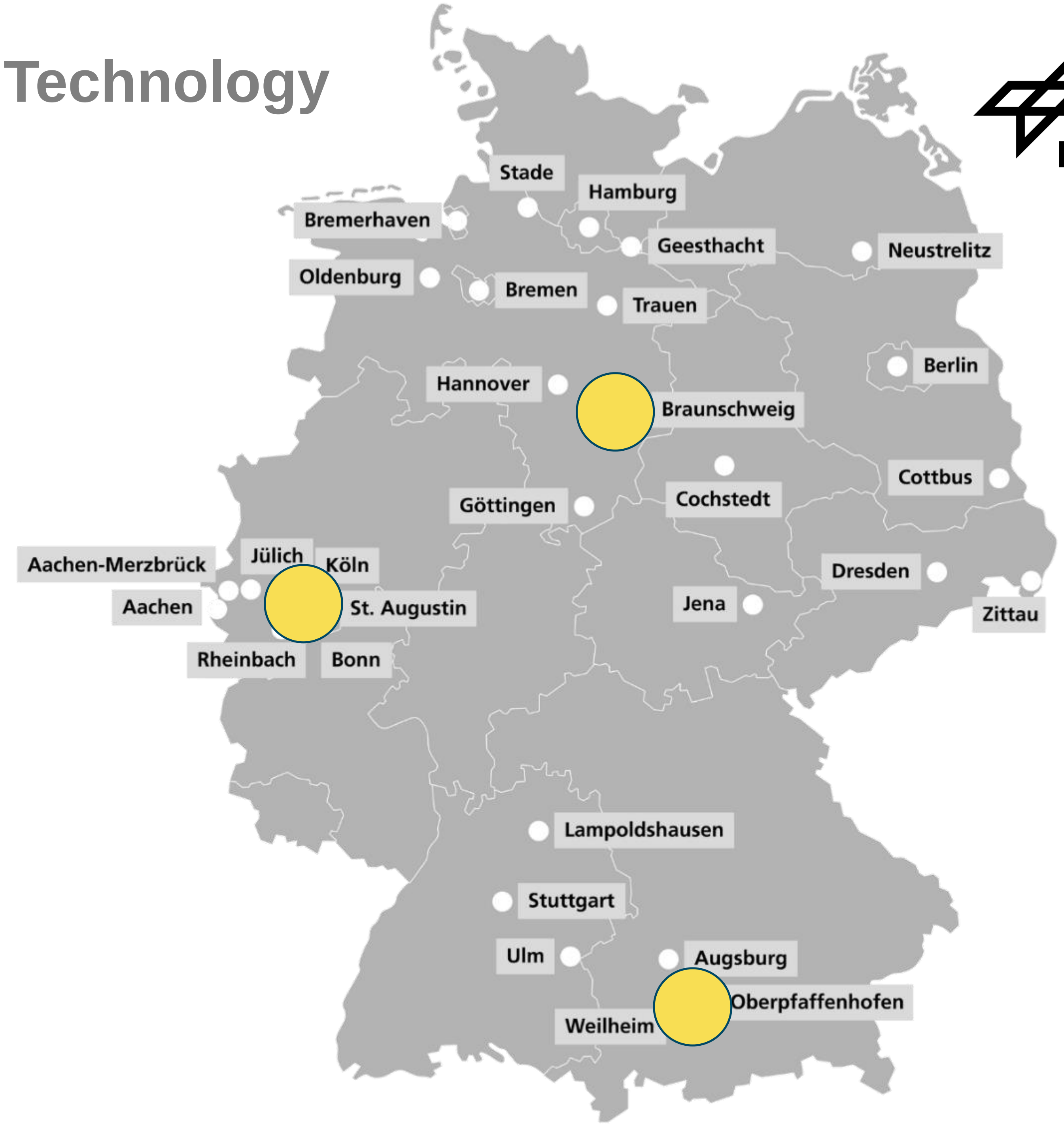
35 Locations and Offices

DLR Institute of Software Technology

200 Employees

4 Departments

3 Main Locations

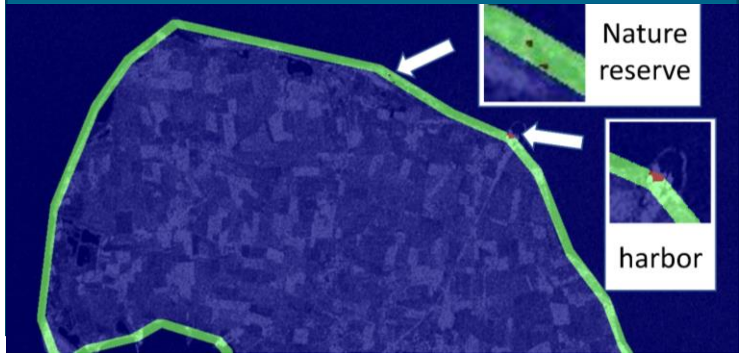


https://www.dlr.de/sc/

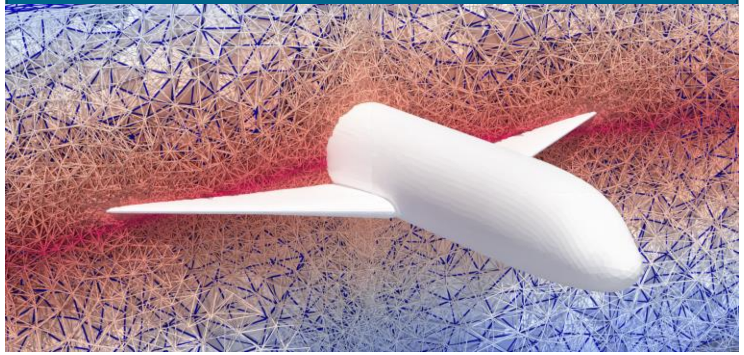
Topics at the Institute of Software Technology

Dependable, Safe and Secure Software Systems

Artificial Intelligence



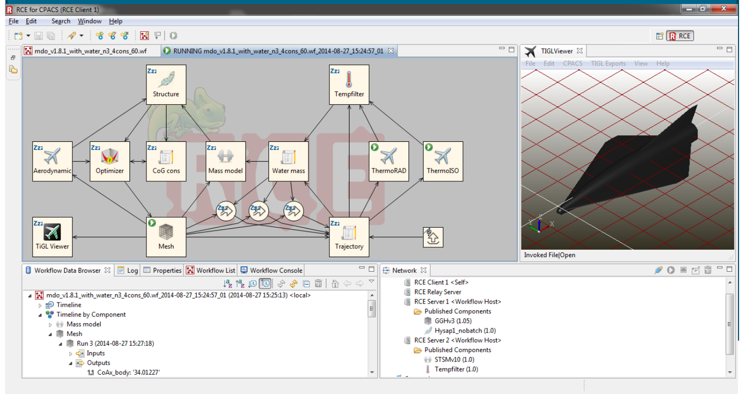
High Performance Computing and Quantum Computing



Human-System-Interaction and Visualisation

Software and Systems Engineering

Digital Twins and Digital Platforms



Outline

Testing Collaborative AI Systems in Simulated Environments

Simulation Software as Research Software

Risk-driven Certification in Simulated Environments

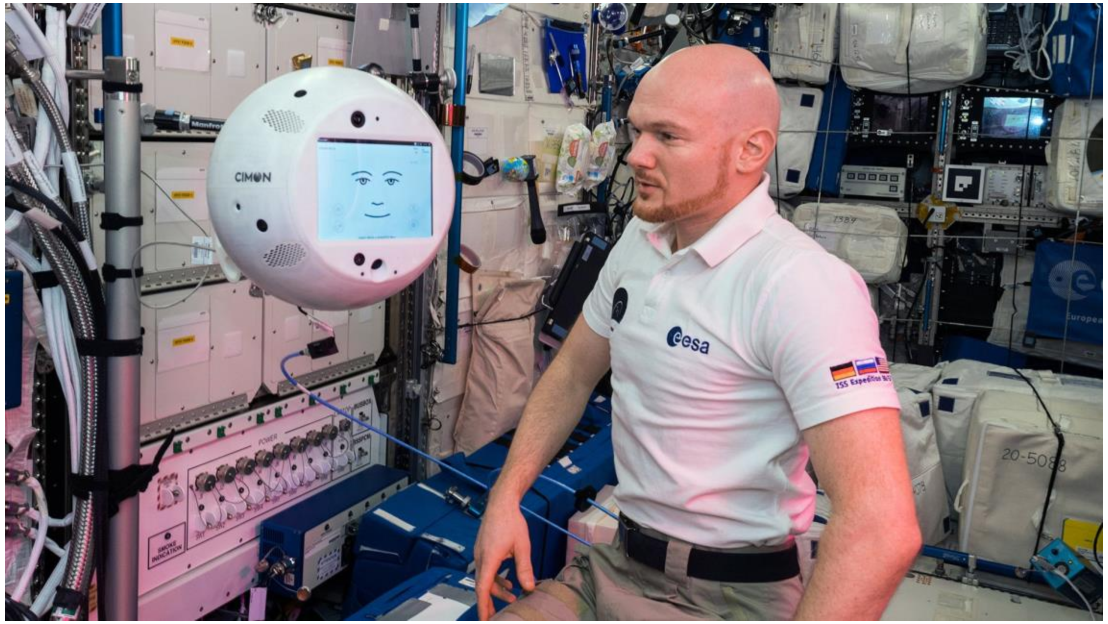
Outline

Testing Collaborative AI Systems in Simulated Environments

Simulation Software as Research Software

Risk-driven Certification in Simulated Environments

Human AI Collaboration (in Space)

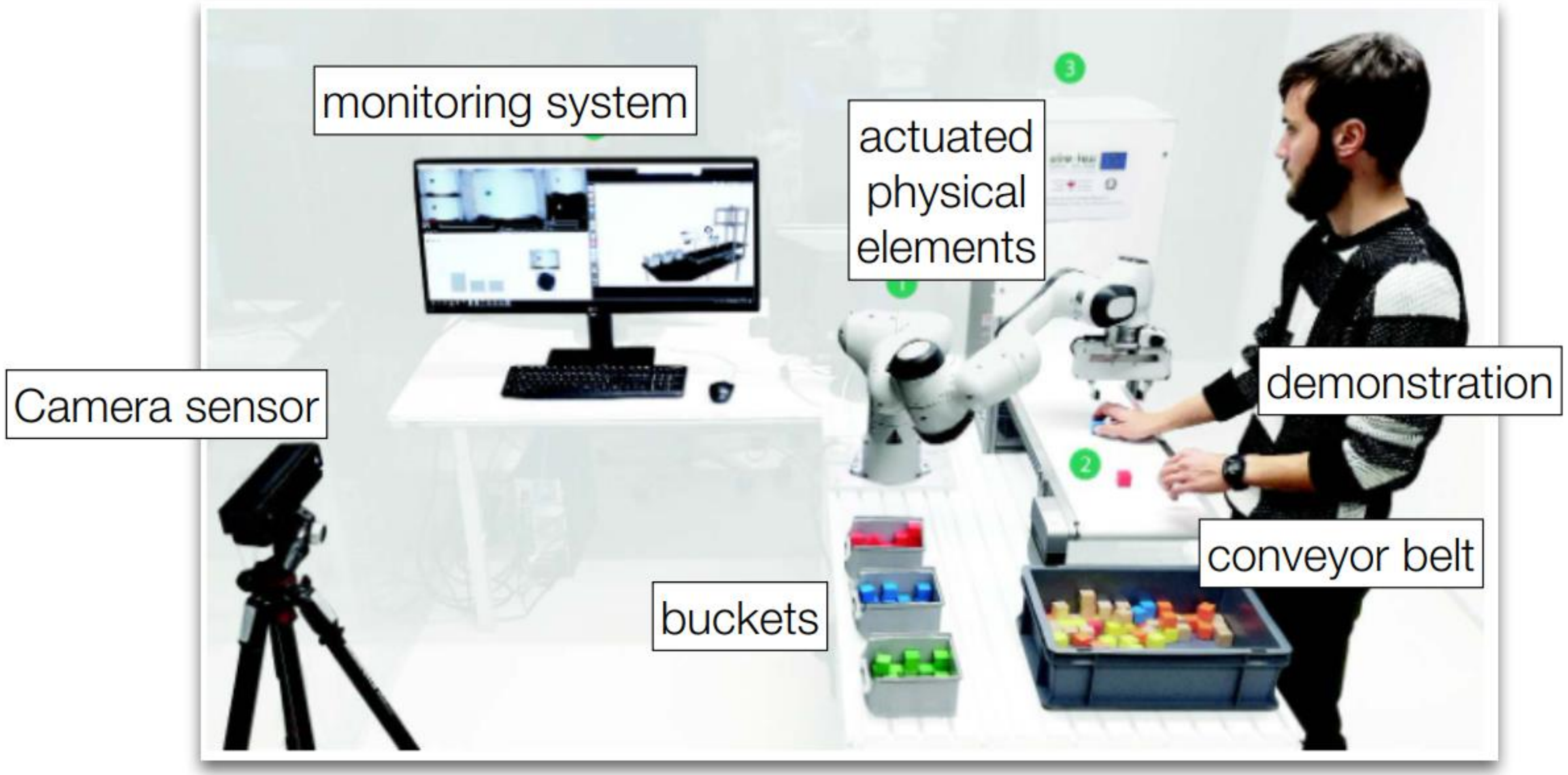


Space Station

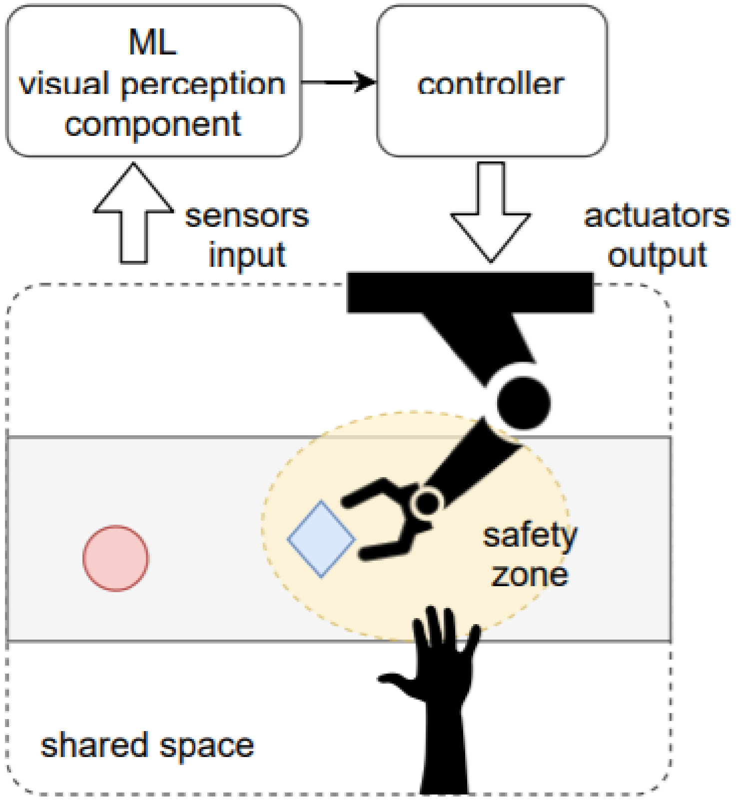
Mars Base

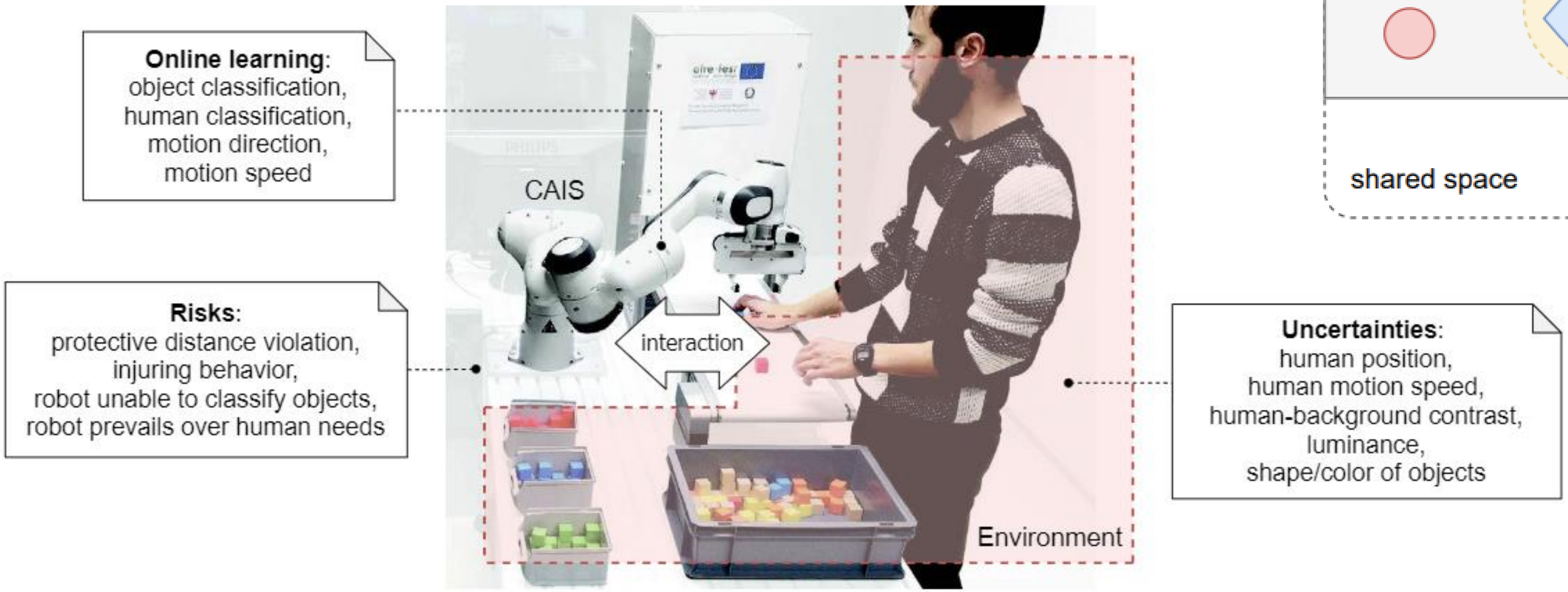
Collaborative Artifical Intelligence System (CAIS)

A **Collaborative Artificial Intelligence System (CAIS)** involves multiple agents, in this case, machine equipped with human-like abilities e.g. vision sensing, and humans working together to achieve common goals, improving efficiency and outcomes in complex tasks.



Testing of a Collaborative AI System (CAIS)

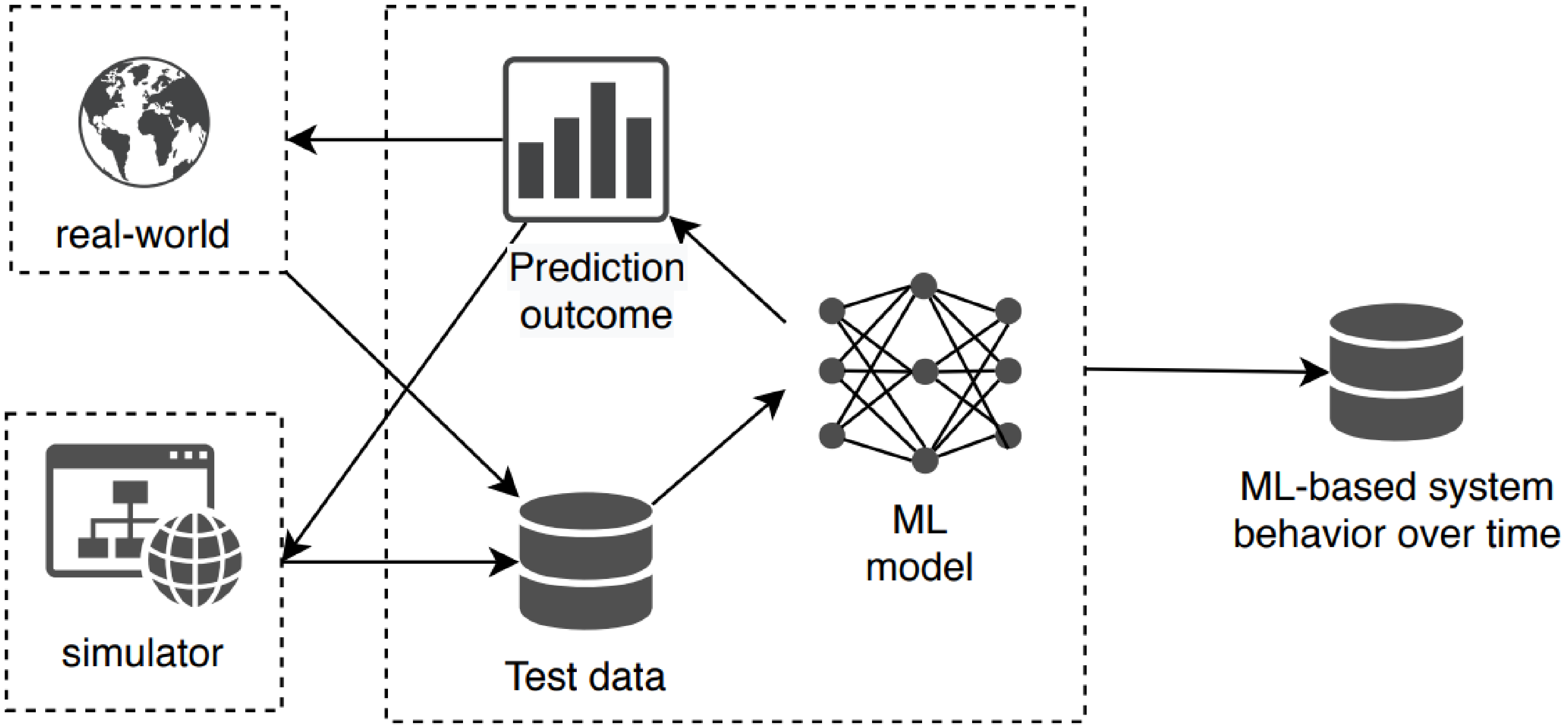


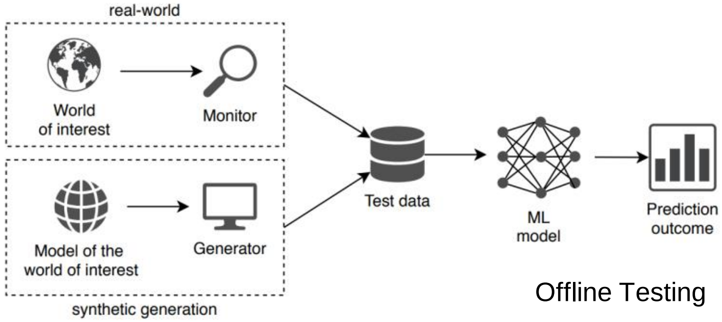


Online Testing

Testing ML model in real or simulated environment

ML model tested as a unit in closed loop mode





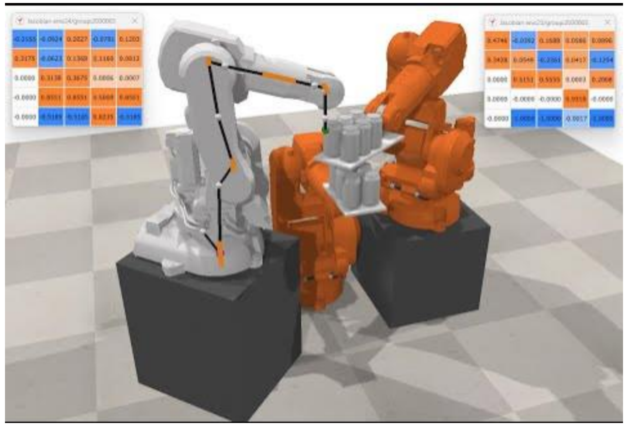
Offline Testing

Testing in Simulated Environments

Simulation/Simulated Environments (simulators) are computer program environments that allow imitation of real-world processes/systems under controlled conditions

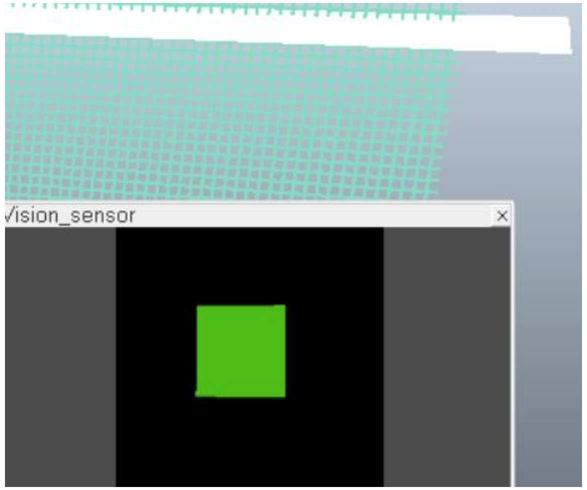
Types are for instance

- software-based simulations
- physical mockups, or
- virtual reality environments

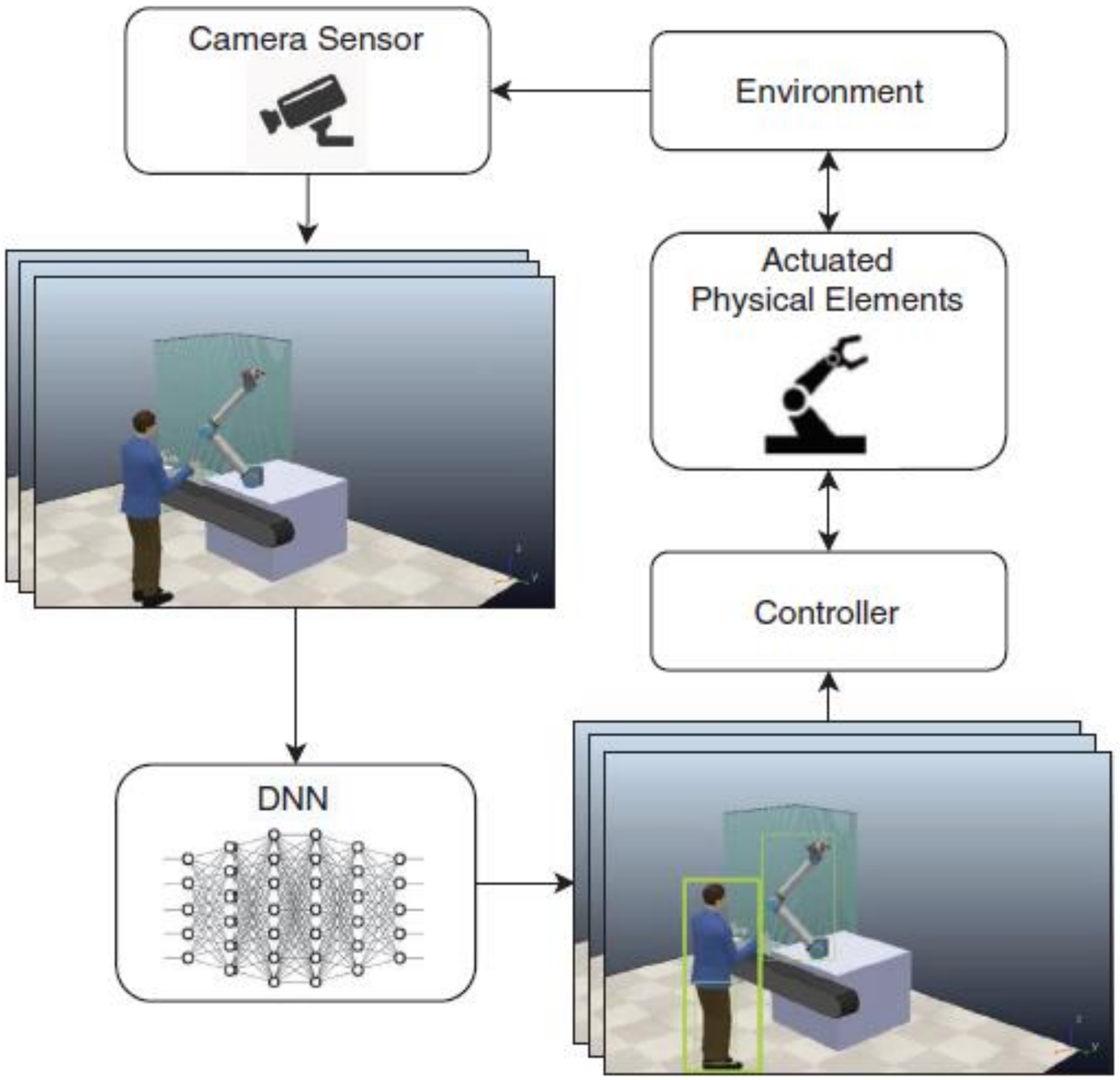


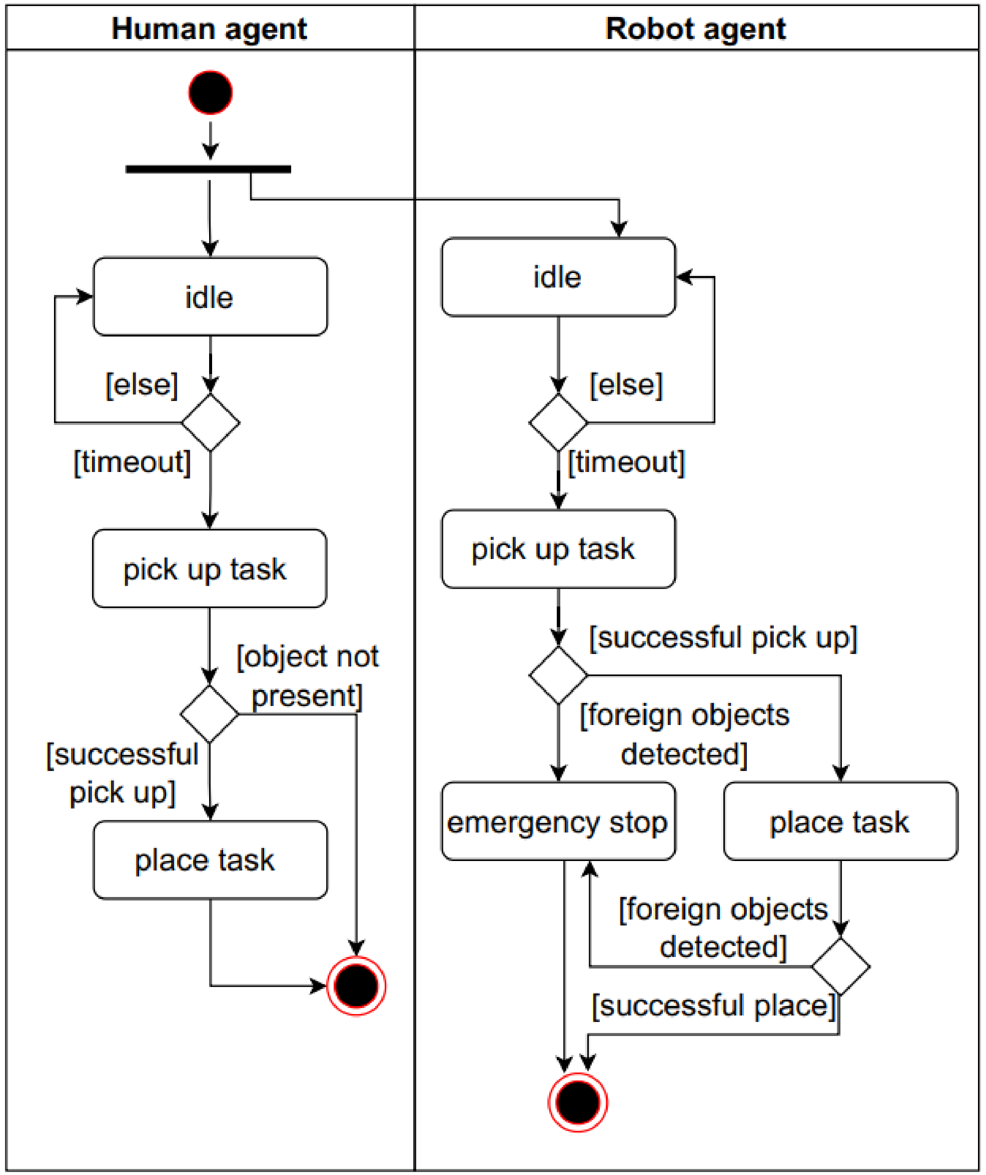
Simulated Environment

Non-trivial implementation of an industrial collaborative system simulation



Simulation Process





Application of Simulations

- **Robotics and Manufacturing:** check computer vision, reinforcement learning
- Autonomous Driving: test lane keeping capabilities, object detection, maneuvering
- Software Development: identifying bugs, and ensuring software stability
- Medical Device Development: simulating patient interactions, evaluating device performance
- Aviation Training: practice emergency procedures in a safe, simulated environment
- Cybersecurity: identify vulnerabilities in network systems
- **Aerospace Engineering**: simulation of flight behavior
- …

Issues of Simulated Environments

1	2	3	4	5
Developing simulation models requires expert knowledge	Restrictions of simulation environments	Simulation results may be difficult to interpret	Modelling and analysis can be time-consuming	Simulation is resource-intensive and often requires HPC

Simulation is expensive and benefits from a risk-driven approach driving scenario selection

Additional Testing Challenges

1	2	3	4
Representative test cases covering failure diversity	Unpredictable behavior of agents at runtime	Identification of critical and meaningful assurance cases	Limited testing resources

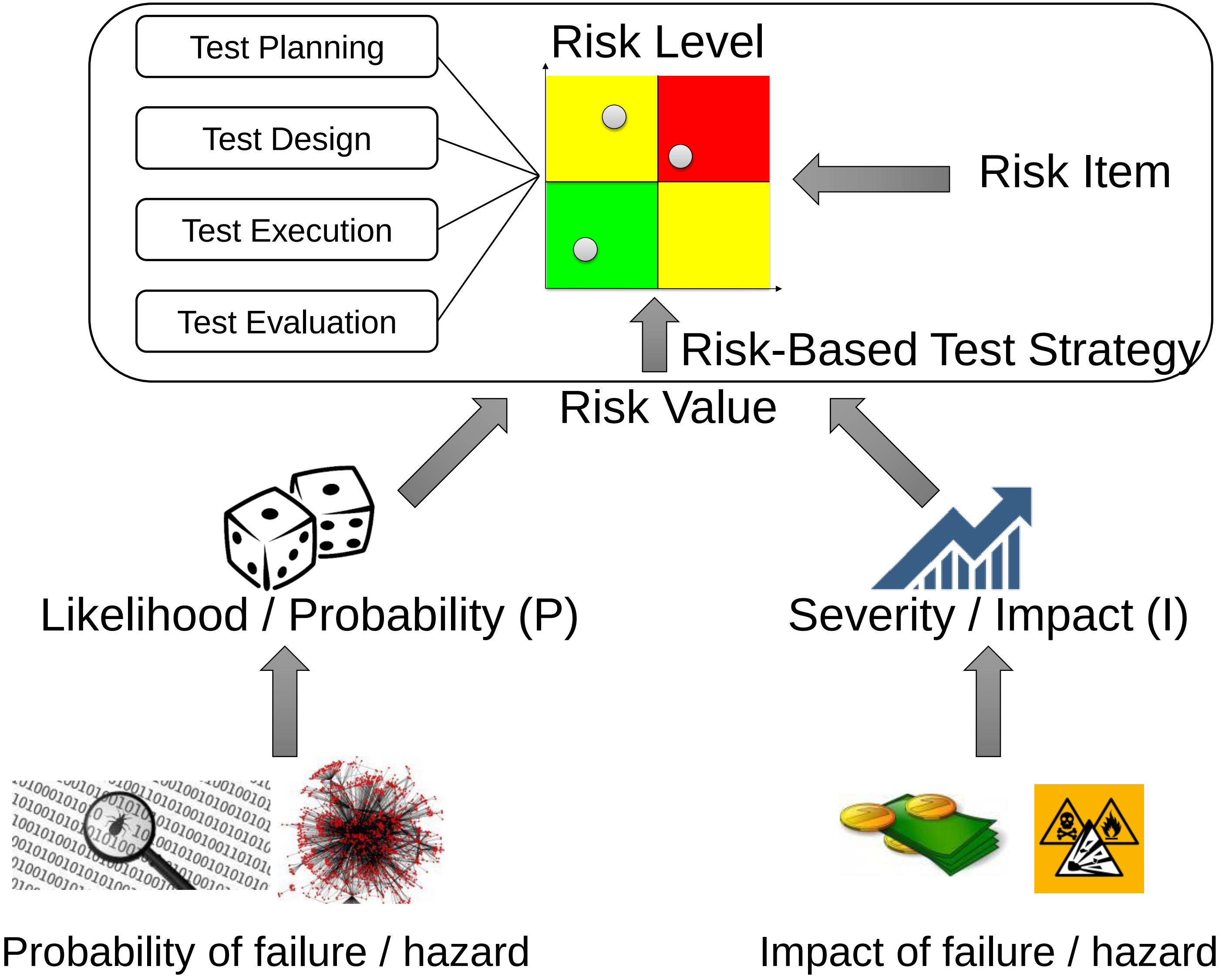
Risk-driven and search-based approach to testing and **test case diversity analysis** in **simulated environments** offer promising solutions to these challenge

Risk-Driven Testing

Risk-driven testing aligns testing activities with the real-world constraints which

- Identifies potential weaknesses and failure points in a system.
- Analyzes the likelihood and severity of each risk.
- Prioritizes testing efforts to focus on high-risk areas.

Risk Assessment and Risk-Based Testing



Underlying Publications


Collaborative Artificial Intelligence Needs Stronger Assurances Driven by Risks

Jubril Gbolahan Adigun, University of Innsbruck
Matteo Camilli, Free University of Bozen–Bolzano
Michael Felderer, University of Innsbruck and Blekinge Institute of Technology
Andrea Giusti, Fraunhofer Italia Research
Dominik T. Matt, Free University of Bozen–Bolzano and Fraunhofer Italia Research
Anna Perini, University of Trento
Barbara Russo, Free University of Bozen–Bolzano
Angelo Susi, Fondazione Bruno Kessler

Adigun, J., Camilli, M., Felderer, M., Giusti, A., Matt, D., Perini, A., Russo, B., Susi, A. (2022) Collaborative AI Needs Stronger Assurances Driven by Risks. Computer, 55(3), IEEE

2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)

Risk-driven Online Testing and Test Case Diversity Analysis for ML-enabled Critical Systems

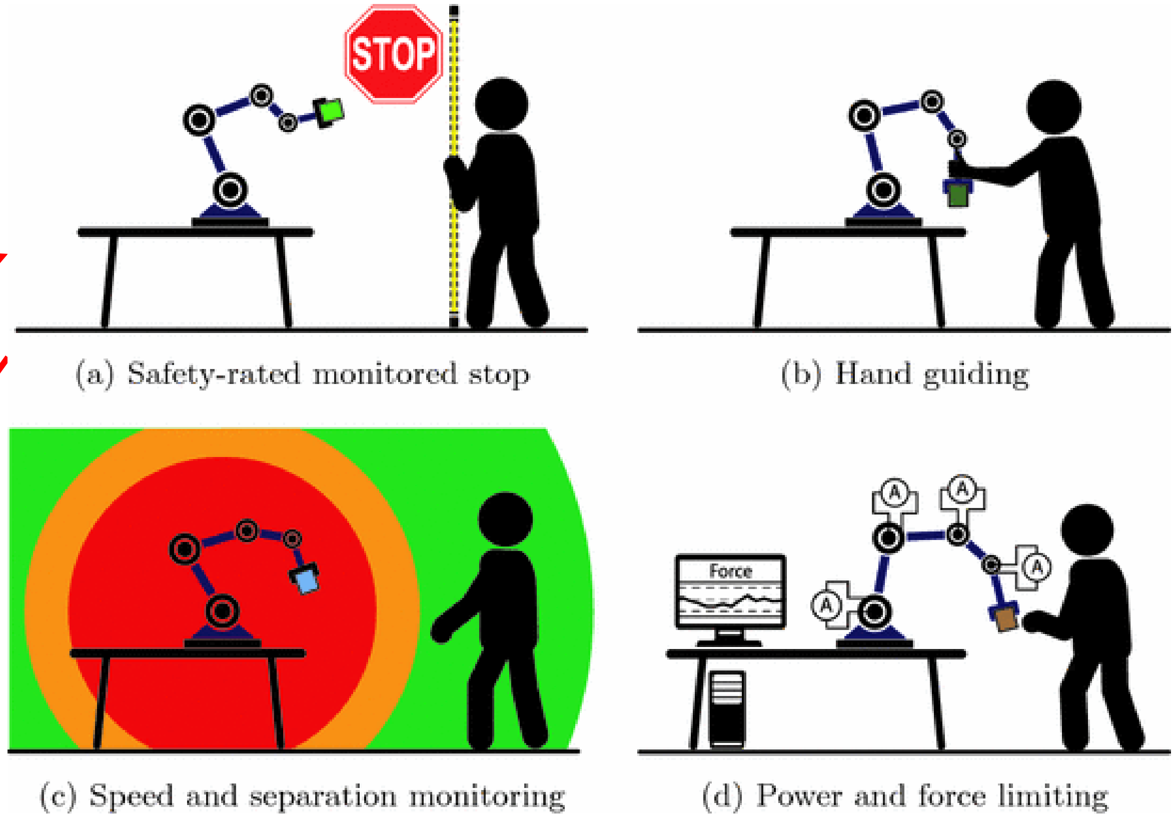
Jubril Gbolahan Adigun[*⊥♠], Tom Philip Huck[¶], Matteo Camilli[†], Michael Felderer[*‡§]
[*] University of Innsbruck, Austria
Email: {first}.{last}@uibk.ac.at
⊥ Ainnov8 Technologies Ltd, Nigeria
Email: jubril@ainnov8.com
♠ Center for Artificial Intelligence (AI) Research Nepal, Nepal
Email: jubril.adigun@cair-nepal.org
¶ Karlsruhe Institute of Technology (KIT), Germany
Email: tom.huck@kit.edu
† Politecnico di Milano, Italy
Email: matteo.camilli@polimi.it
‡ German Aerospace Center (DLR), Germany
Email: michael.felderer@dlr.de
§ University of Cologne, Germany
Email: michael.felderer@uni-koeln.de

Adigun, J., Huck, T., Camilli, M., Felderer, M. (2023) Risk-driven Online Testing and Test Case Diversity Analysis for ML-enabled Critical Systems. ISSRE 2023, IEEE

Industrial Robot Safety

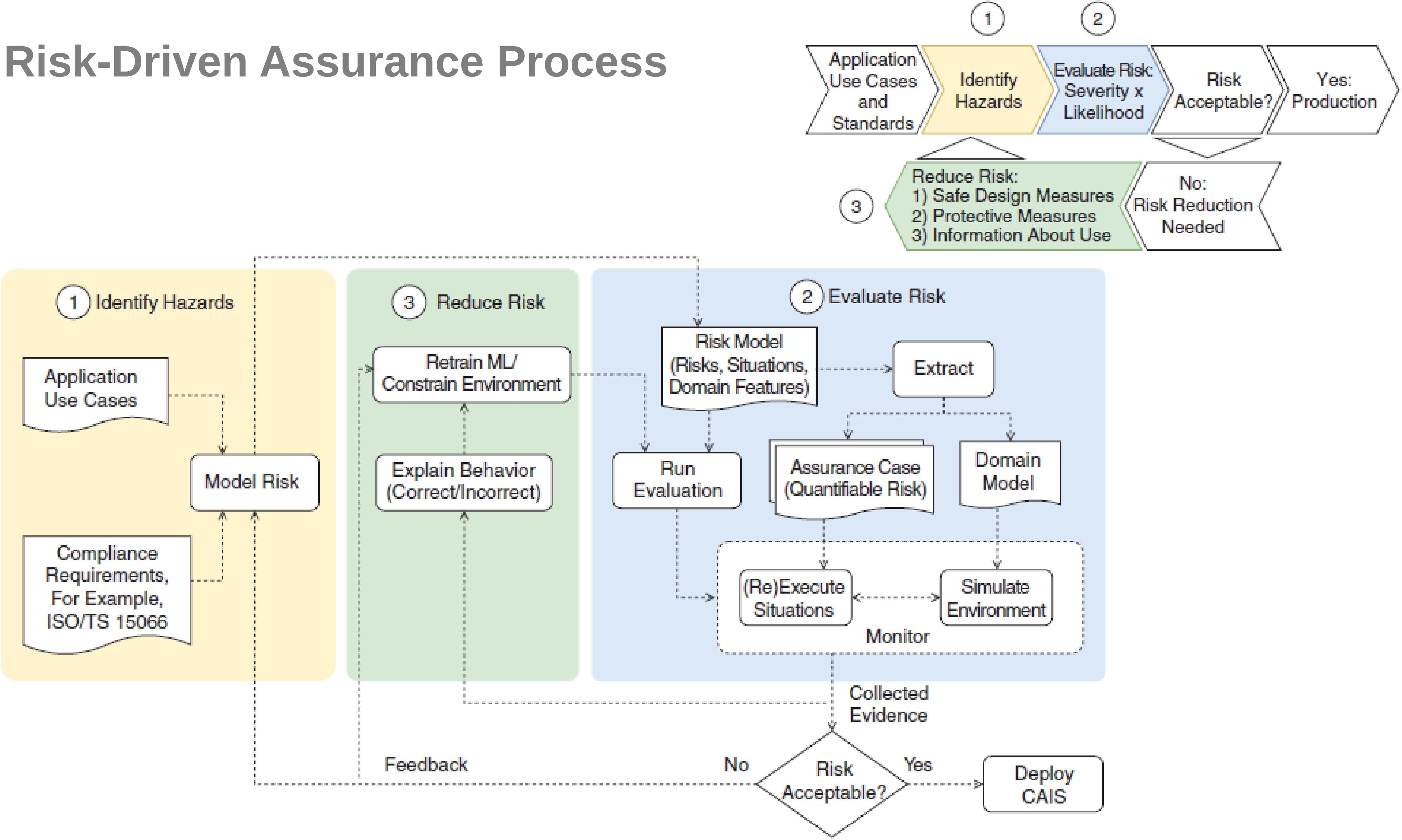
ISO 10218 and **ISO/TS 15066** which specify risk management processes for robots and robotic devices and safety requirements for industrial robots and collaborative industrial robots define **four collaborative operating modes**

Our work is based on the **Safety-rated monitored stop** operating mode

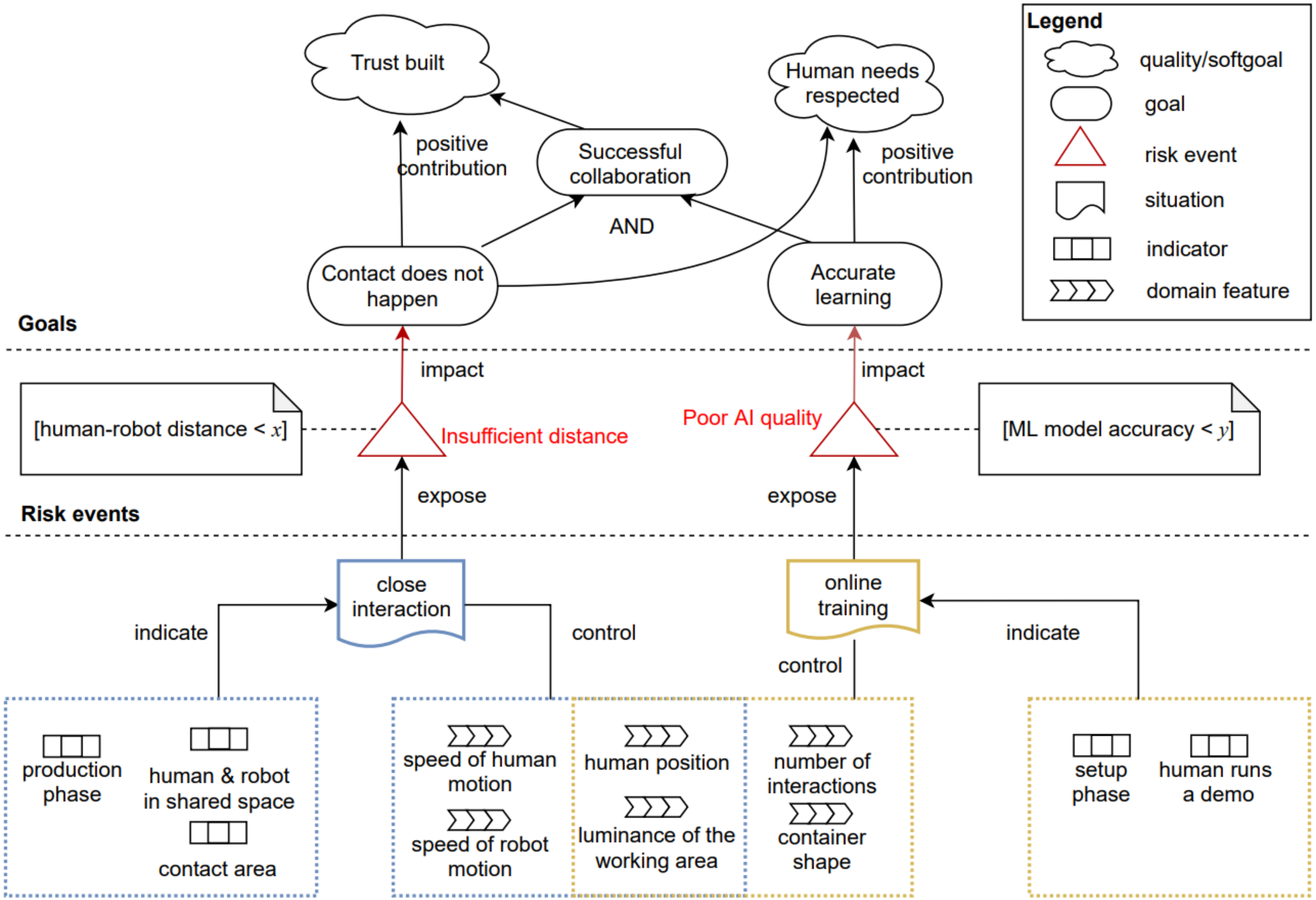


Risk assessment to deal with ML and related risks in CAIS not considered in current standards like ISO 10218 or ISO/TS 15066

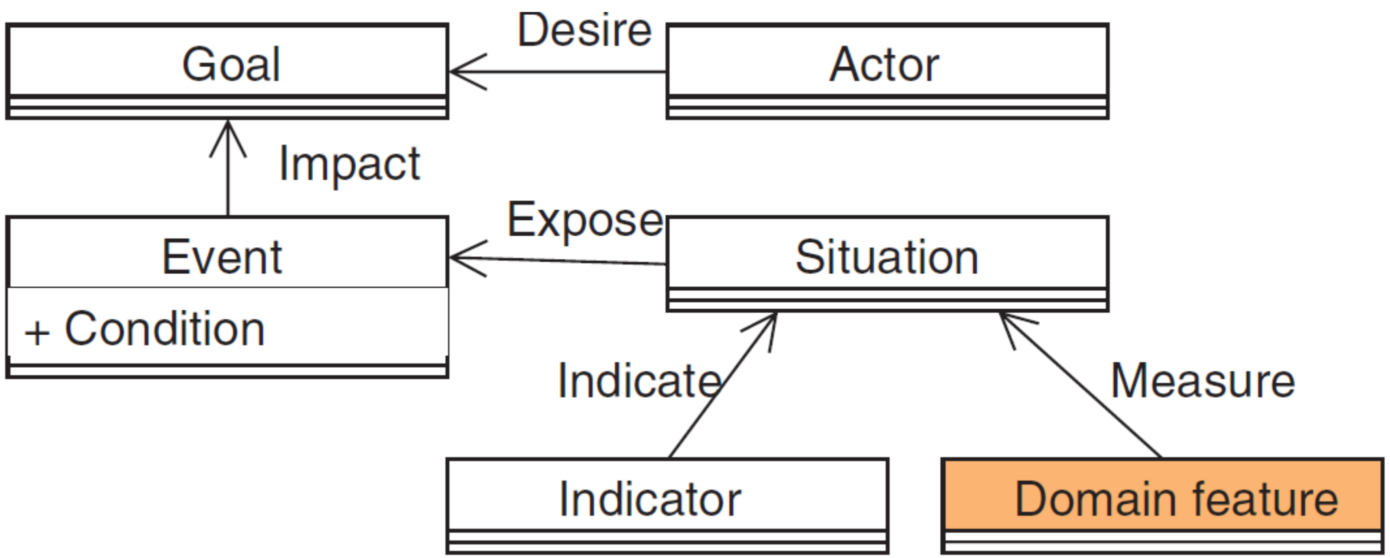
Risk-Driven Assurance Process

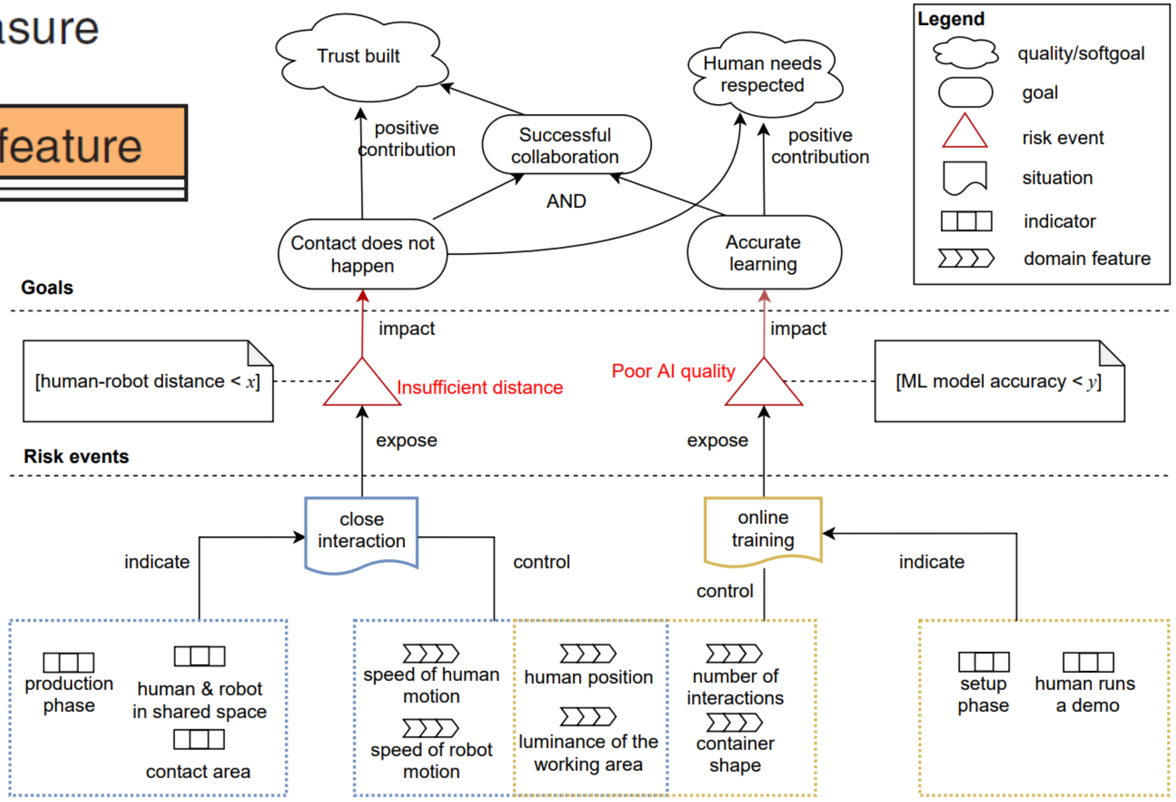


Hazard Identification

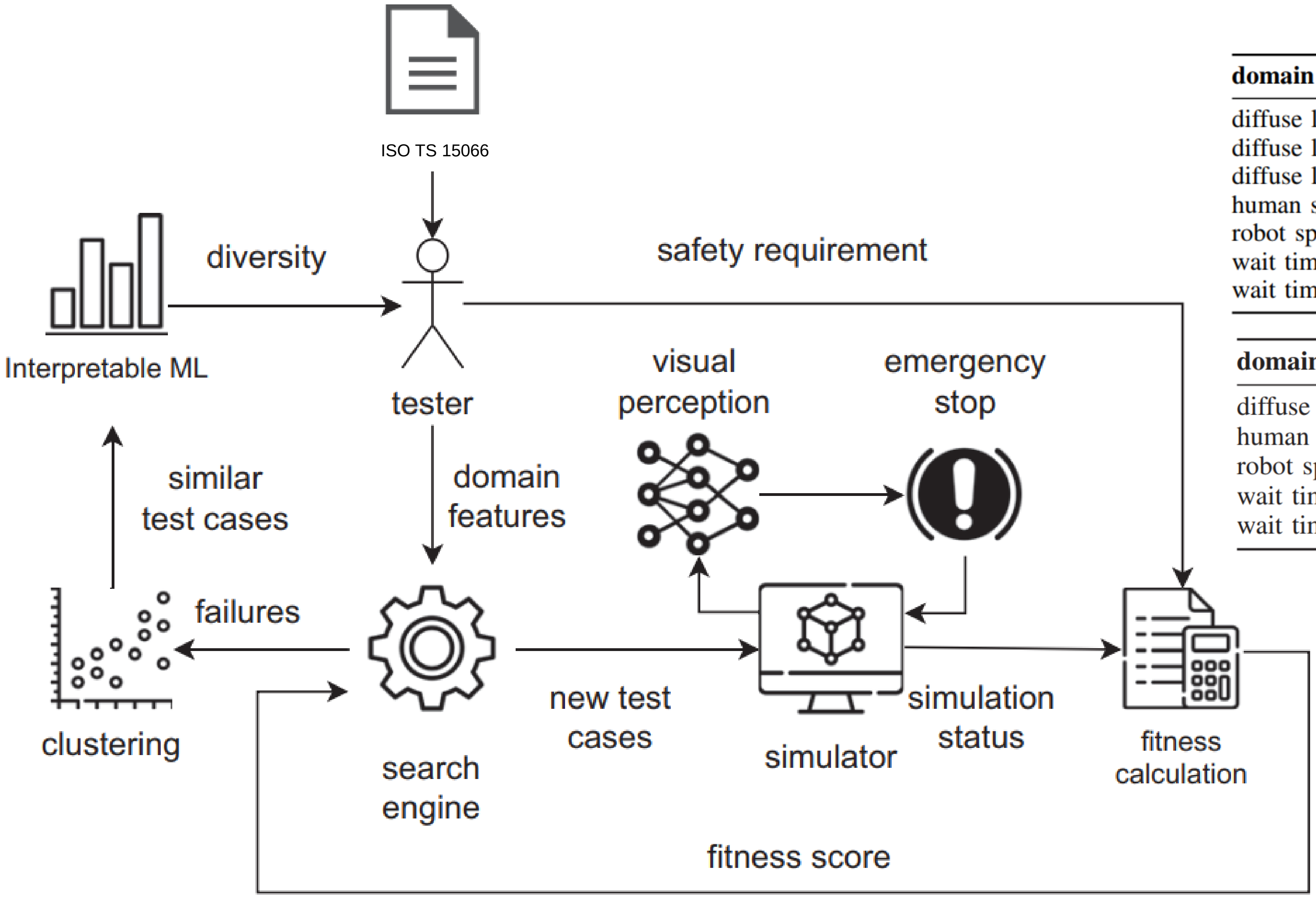


RiskML Metamodel





Risk-driven Online Testing



domain feature	type	lower bound	upper bound
diffuse light (R)	float	0.0	1.0
diffuse light (G)	float	0.0	1.0
diffuse light (B)	float	0.0	1.0
human speed (m/s)	float	0.1	0.5
robot speed (m/s)	float	0.05	0.5
wait time human (s)	integer	1	50
wait time robot (s)	integer	1	50

domain feature	value
diffuse light 1	(0.1, 0.2, 0.3) RGB
human speed	0.33 m/s
robot speed	0.25 m/s
wait time human	2 s
wait time robot	5 s

Objective Functions:
Minimum distance between human and robot arm
Relative speed of human and robot arm

Encoding the Problem

We defined an optimization problem using different metaheuristic optimizing search algorithms to drive tests

- Random Search (RS) as a baseline
- Genetic Algorithm (GA)
- Evolutionary Strategy (ES)
- Simulated Annealing (SA)

Algorithm	Configuration parameters
GA	Polynomial mutation probability 0.143 and DI 100.0, Binary crossover probability 0.9 and DI 100.0
ES	Polynomial mutation probability 0.143, Elitist option *true*, $\lambda = 20$, $\mu = 20$
SA	Temperature $T_0 = 1.0$, Min temperature 0.000001, Temperature variation coefficient $\alpha = 0.95$, Polynomial mutation probability 0.143

jMetalPy
Python version of the jMetal framework

Encoding the Problem

Define fitness function to derive test outcome

$$\forall t \in T, \forall p \in \mathcal{S}(r_t, s_t), \|h_t - p\| > 0$$

Optimize search

$$f(\omega) = \min_{t \in T, p \in \mathcal{S}(\omega.r_t)} \|\omega.h_t - p\|$$

T is observation period,
$\|\cdot\|$ is the magnitude of the distance between locations in the (3D) collaborative space,
r_t, is the location at time t of the robot,
h_t are locations at time t of the human,
s_t is the speed of the robot at time$t$, and
ω is simulation status

Research Questions

RQ1: What is the effectiveness of the risk-driven test case generation across different search strategies?

We compared the effectiveness of the metaheuristic optimizing search algorithms against the baseline random search using statistical significance test and effect sizes

RQ2: What is the diversity of generated test cases causing hazards?

We applied cluster analysis using DBSCAN, dimension reduction with PCA and then diversity validation using Local Interpretable Model-agnostic explanations for Local Explanation Diversity (LED) measure

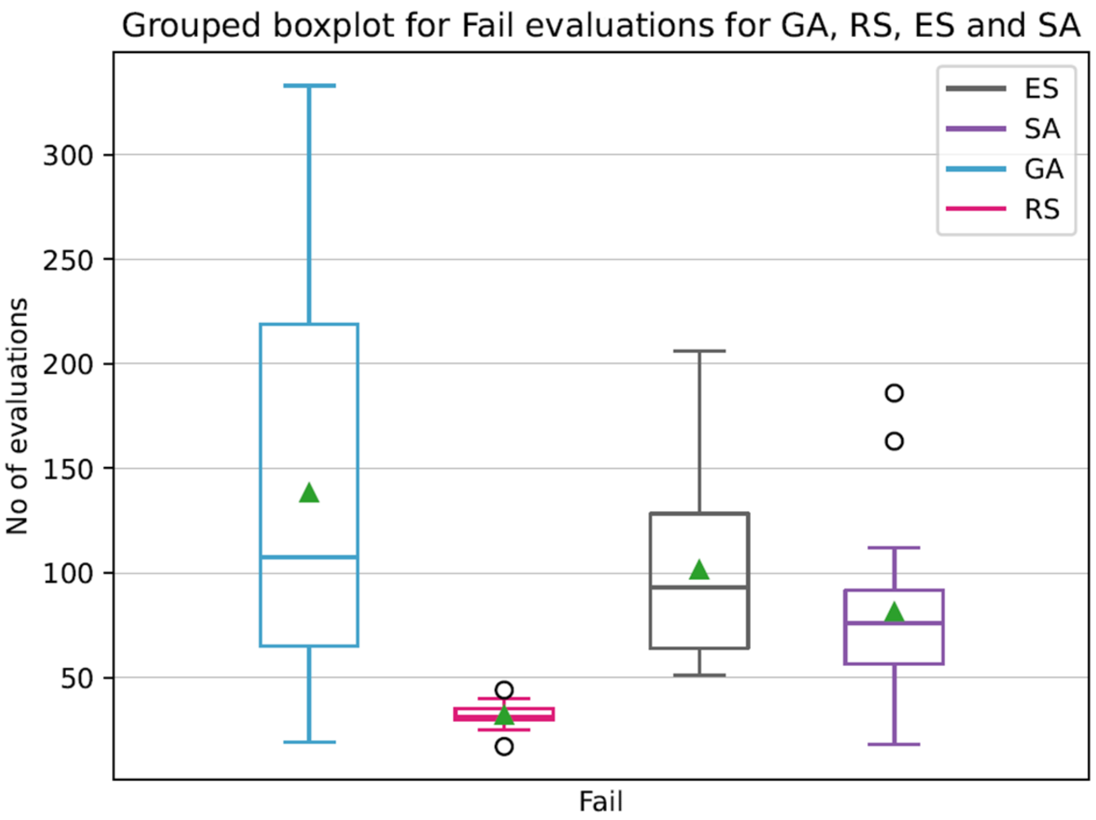
RQ3: What are the most important domain features?

We applied Shapley Additive Explanations (SHAP) feature importance explainer then developed a feature score ranking matrix to determine the average contribution of each feature

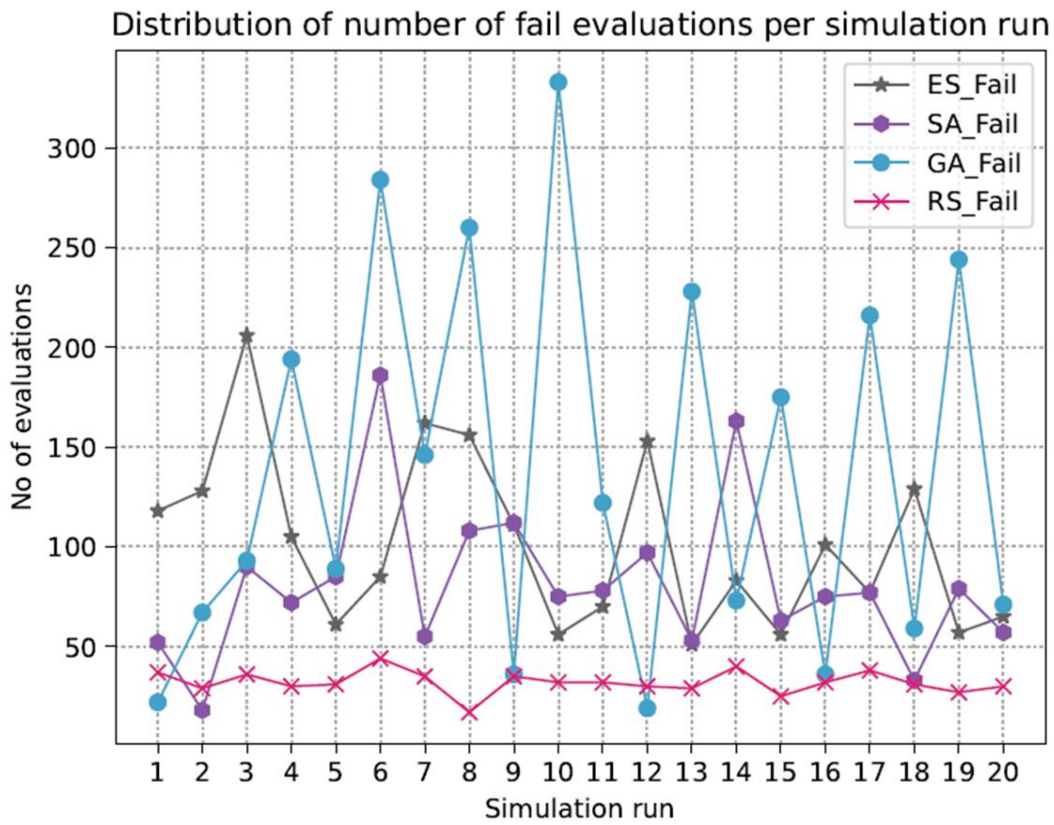
Note:

- We had **20** simulation "runs“ relating tests resulting from a particular algorithm configuration
- Per run: **400** evaluations (an instance of a concrete scenario)

Results – RQ1 (Effectiveness of Test Case Generation)



GA, ES, SA found more failed test cases respectively compared to **RS**



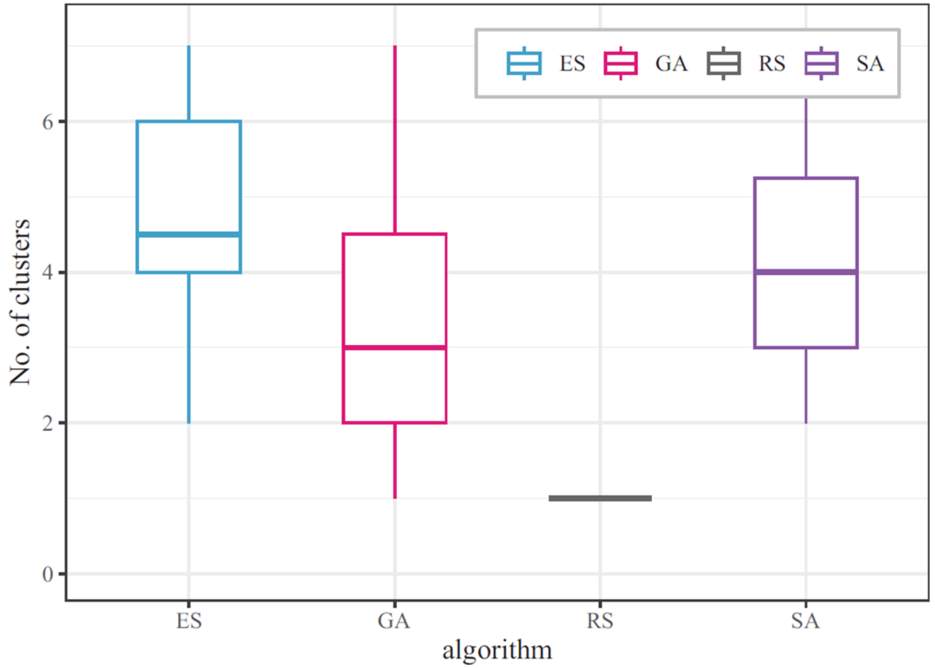
Also, **GA** showed the highest peak when all runs are considered

Groups		**Mann-Whitney U-test**		$\hat{A}_{AB}$ **effect size**	
A	**B**	U statistic	p-value	estimate	magnitude
GA	RS	353.0	0.000	0.88	L
ES	RS	400.0	0.000	1.0	L
SA	RS	374.0	0.000	0.94	L
GA	ES	168.0	0.394	0.58	S
GA	SA	142.0	0.119	0.64	M
ES	SA	251.5	0.168	0.63	S

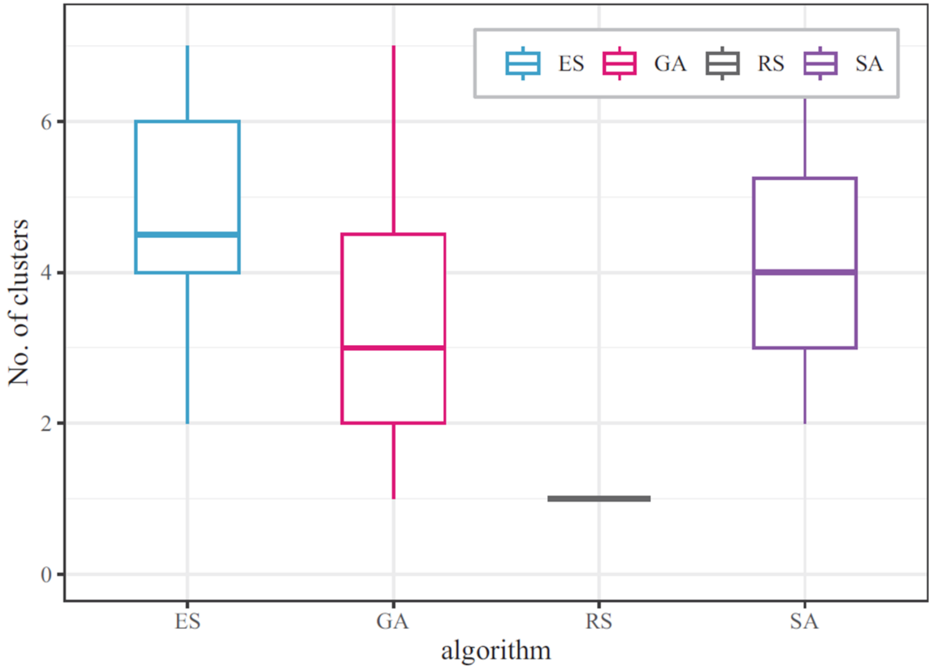
H_0: A and B are extracted from the same distribution (Null);
H_1: A and B are extracted from different distributions (Alternative).
p-value: 0.05 to reject H_0 /accept H_1

L – large, M – medium, S – small effect

Results – RQ2 (Diversity of Generated Test Cases)



(b) Intra- and inter-cluster diversity



(b) Intra- and inter-cluster diversity

The median intra-cluster LED is around ***0.25,*** *while the median inter-cluster LED is around* ***0.65*** *(almost 3 times higher)*

Local Explanation Diversity (LED) is defined as the average pairwise (normalized Levenshtein) distance between all "sorted" feature sequences (based on LIME weights) of two sets of test cases.

$$LED = avg(\{L^*(seq(A), seq(A')) \; \forall A \in C, A' \in C'\})$$

L - normalized Levenshtein distance*
A and A' – features of test cases C and C' respectively

Algorithm	#Clusters w/o PCA	#Clusters w PCA	#PCA components
ES	75	82	2
SA	69	67	2
GA	55	54	2
RS	1	1	2

(c) Clustering with and without dimension reduction

Even though **GA** yields more failures in general, both ES and SA lead to more clusters (more diversity).

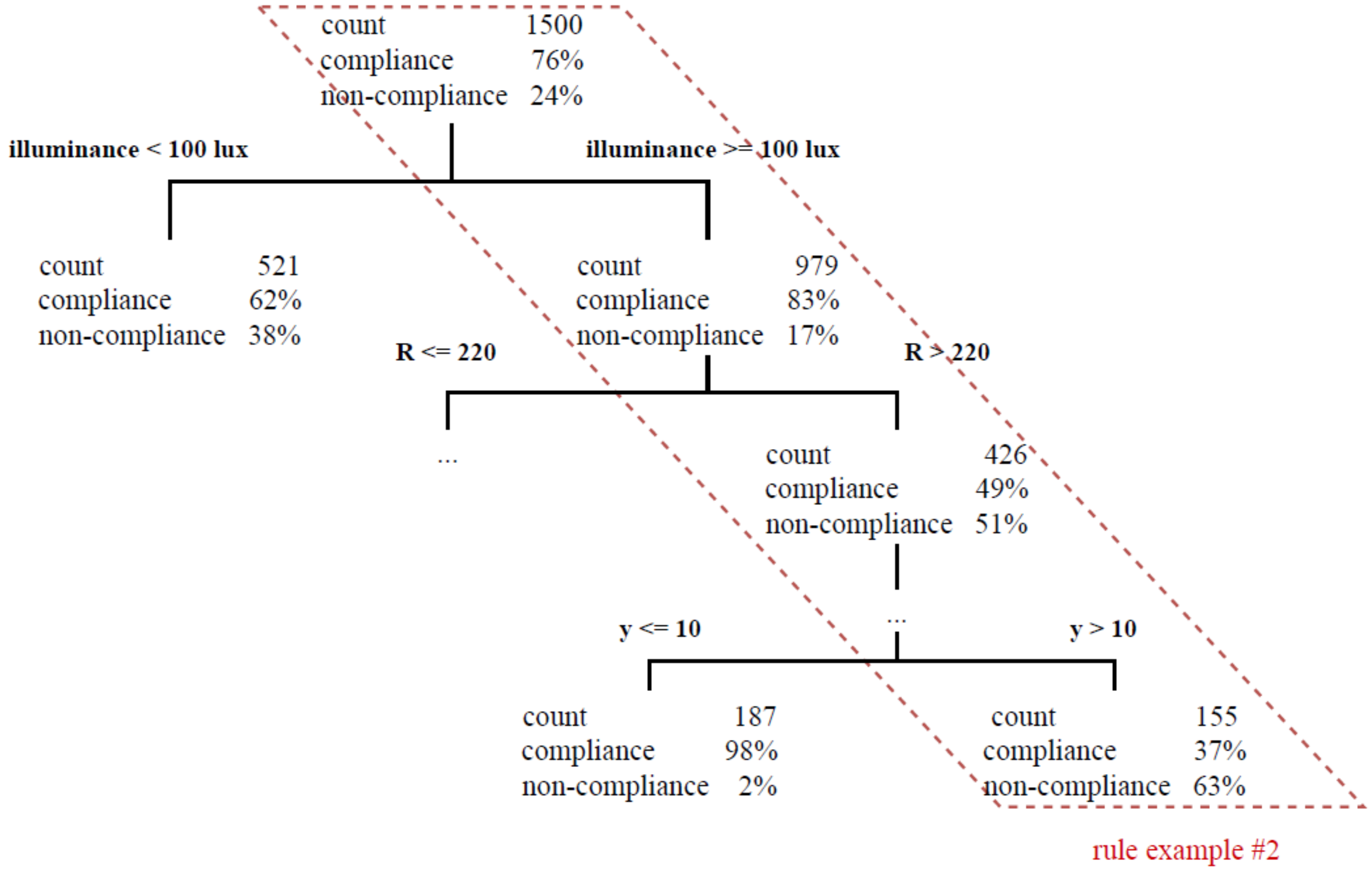
Results – RQ3 (Importance of Domain Features)

Feature	Rank (1st — 7th) ES	SA	GA	RS	Overall score ↓
robot speed	1	2	2	1	1.5
robot wait time	3	4	1	3	2.75
human wait time	4	3	3	2	3.0
human speed	6	1	4	4	3.75
diffuse light (R)	6	5	6	5	5.5
diffuse light (B)	5	6	5	7	5.75
diffuse light (G)	7	7	7	6	6.75

Robot speed has the highest importance, followed by *human speed*, *robot wait time* and *human wait time*. The three diffuse light features are generally of lower importance,

Implication - the ML visual perception component is fairly robust to changes in the lighting condition.

Further Challenge:
Understanding Hazards via Decision Trees and Rule Extraction



rule example #	domain features: illuminance (lux)	domain features: operator arms color (R,G,B)	domain features: operator position (x,y)
1	<100	-	-
2	>100	R >220, G >236, B >200	x >180, y >10

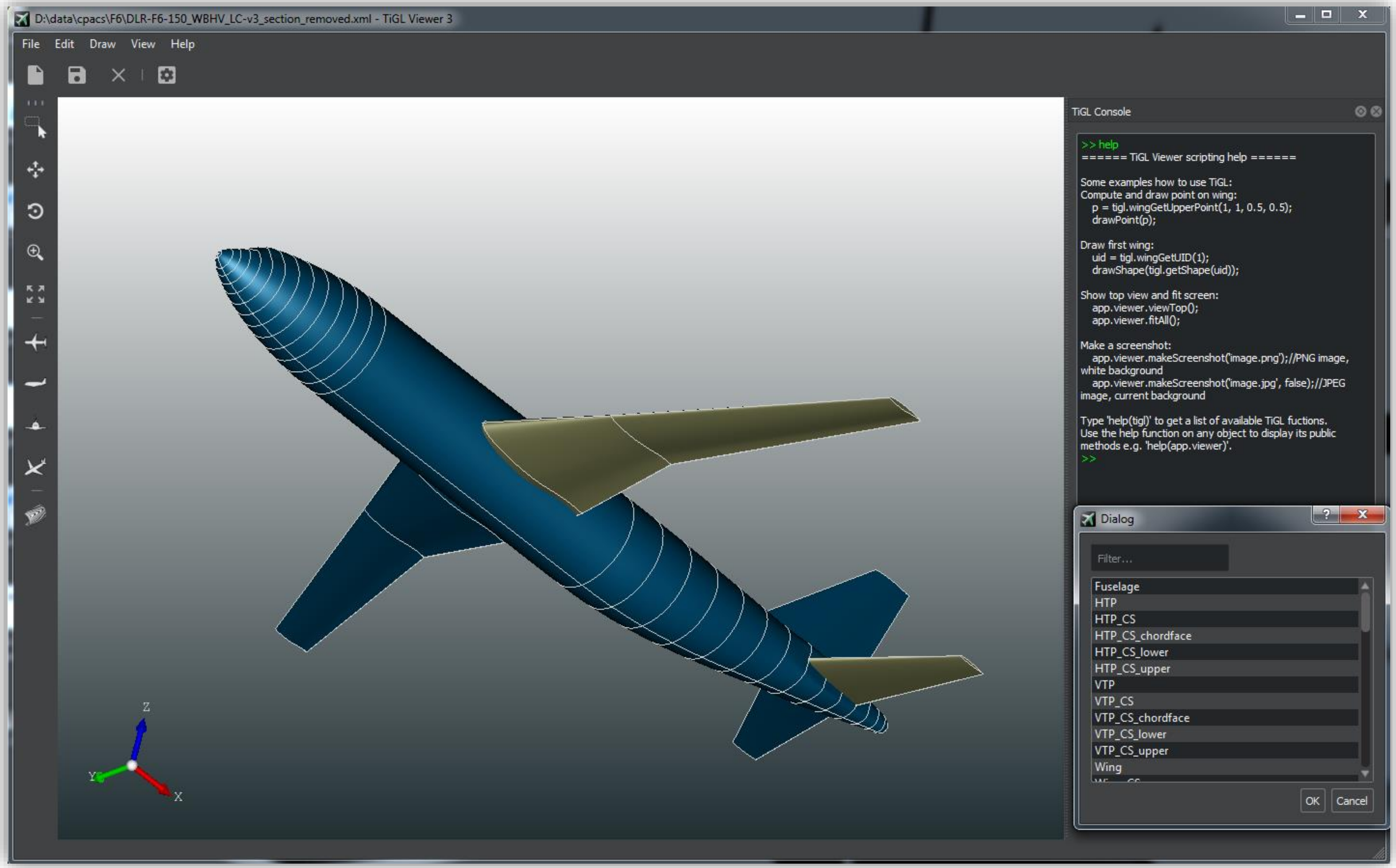
Outline

Testing Collaborative AI Systems in Simulated Environments

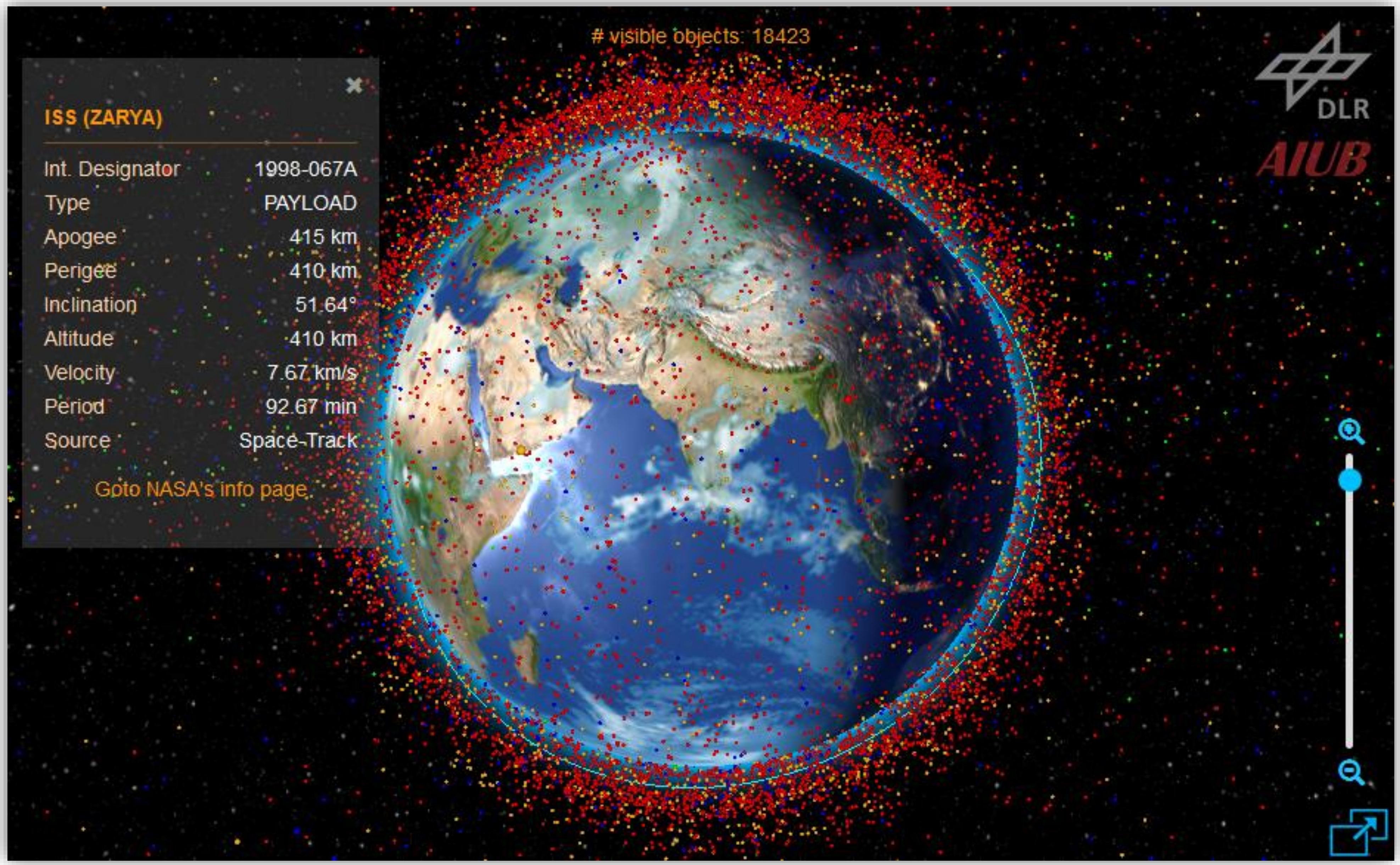
Simulation Software as Research Software

Risk-driven Certification in Simulated Environments

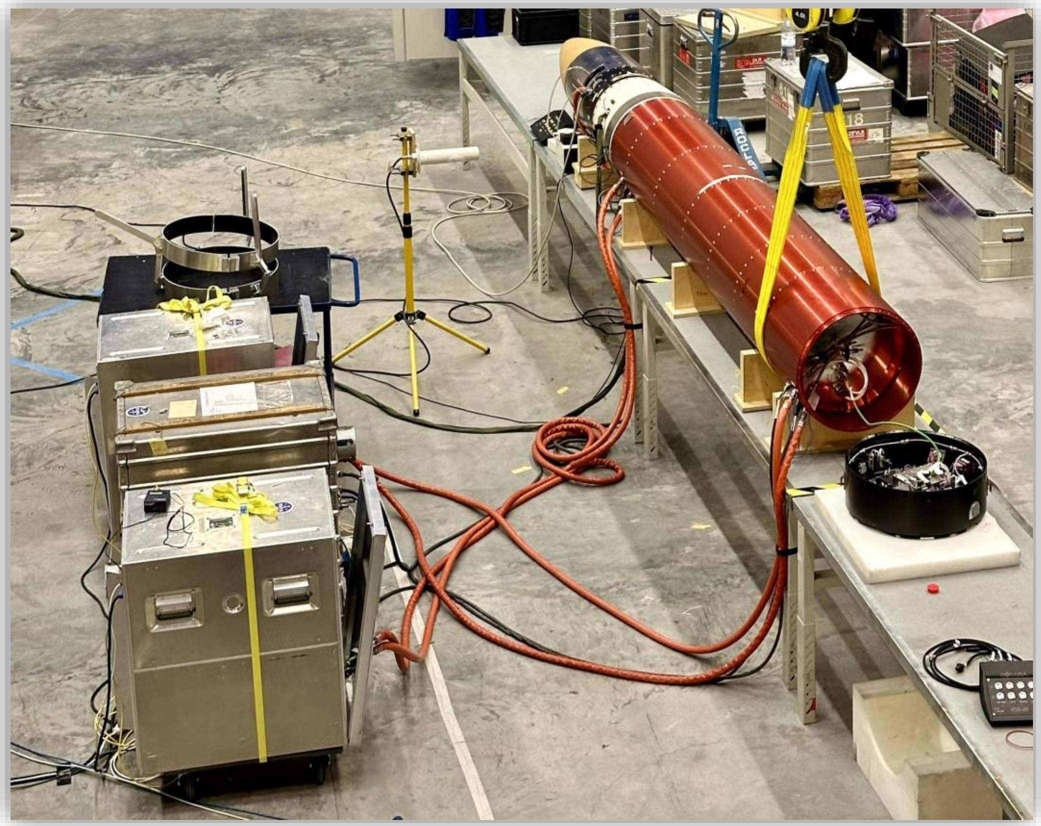
Modeling and Simulation Software (1/2)

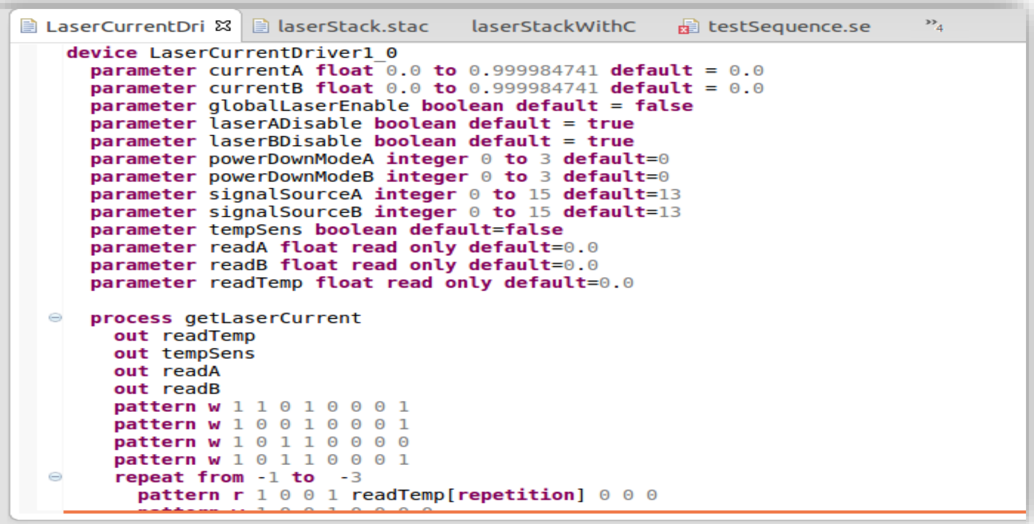


Modeling and Simulation Software (2/2)



Embedded Control Software

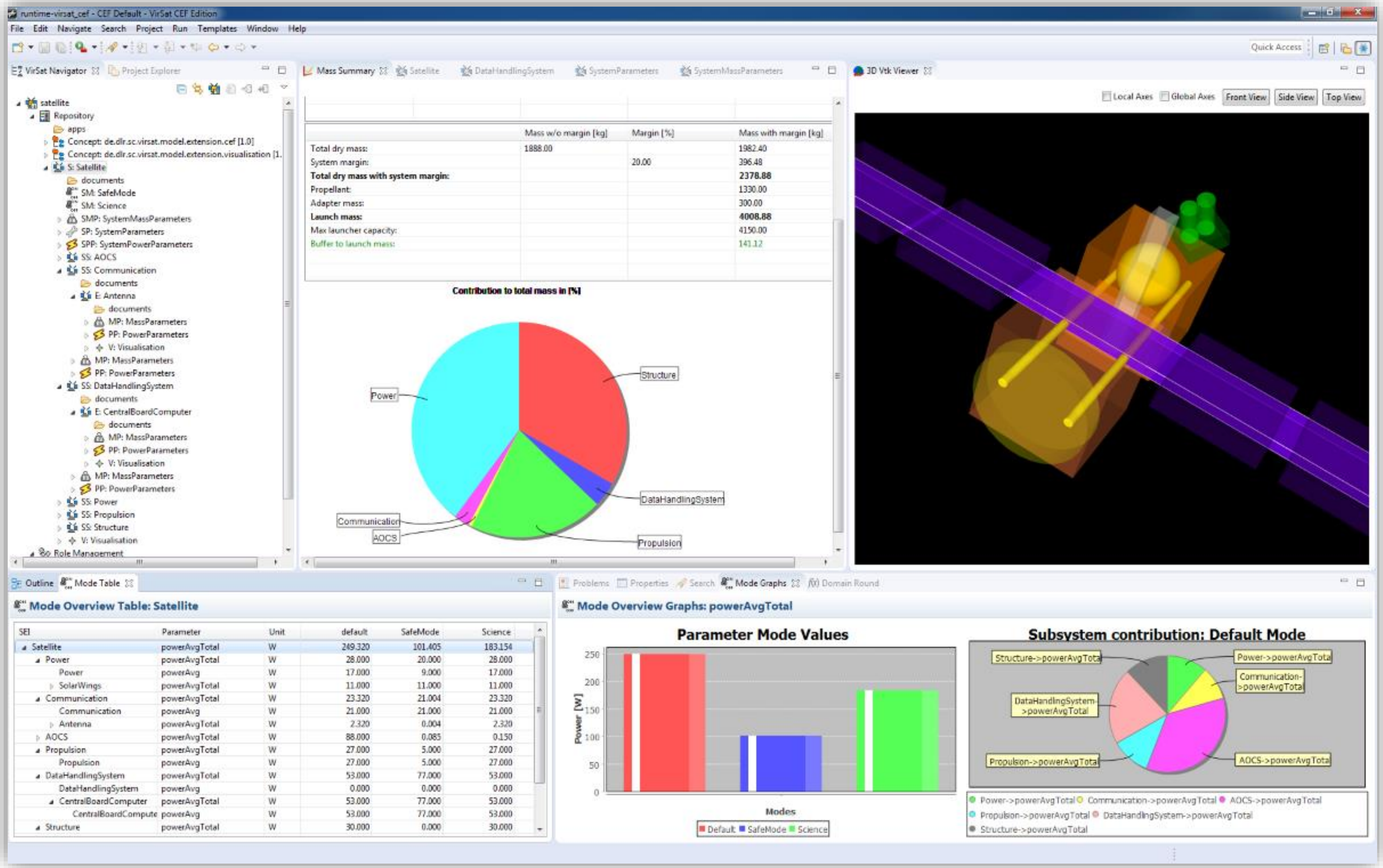




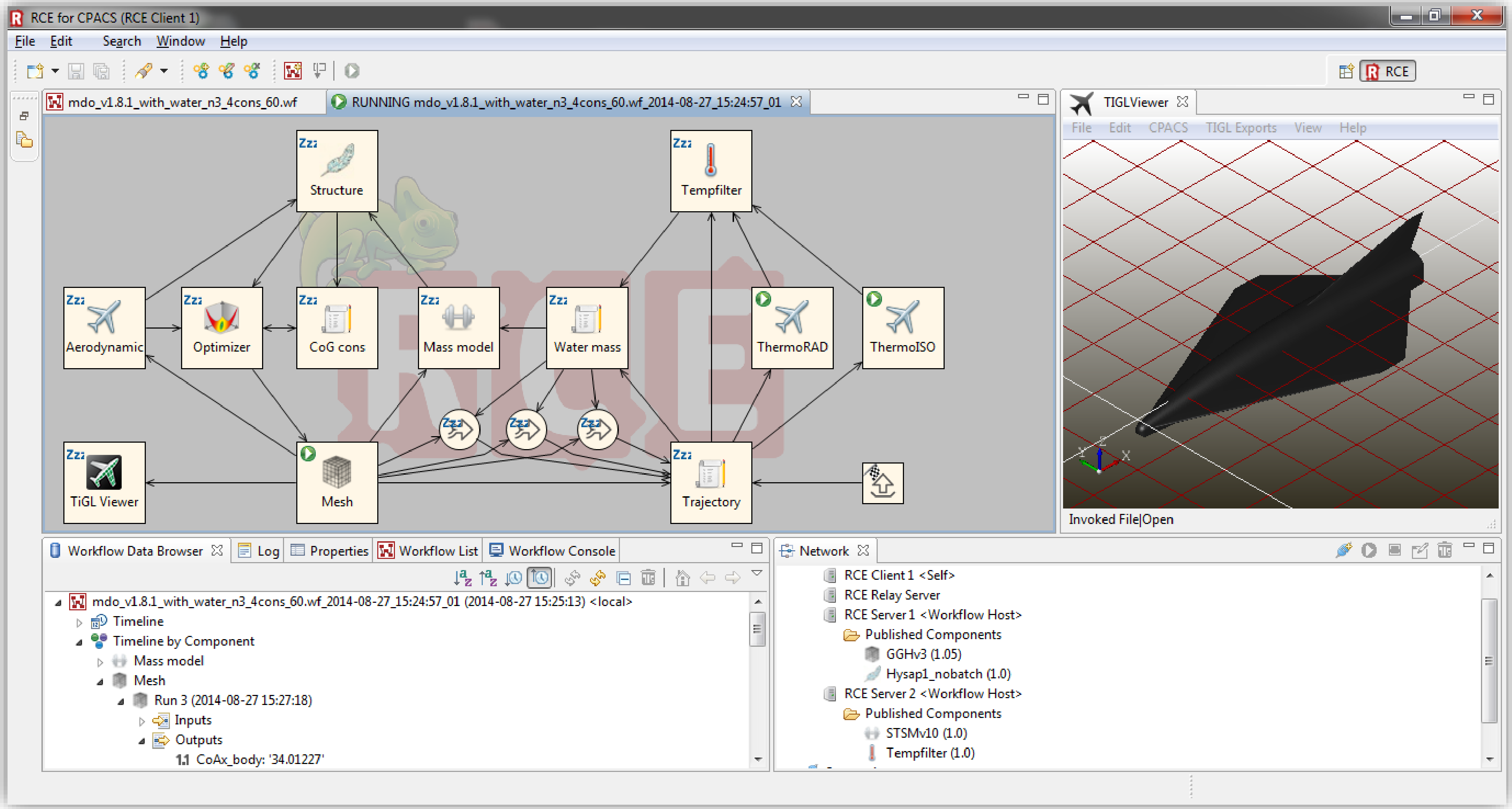

```
device LaserCurrentDriver1_0
  parameter currentA float 0.0 to 0.999984741 default = 0.0
  parameter currentB float 0.0 to 0.999984741 default = 0.0
  parameter globalLaserEnable boolean default = false
  parameter laserADisable boolean default = true
  parameter laserBDisable boolean default = true
  parameter powerDownModeA integer 0 to 3 default=0
  parameter powerDownModeB integer 0 to 3 default=0
  parameter signalSourceA integer 0 to 15 default=13
  parameter signalSourceB integer 0 to 15 default=13
  parameter tempSens boolean default=false
  parameter readA float read only default=0.0
  parameter readB float read only default=0.0
  parameter readTemp float read only default=0.0

  process getLaserCurrent
    out readTemp
    out tempSens
    out readA
    out readB
    pattern w 1 1 0 1 0 0 0 1
    pattern w 1 0 0 1 0 0 0 1
    pattern w 1 0 1 1 0 0 0 0
    pattern w 1 0 1 1 0 0 0 1
    repeat from -1 to  -3
      pattern r 1 0 0 1 readTemp[repetition] 0 0 0
```

Software Prototypes in Engineering Research



Infrastructure and Platform Software



Empirical Investigation on Research Software

DLR

Software Development at the German Aerospace Center: Role and Status in Practice

Lynn von Kurnatowski
German Aerospace Center (DLR)
Oberpfaffenhofen, Germany
lynn.kurnatowski@dlr.de

Tobias Schlauch
German Aerospace Center (DLR)
Braunschweig, Germany
tobias.schlauch@dlr.de

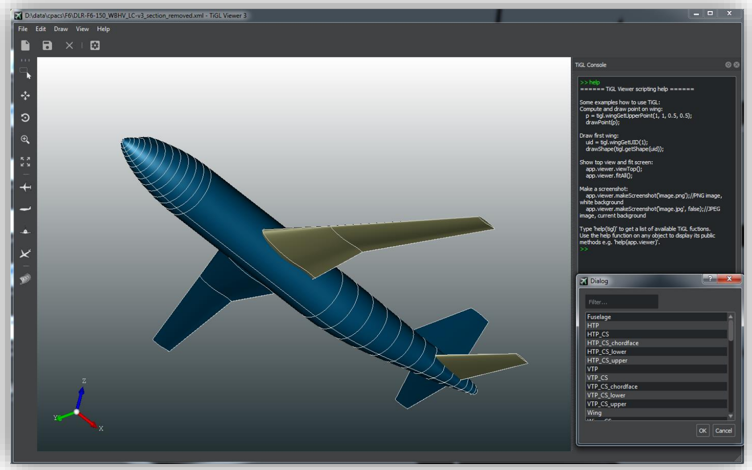
Carina Haupt
German Aerospace Center (DLR)
Berlin, Germany
carina.haupt@dlr.de

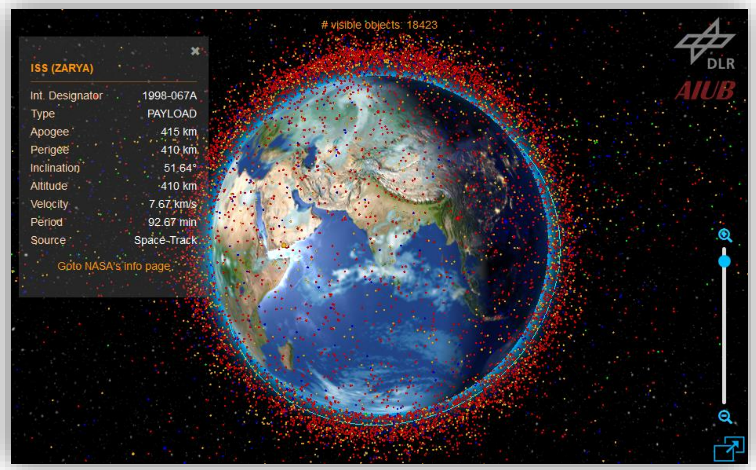
The diversity of research focuses is also reflected in the programming languages that are used. The most frequently used programming language is Python with about 23%, followed by C++ with about 14%, MATLAB with about 12% and C with about 11%.

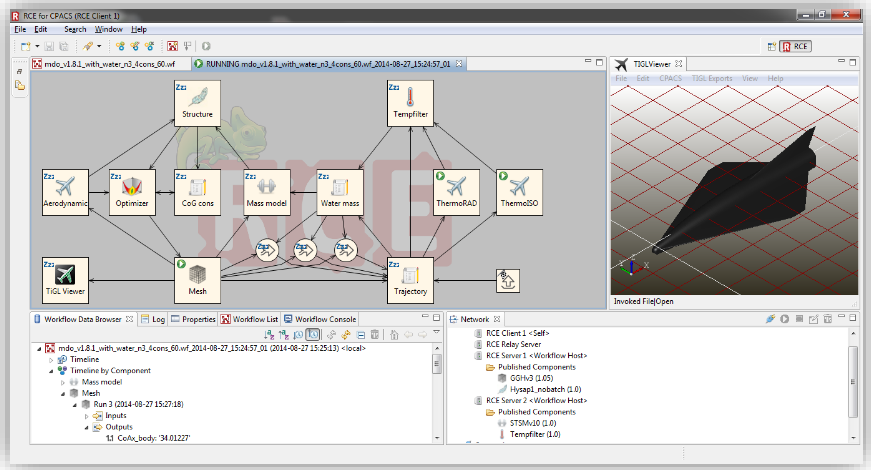
Year: 2018
N: 773

Research Software and Simulation

Research software
(and in particular simulation software)
is a critical artifact that requires
software engineering







Outline

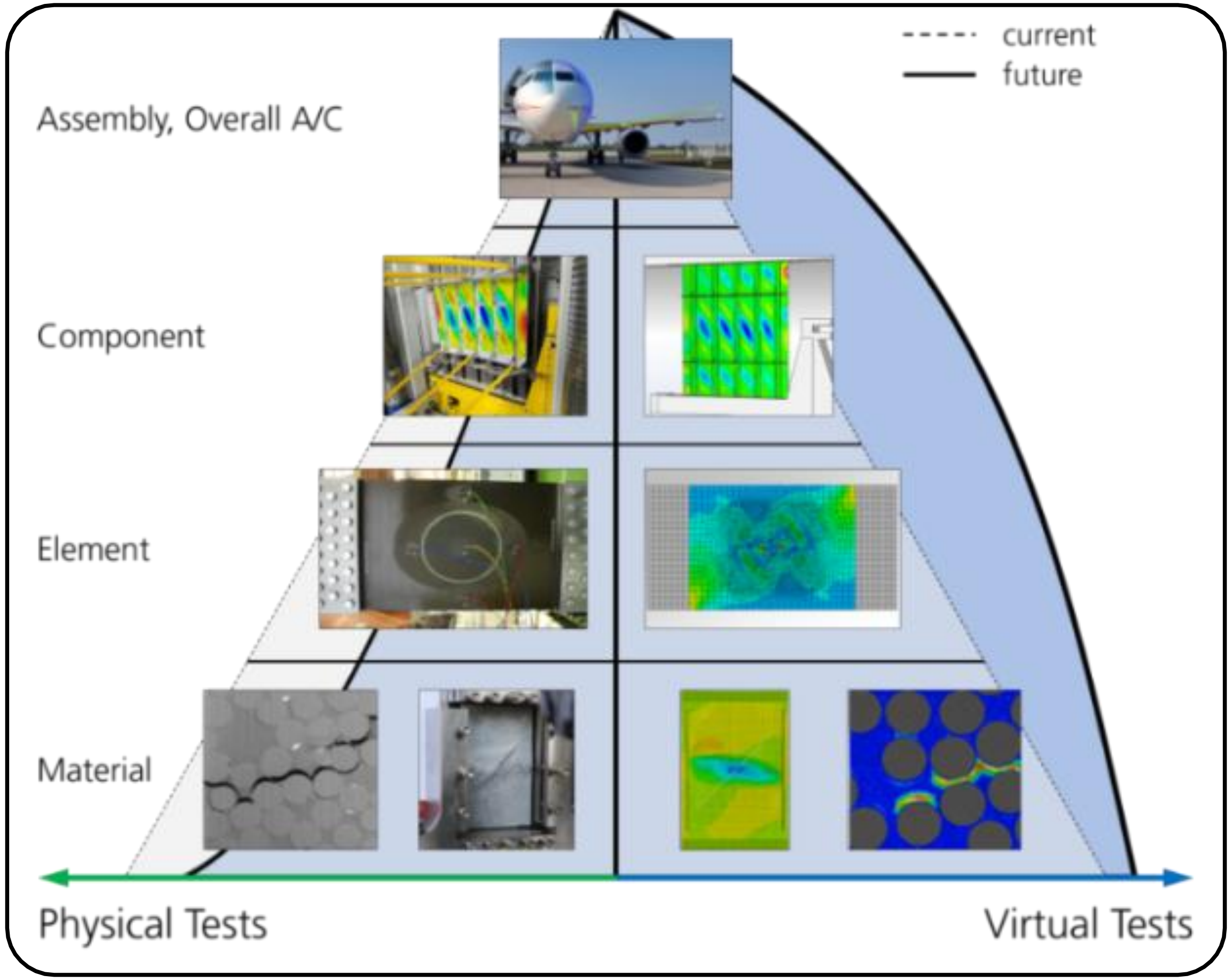
Testing Collaborative AI Systems in Simulated Environments

Simulation Software as Research Software

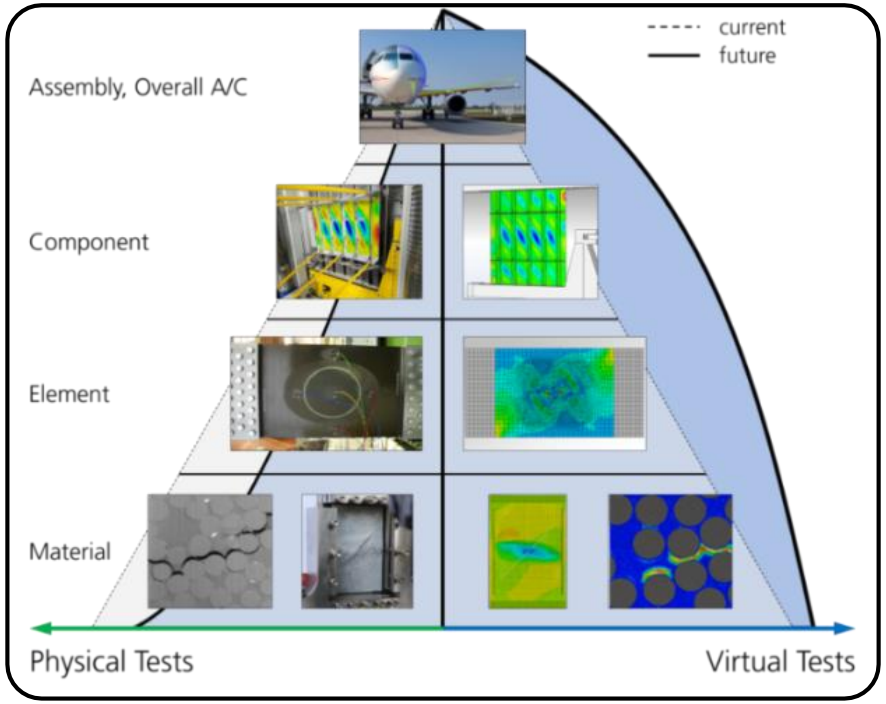
Risk-driven Certification in Simulated Environments

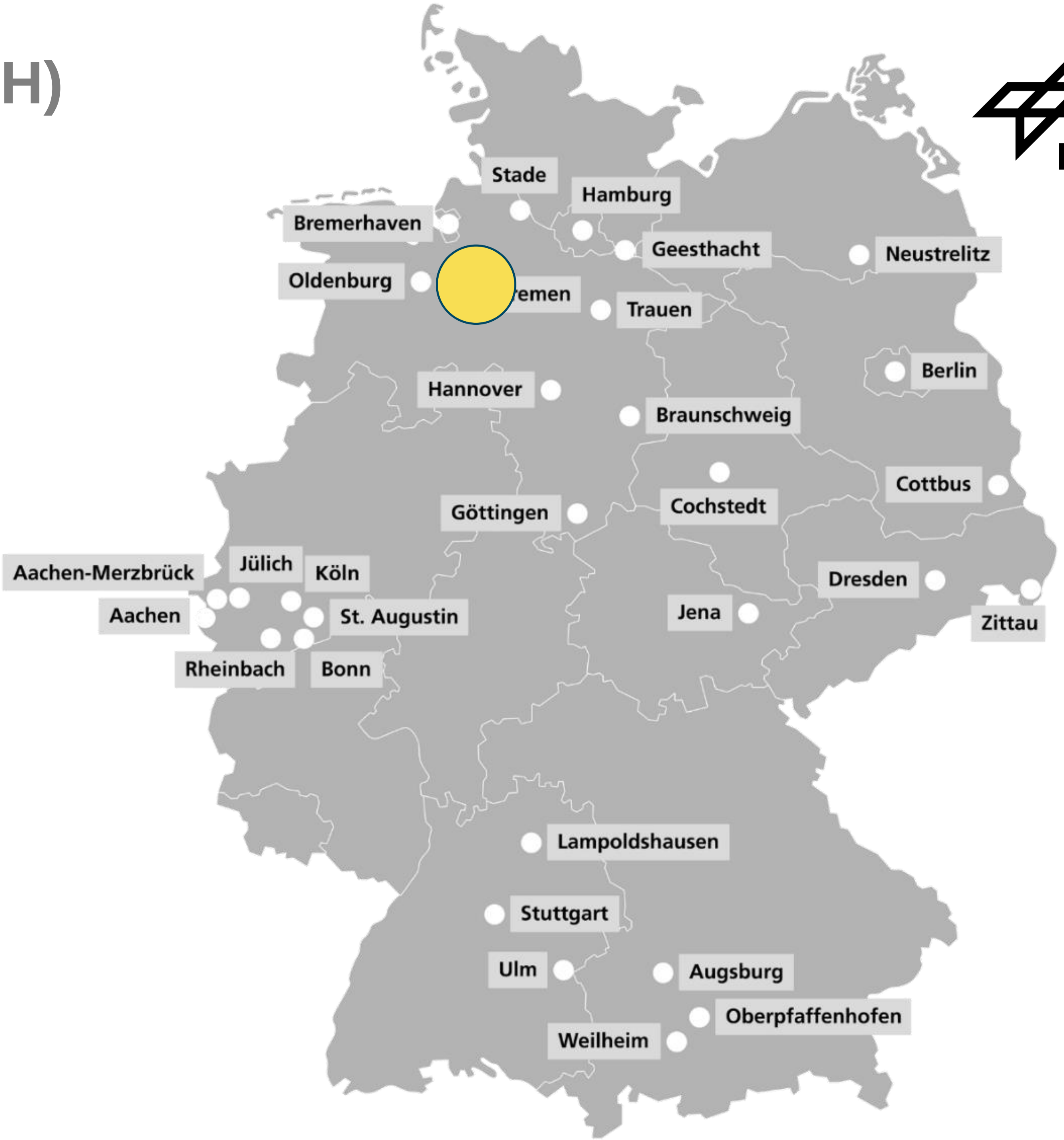
Virtual Product House (VPH): Overview

- Multidisciplinary DLR research collaboration
  - Aerodynamics
  - Aeroelastics
  - Software
  - Structure
  - Systems
- Objectives
  - Virtual Aircraft Development & Evaluation
  - Reduce physical tests
  - Improvements in aircraft emissions
  - Virtual Certification

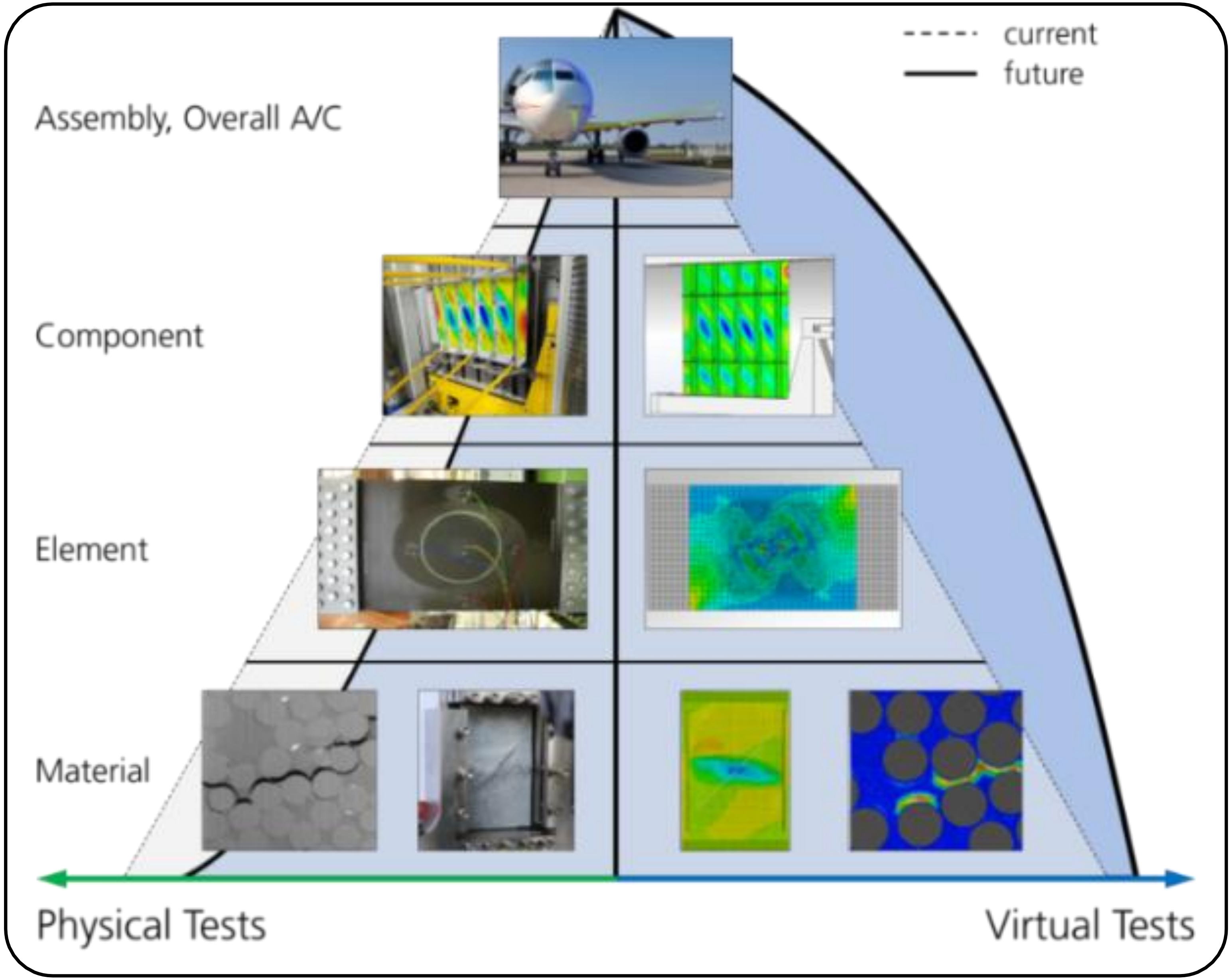


Virtual Product House (VPH)





Simulations for Virtual Tests available



Requirements in Aviation
Source: CS 25, European Union Aviation Safety Agency (EASA)

„The aeroplane […] must be designed […] so that […]

- Any catastrophic failure condition is **extremely improbable**; and does not result from a single failure; and
- Any hazardous failure condition is **extremely remote**; and […]

For each catastrophic failure condition that results from two failures […] it must be shown that […]

- The **sum of the probabilities** […] does not exceed 1/1000“

Requirement: Minimize risk of failure

Requirements in Aviation
Status Quo

Existing Process
- Design and build prototype
- Test on purpose-built test rig
- Measure effects of failures
- Calculate risk of failure conditions

Pro
- Accepted by community
- Accepted by authorities
- Decades of experience

Cons
- Expensive in money and time
- Long feedback cycles

Source: https://en.igh.de/slat-flap-test-rig

Future Vision
Fully risk-driven certification

Current

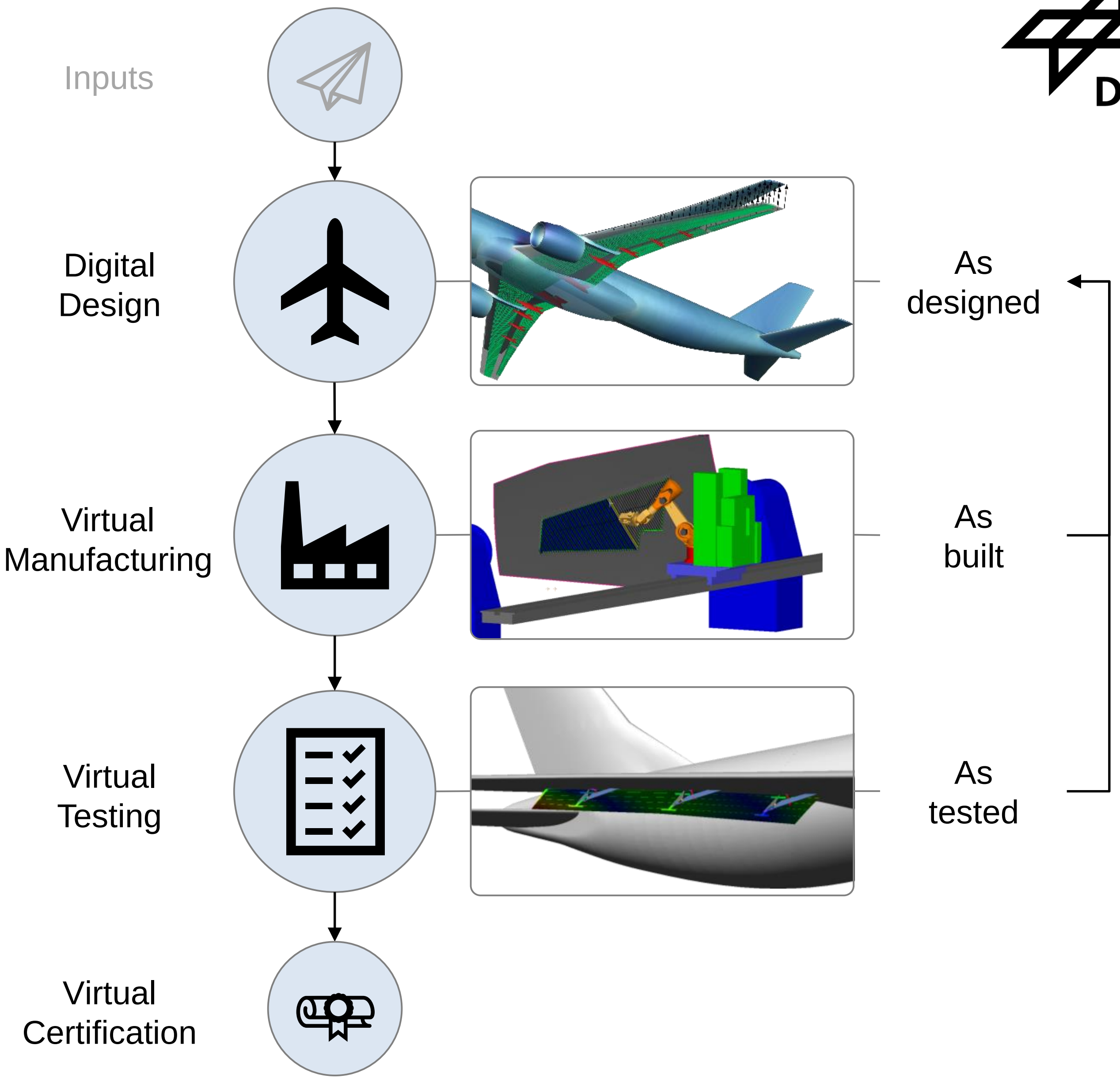
Future

Now: Conservative Airplane Design driven by Top-Down Waterfall Process
Future: Risk-Driven Agile Airplane Design

Virtual Product House

Contribution to Aircraft Lifecycle

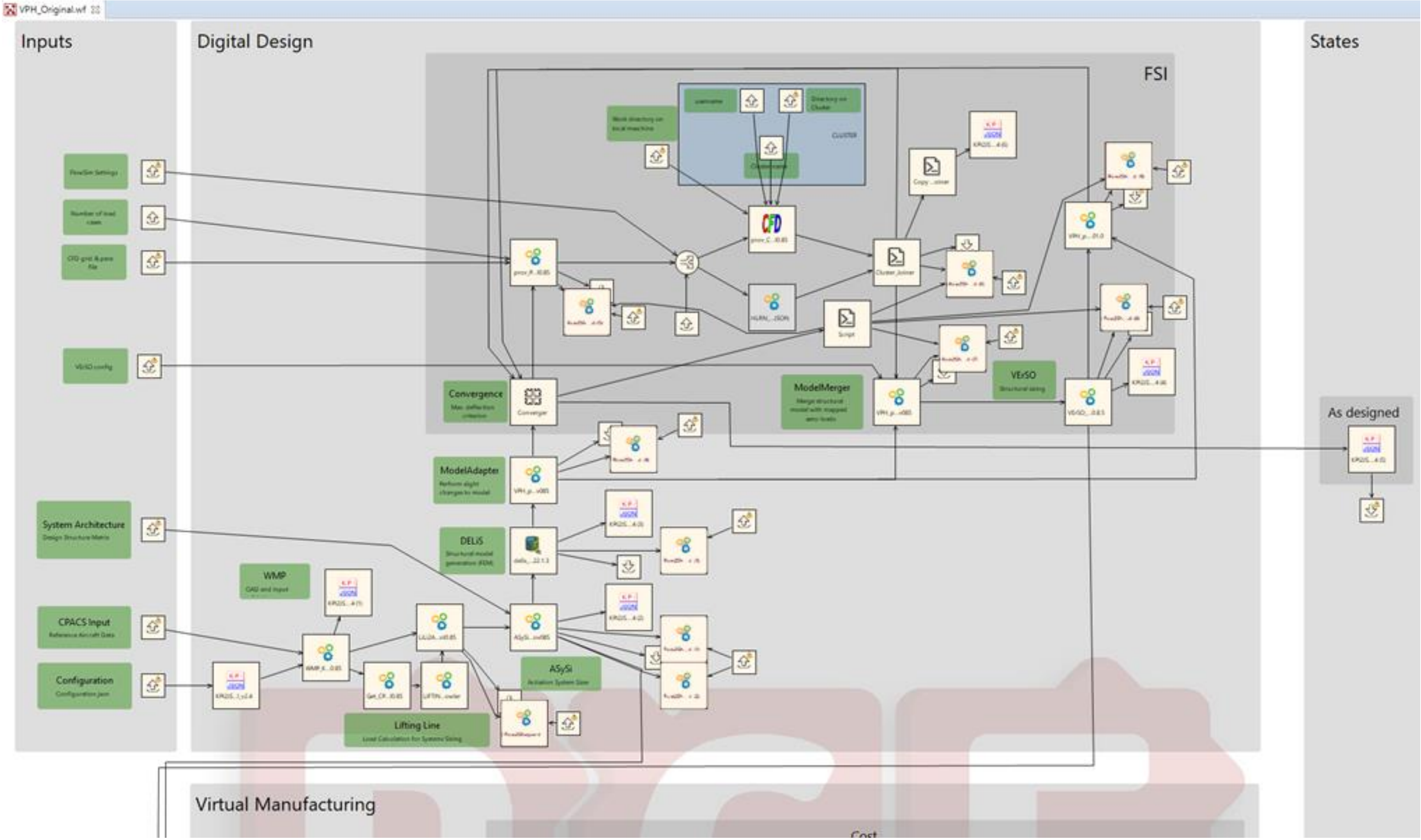
- Phases considered
  - Digital Design
  - Virtual Manufacturing
  - Virtual Testing
  - Virtual Certification
- Research topics
  - Simulation and validation
  - Virtual certification
  - Uncertainty quantification and robustness



Virtual Product House

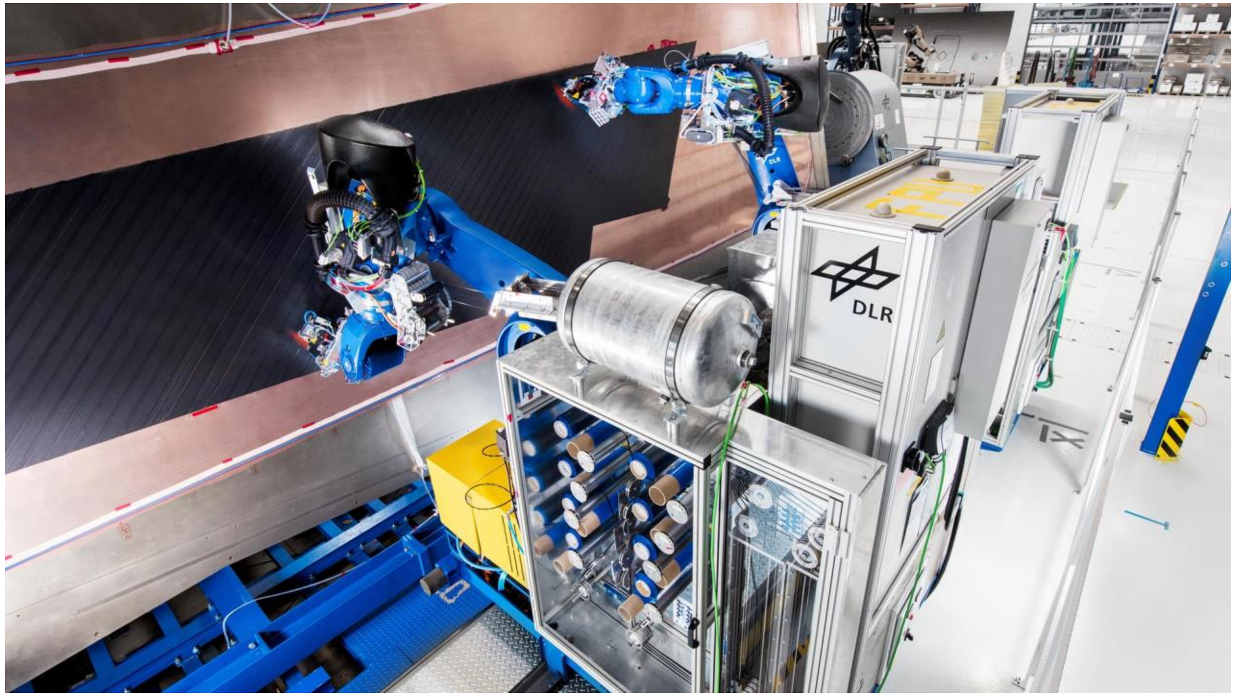
Virtual Design

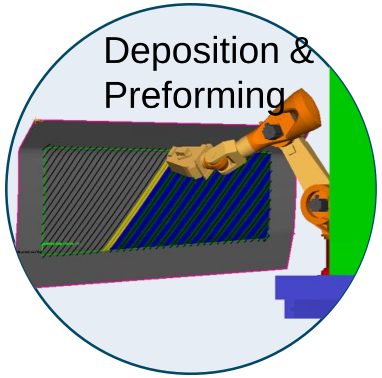
- Digital aircraft (component) design process
- Input: Initial design of aircraft component
- Output: sized component structure ("As designed")

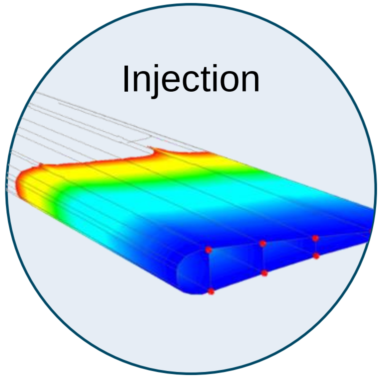


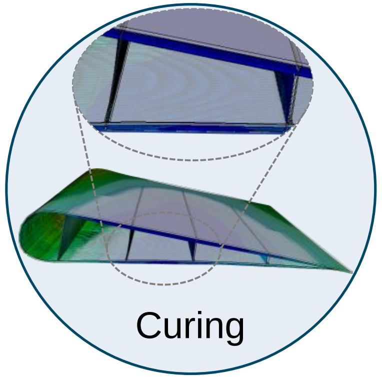
Virtual Product House
Virtual Manufacturing

- Gives a physical model at state “manufactured”
- Enables considering manufacturing related deformations
- Strength distribution for virtual tests

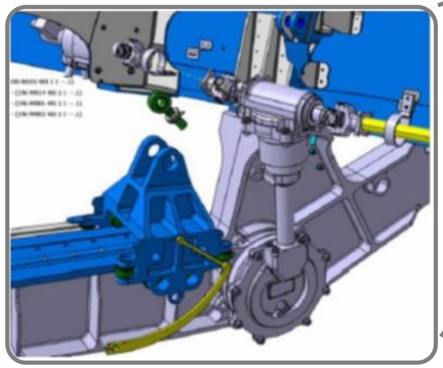


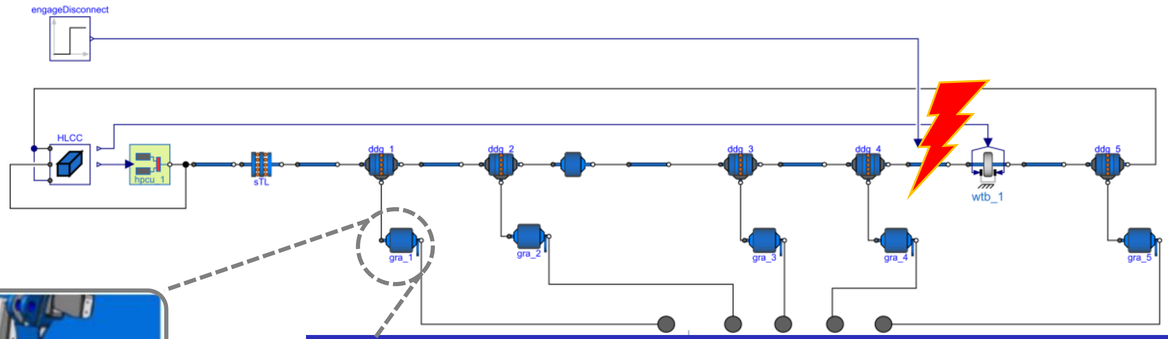


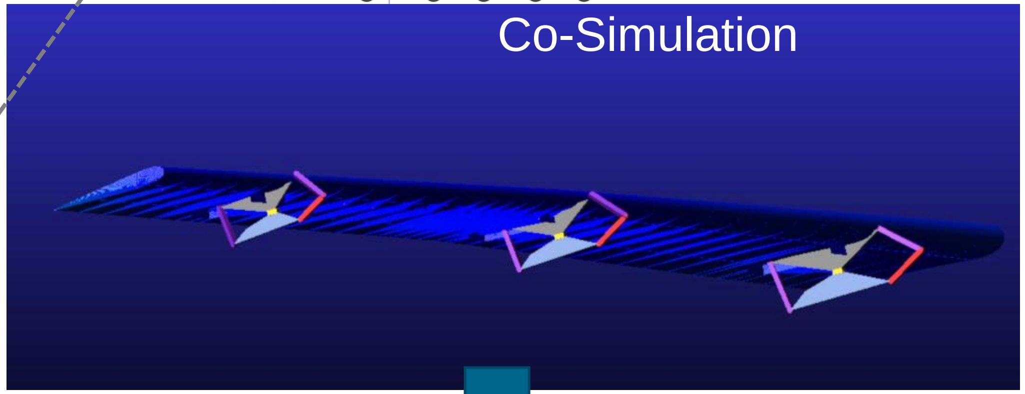


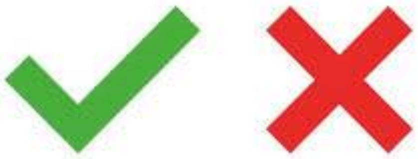


Virtual Product House
Virtual Testing



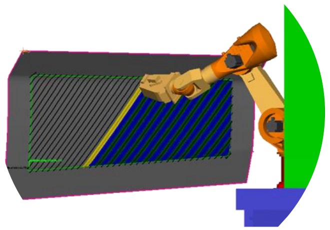


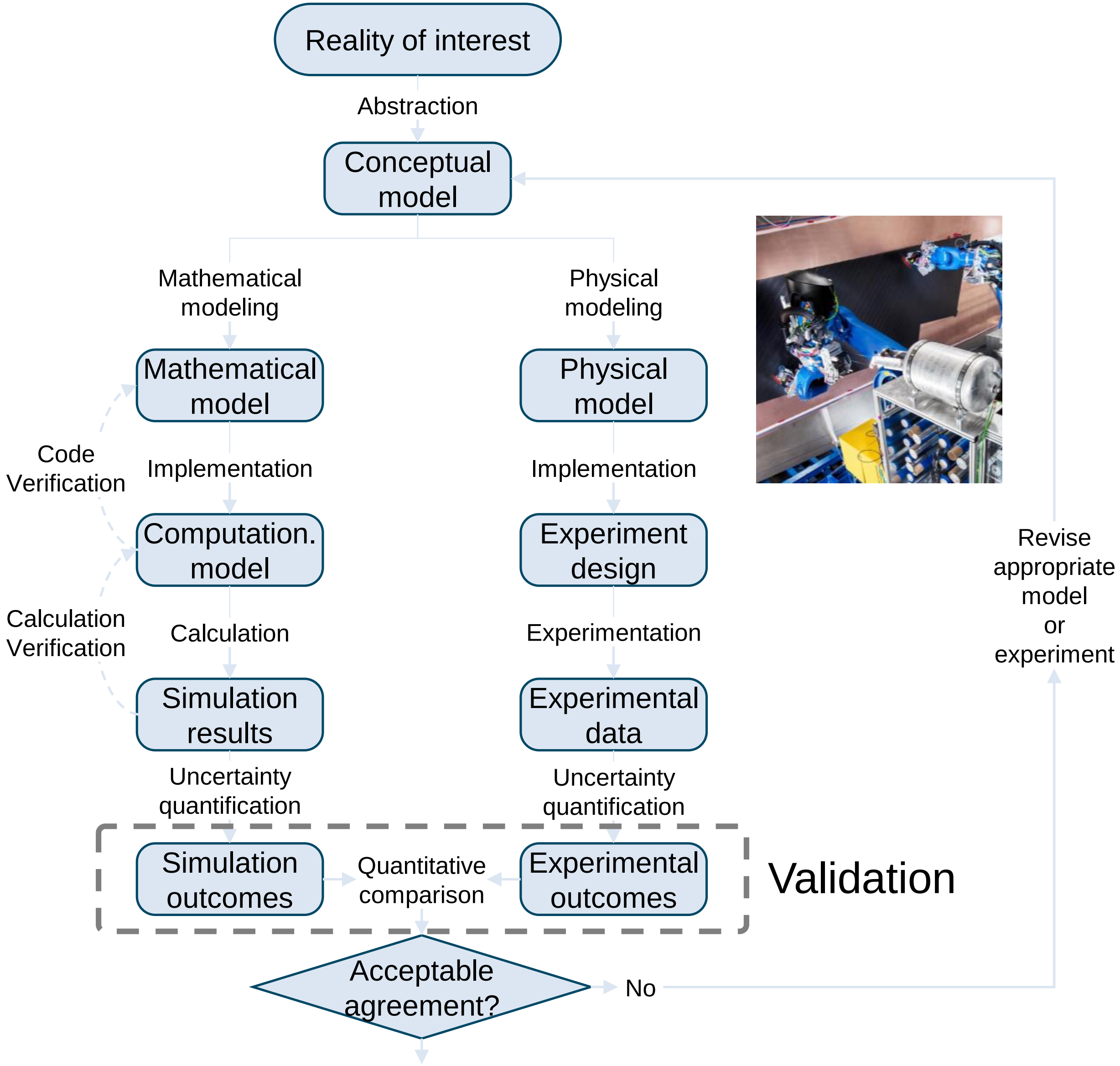




Virtual Produce House

Validation





Virtual Product House
Results so far

Results so far

- **Automated simulations** of design, manufacturing, testing
- **Validated simulations** by comparing with actual production and testing
- **Fidelity deemed sufficient** via real-world comparison

Benefits

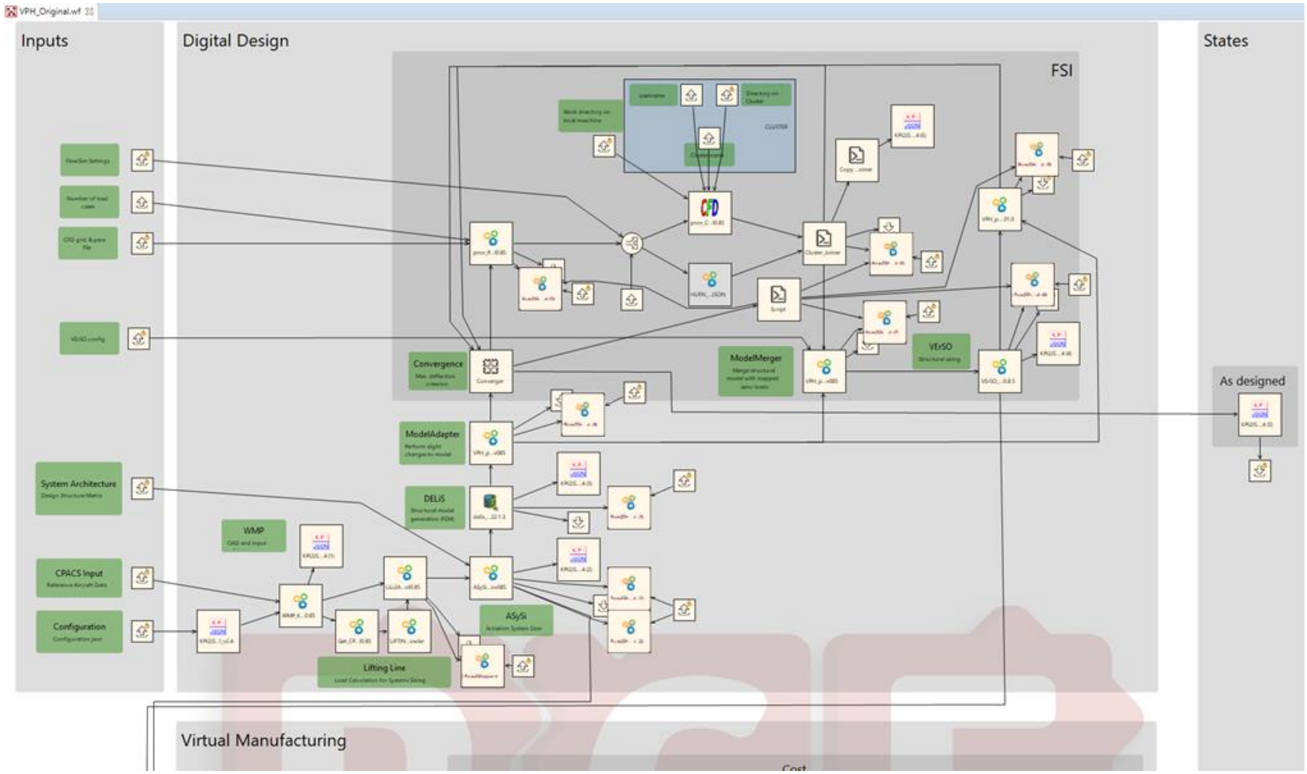
- Fewer prototypes, reduced cost
- Shorter feedback cycles

Future Work and Research

- Development and design process
- Uncertainty quantification
- Resilience
- Validation
- Credibility for authorities

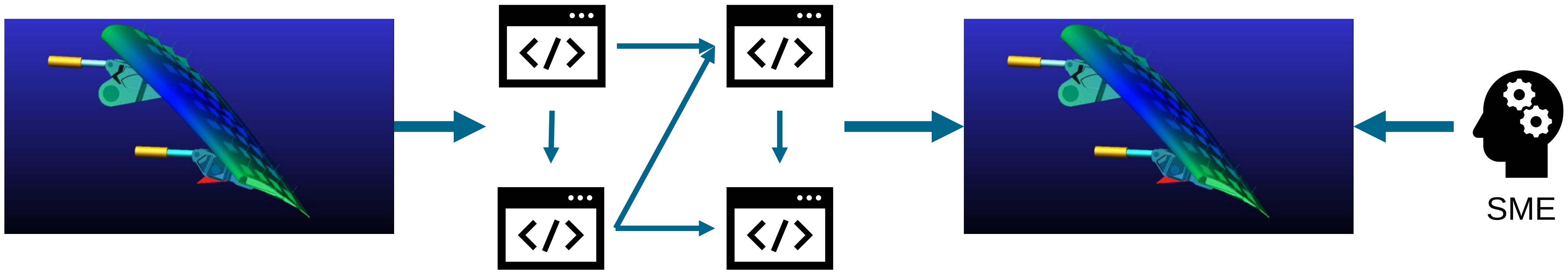
Software Testing at VPH
Current Status

- System under test: Software tool chain
  - ~10 discipline-specific tools
  - pre- and postprocessing for each tool
- Quality assurance for whole system via ad-hoc, manual testing



Software Testing at VPH
Requirements

- No codified, testable requirements
  - Input: Wing model
  - Output: Modified wing model
- Acceptance criteria
  - practical knowledge of Subject Matter Expert (SME)
  - similarity to previous results

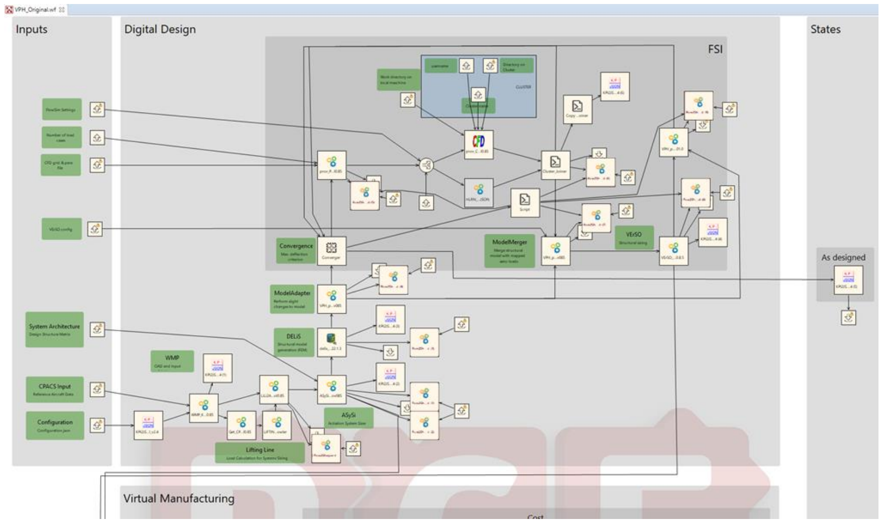


Software Testing at VPH
Future Vision

- Testing goal: Find wing model where output becomes implausible for SME

- Intermediate goal: Construct set of edge case wing models
  - Find reasonable parameter space (length, width, shape, no. of flaps, …) with SME
  - Use SME as binary oracle, perform random search
  - Use SME as gradient oracle, let feedback guide search
  - Use SME to provide training data for AI-SME

- Major issues:
  - Single execution currently takes long
  - SME feedback not necessarily consistent



Research directions

Uncertainty Quantification

- Estimate the error range, handle uncertain data, and quantify fidelity of the simulation, application of AI to increase fidelity

Resilience

- Apply simulation to unvalidated input parameters and the whole range within the validated design space

Validation

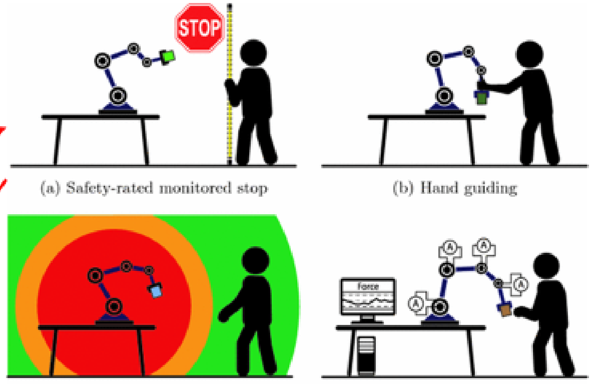
- Correctness of the simulation and behavior of the digital twin in relation to the real world phenomenon

Credibility for Authorities

- Trustworthiness of the whole process

Industrial Robot Safety

ISO 10218 and **ISO/TS 15066** which specify risk management processes for robots and robotic devices and safety requirements for industrial robots and collaborative industrial robots define **four collaborative operating modes**

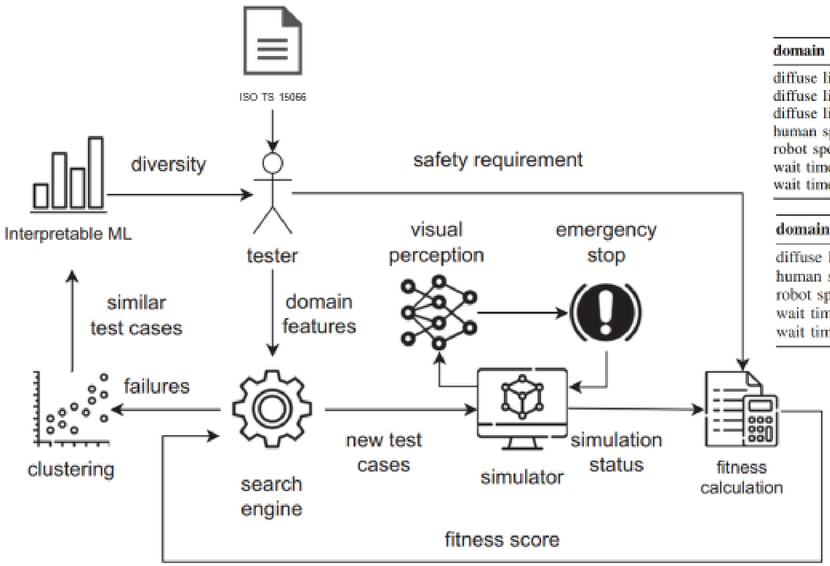


(a) Safety-rated monitored stop (b) Hand guiding (c) Speed and separation monitoring (d) Power and force limiting

Our work is based on the **Safety-rated monitored stop** operating mode

Risk assessment to deal with ML and related risks in CAIS not considered in current standards like ISO 10218 or ISO/TS 15066

Risk-driven Online Testing

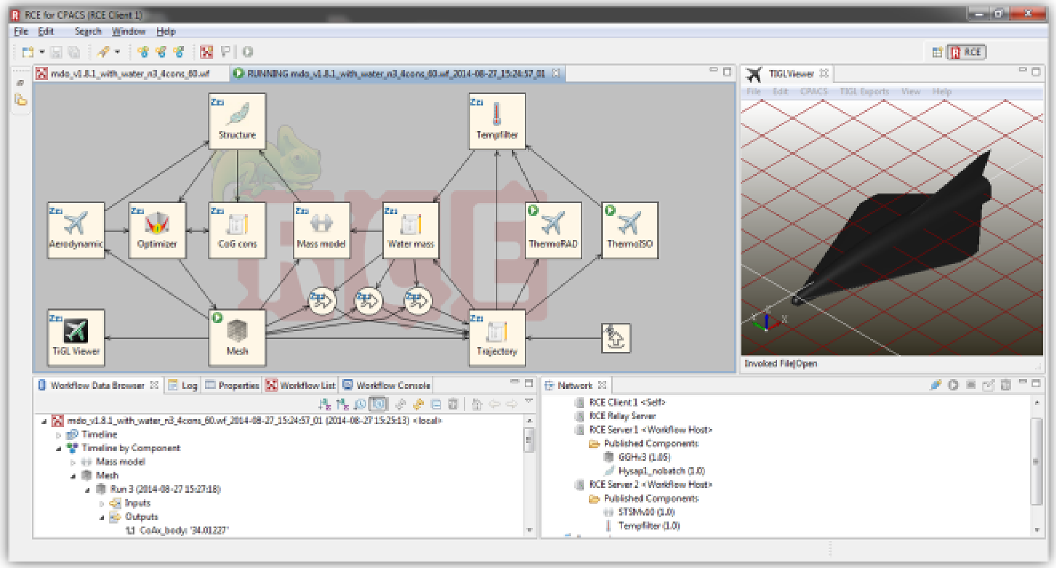


domain feature	type	lower bound	upper bound
diffuse light (R)	float	0.0	1.0
diffuse light (G)	float	0.0	1.0
diffuse light (B)	float	0.0	1.0
human speed (m/s)	float	0.1	0.5
robot speed (m/s)	float	0.05	0.5
wait time human (s)	integer	1	50
wait time robot (s)	integer	1	50

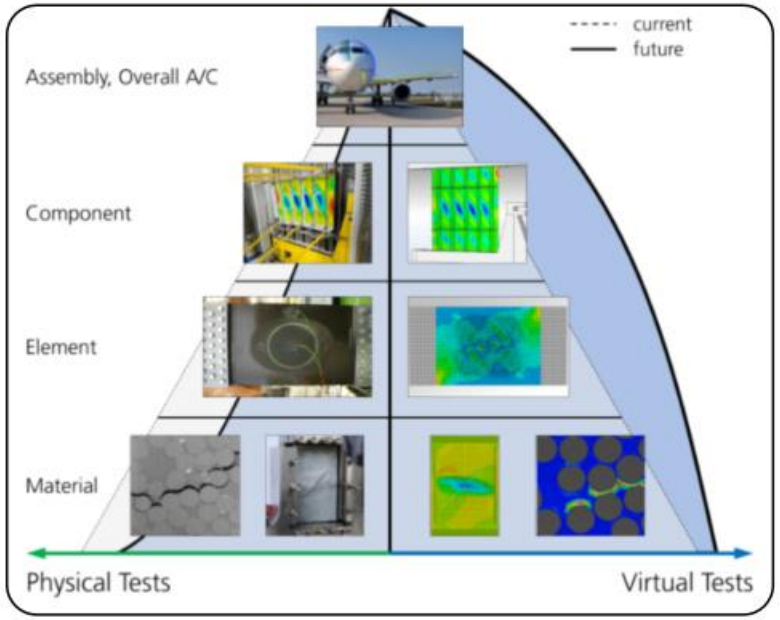
domain feature	value
diffuse light 1	(0.1, 0.2, 0.3) RGB
human speed	0.33 m/s
robot speed	0.25 m/s
wait time human	2 s
wait time robot	5 s

Objective Functions:
Minimum distance between human and robot arm
Relative speed of human and robot arm

Infrastructure and Platform Software



Simulations for Virtual Tests available



References

[1] Adigun, J., Camilli, M., Felderer, M., Giusti, A., Matt, D., Perini, A., Russo, B., Susi, A. (2022) Collaborative AI Needs Stronger Assurances Driven by Risks. Computer, 55(3), IEEE

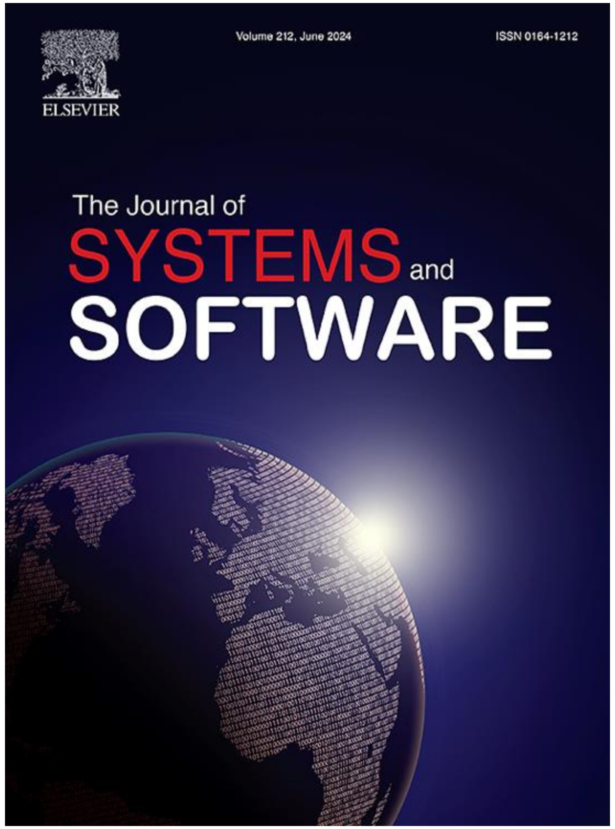
[2] Adigun, J., Huck, T., Camilli, M., Felderer, M. (2023) Risk-driven Online Testing and Test Case Diversity Analysis for ML-enabled Critical Systems. ISSRE 2023, IEEE

[3] Felderer, M., Schieferdecker, I. (2014) A taxonomy of risk-based testing. International Journal on Software Tools for Technology Transfer, 16(5), Springer

[4] Kurnatowski, L., Schlauch, T., Haupt, C. (2020) Software Development at the German Aerospace Center: Role and Status in Practice. ICSE (Workshops) 2020

[5] Mischke, R., Schaffert, K., Schneider, D., Weinert, A. (2022) Automated and Manual Testing in the Development of the Research Software RCE. ICCS 2022, Springer

Call for Papers

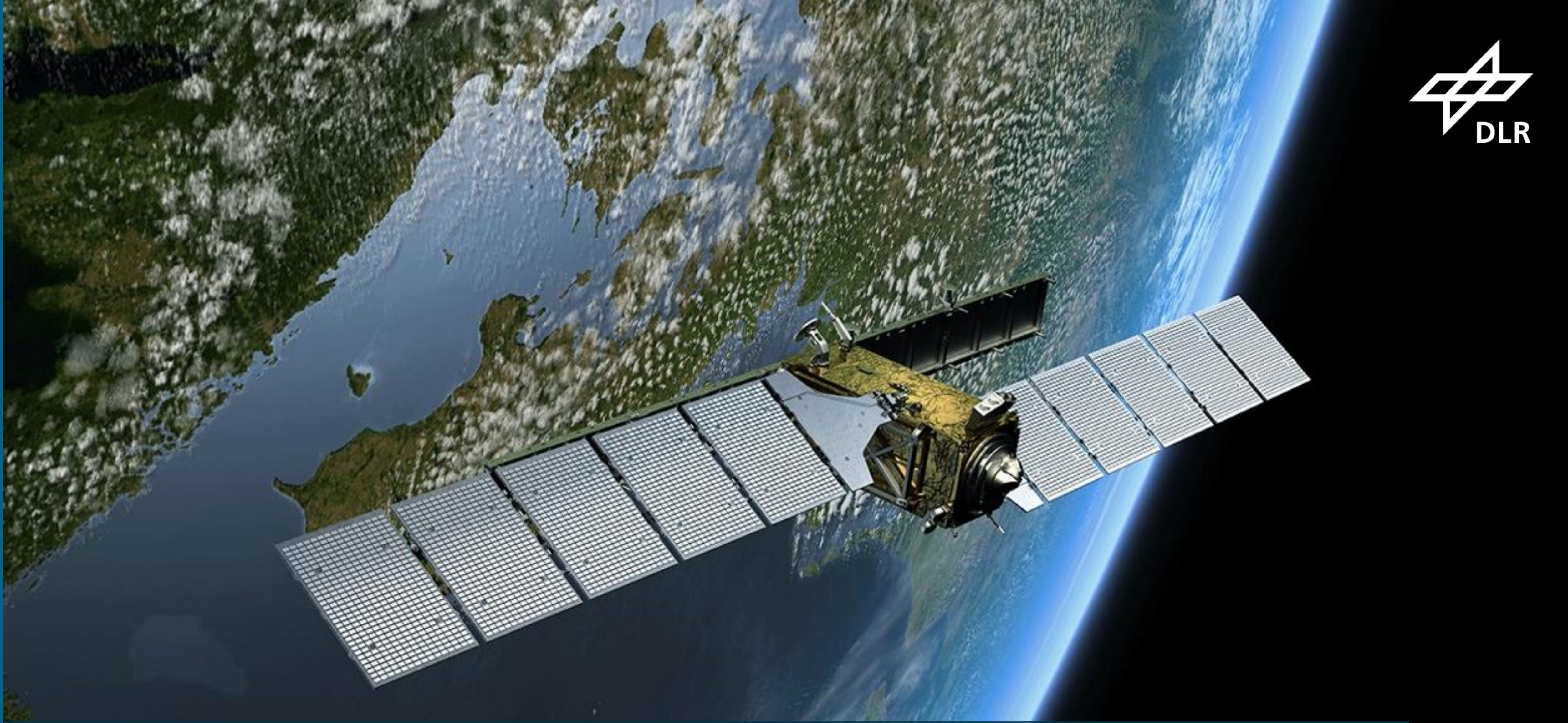


Special Issue on:
Automated Testing and Analysis for Dependable AI-enabled Software and Systems

Guest editors

Matteo Camilli, Politecnico di Milano, Italy
Michael Felderer, German Aerospace Center (DLR) and University of Cologne, Cologne, Germany
Alessandro Marchetto, University of Trento, Italy
Andrea Stocco, Technical University of Munich (TUM) and fortiss GmbH, Germany

Submission Deadline: August 31, 2024



Prof. Dr. Michael Felderer
Institute of Software Technology

michael.felderer@dlr.de
https://www.dlr.de/sc/