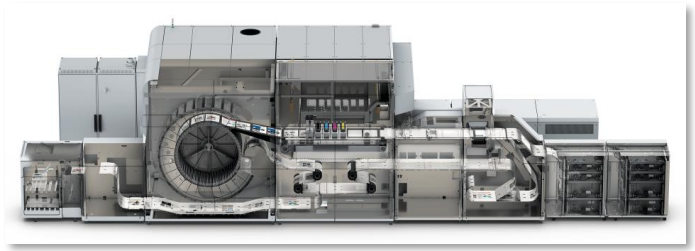
Synthesis-Based Engineering (SBE) of Supervisory Controllers

Dennis Hendriks | dennis.hendriks@tno.nl + dennis.hendriks@ru.nl

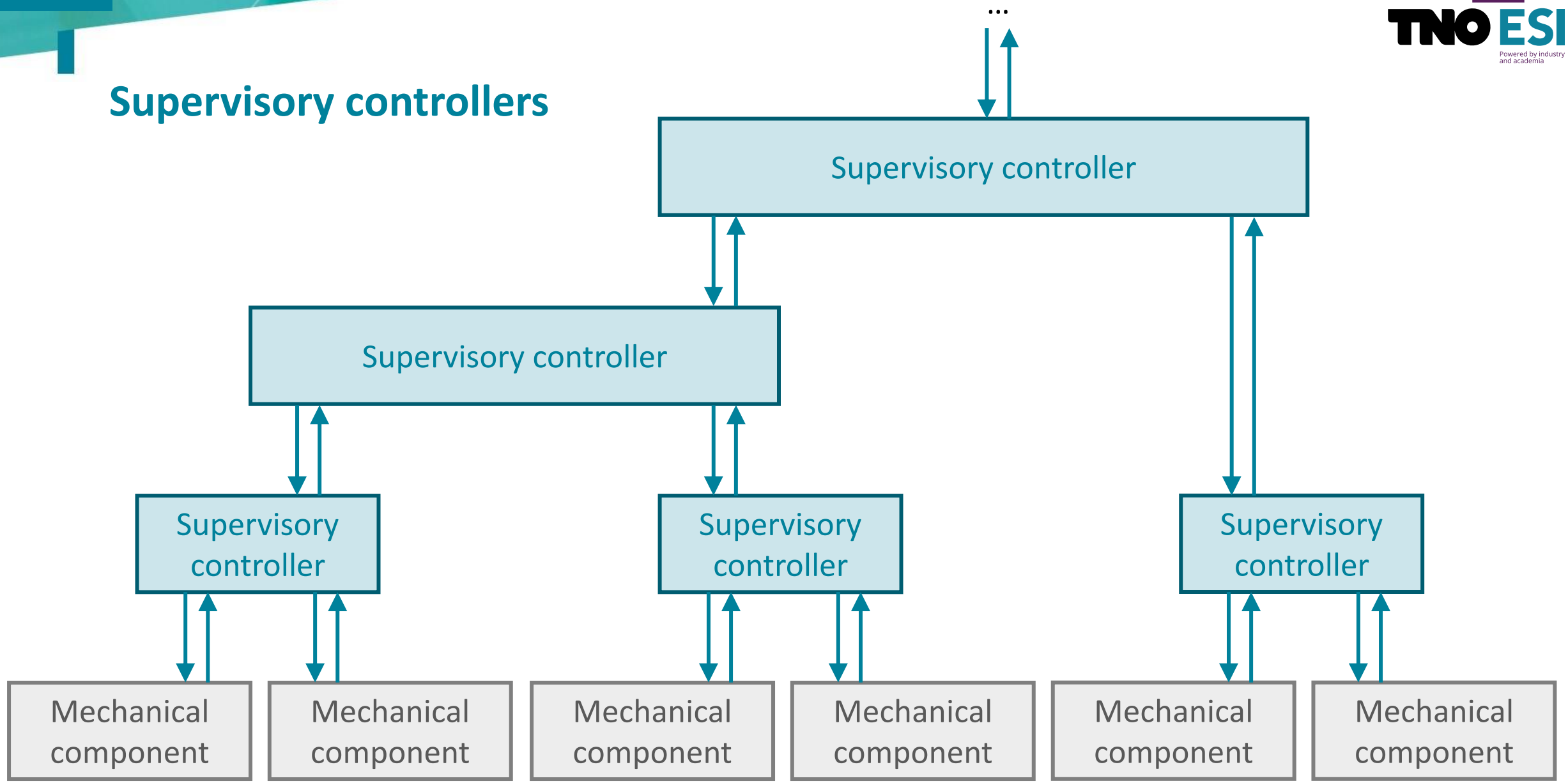
ESI symposium | October 7, 2025

TNO ESI
Powered by industry and academia

Cyber-physical systems



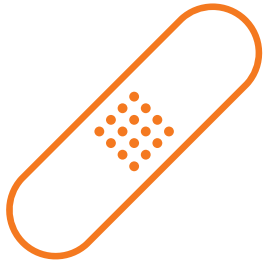
Supervisory controllers



Challenges of engineering supervisory controllers

(1)
Dealing with complexity

E.g., integrating many components and supporting many variants



(2)
Ensuring high quality

E.g., preventing deadlocks in the controller design

(3)
Shortage of skilled people

E.g., being more productive with the people we have

Engineers continuously face difficult trade-offs and arrive at sub-optimal solutions

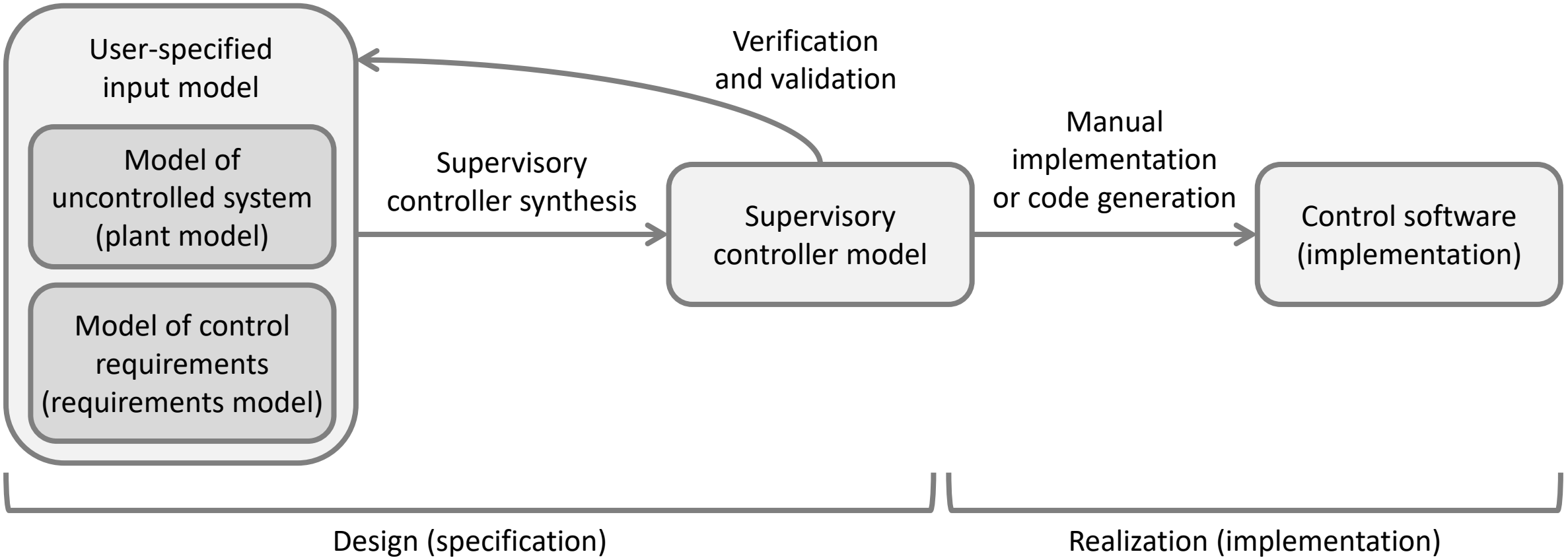
Synthesis-Based Engineering (SBE) of supervisory controllers

Engineering approach → / ↓ Development step	Traditional Engineering	Model-Based Engineering	Verification-Based Engineering	Synthesis-Based Engineering
Requirements design	Document-based	Document-based	**Model-based (formal)**	Model-based (formal)
Controller design	Document-based	**Model-based (formal)**	Model-based (formal)	**Computer-aided (formal, synthesized)**
Realization in software (implementation code)	Traditional software engineering (coding)	**Code generation (fault-free code)**	Code generation (fault-free code)	Code generation (fault-free code)
Verification (against requirements)	Testing	Testing + **Model-based testing**	**Formal verification (model checking)**	**Correct-by-construction (guaranteed)**
Validation (of requirements)	Testing	Testing + **Simulation**	Testing + Simulation	Testing + Simulation

Legend: Manual work / Focus (Semi-)automatic

(Simplified overview.)

The Synthesis-Based Engineering process



(Simplified process.)

Synthesis-Based Engineering helps to manage the high complexity

Better understanding

Reducing the engineering effort

Specify the **what** rather than the **how**

Increased efficiency

Improving time to market

Highly automated: ‘push-button’ approach

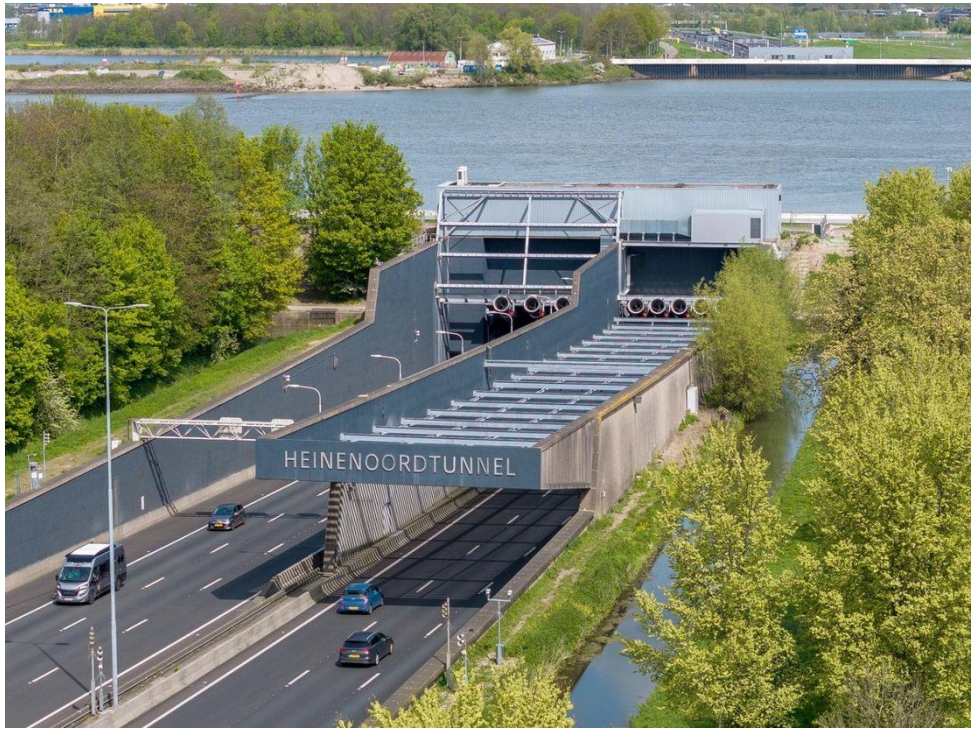
Improved quality

Avoiding costly engineering mistakes

Correctness by construction

Develop trustworthy control software, at reduced effort and cost

Supervisor synthesis example: a tunnel



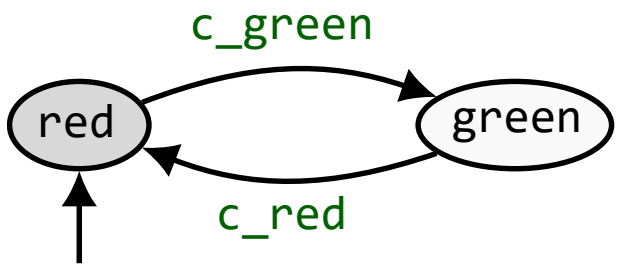
Eerste Heinenoordtunnel, A29, the Netherlands

Source: Rijkswaterstaat

Supervisor synthesis example: a tunnel

Example plant component

```
TrafficLight:
```



Example requirement

Textual description:

"The boom barrier may only be lowered when the traffic light is red."

Formalized in CIF model:

```
requirement BoomBarrier.c_down needs TrafficLight.red;
```

Supervisor controller synthesis

- Number of plant components: **1,163**
- Number of requirements: **2,015**
- Size of controlled system: **7.00×10^{311}** states
- Synthesis time: **< 5 seconds** (regular laptop)

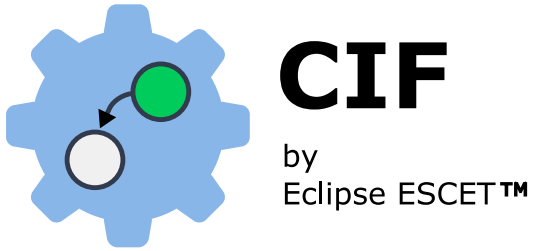
Sources: Rijkswaterstaat, YouTube

Synthesis-Based Engineering tools

Eclipse ESCET project/toolkit (pronounced as èsèt)

- "Eclipse Supervisory Control Engineering Toolkit"
- Open-source project at the Eclipse Foundation
- Collaborative tool development ecosystem (researchers, developers, users)

Visit: https://eclipse.dev/escet



CIF language/tools

- Powerful automata-based modeling language
- Tools to support the entire SBE process

Visit: https://eclipse.dev/escet/cif

Synthesis-Based Engineering community

Prabhat Kumar Sharma

Lars Moormann

Growing community

Community activities

- Academic research
- Applied research
- Joint publications
- Tool development
- Community meetings
- Workshops
- Ecosystem orchestration (roadmap, etc)
- …

Develop trustworthy control software, at reduced effort and cost

Some of the work presented in this presentation is carried out at part of the Poka Yoke program under the responsibility of TNO-ESI in cooperation with ASML and VDL-ETG. The Poka Yoke program is funded by Holland High Tech | TKI HSTM via the PPP Innovation Scheme (PPP-I) for public-private partnerships.

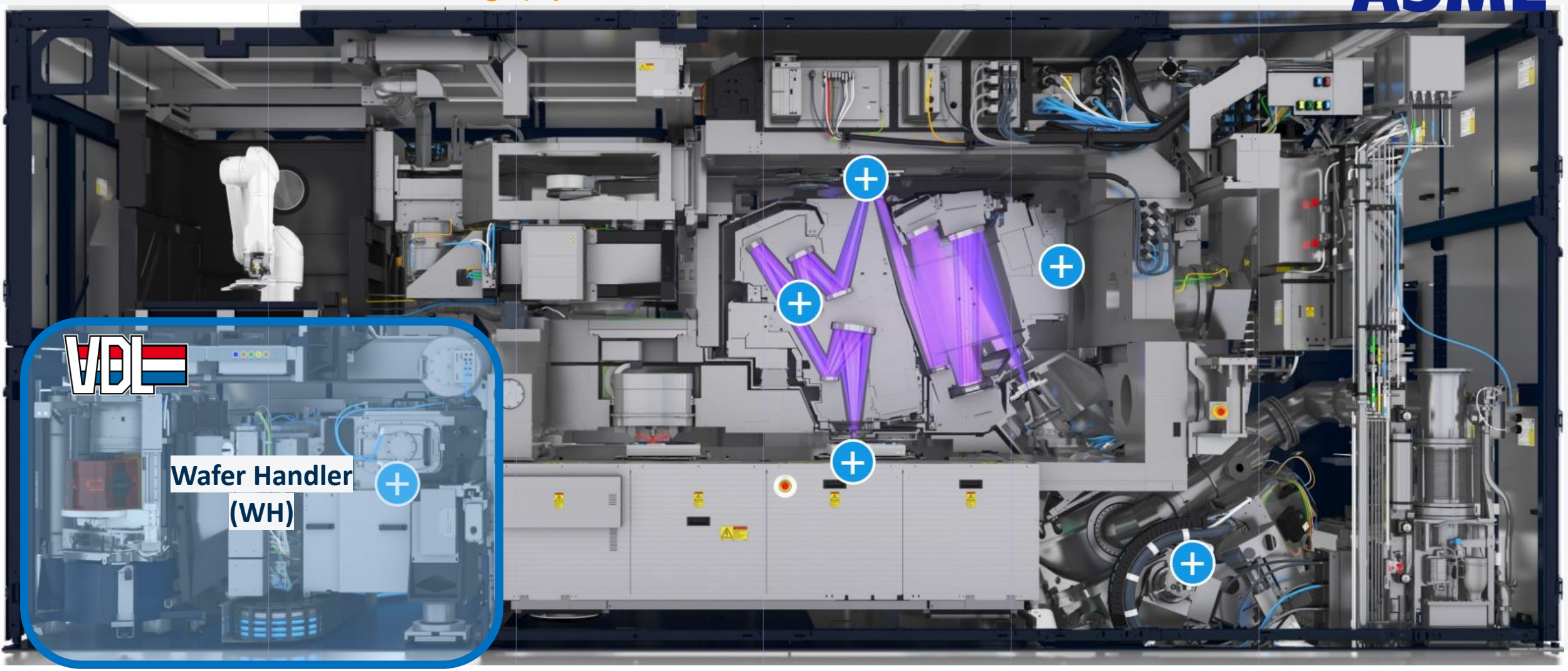
TNO ESI
Powered by industry and academia

Synthesis-Based Engineering (SBE) for supervisory control

Prabhat K Sharma
Mechatronics Design Engineer, VDL ETG T&D
Supervisory Control Technology group
Oct 07, 2025

Context

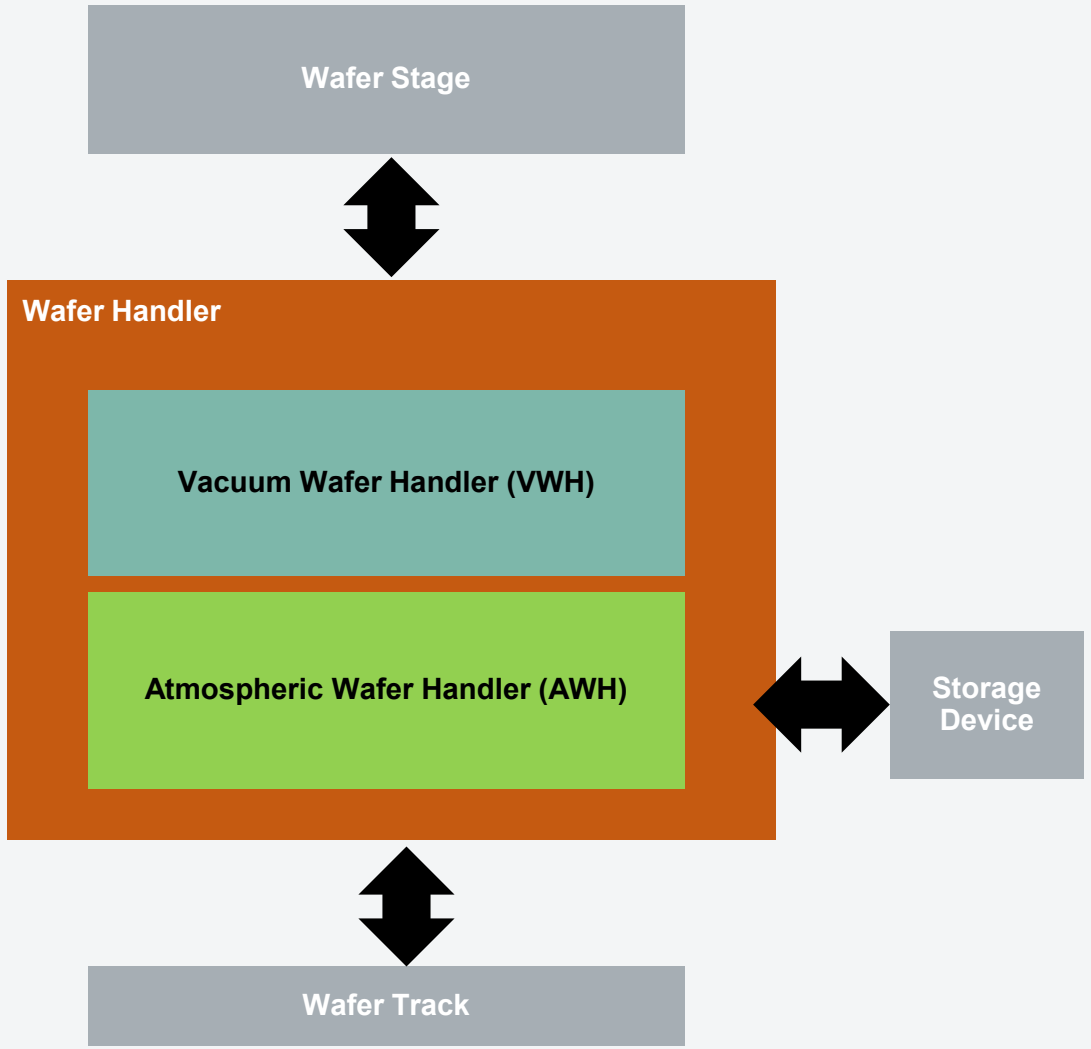
Wafer Handler in ASML's EUV lithography machine (Twinscan)



Context

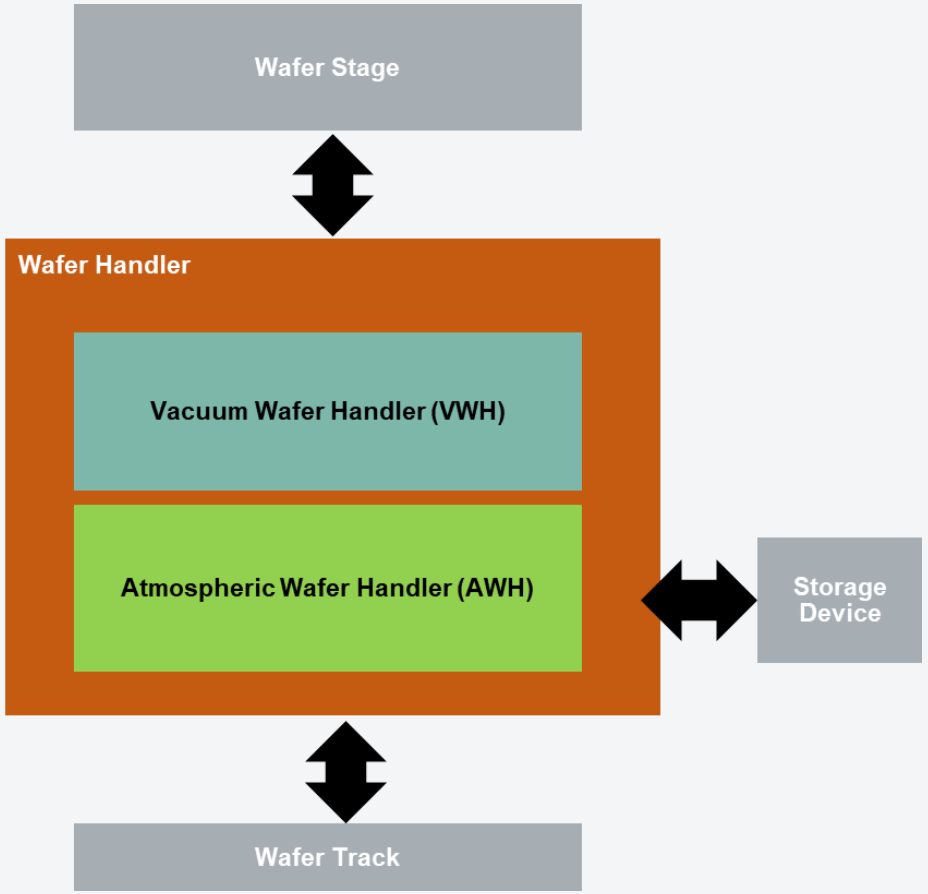
High level function of WH

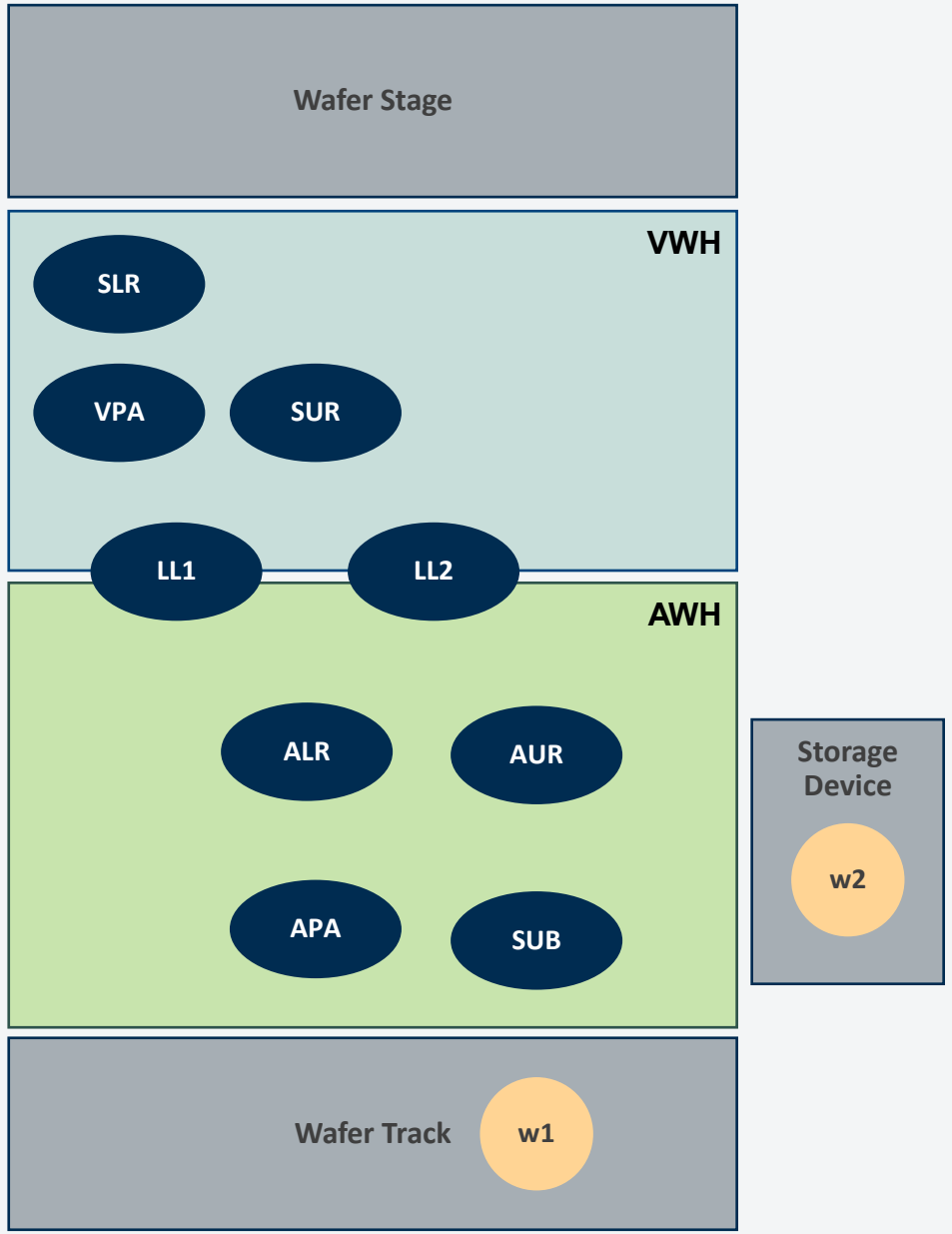
- Inputting and outputting wafers to and from Twinscan
- High throughput (>300 wafers per hour)
- Thermal conditioning (~3 mK)
- Accuracy/Alignment (~45 μm)
- Defectivity free (<0.004 particles)



Context

Wafer flow: Simplified WH





Context

Complexity of wafer handler

Different configurations:

- *Shifted*
- *Non-Shifted*
- *EUV vs DUV*
- ..

Multiple use cases

- *Production*
- *Recycling*
- *Calibration*
- ..

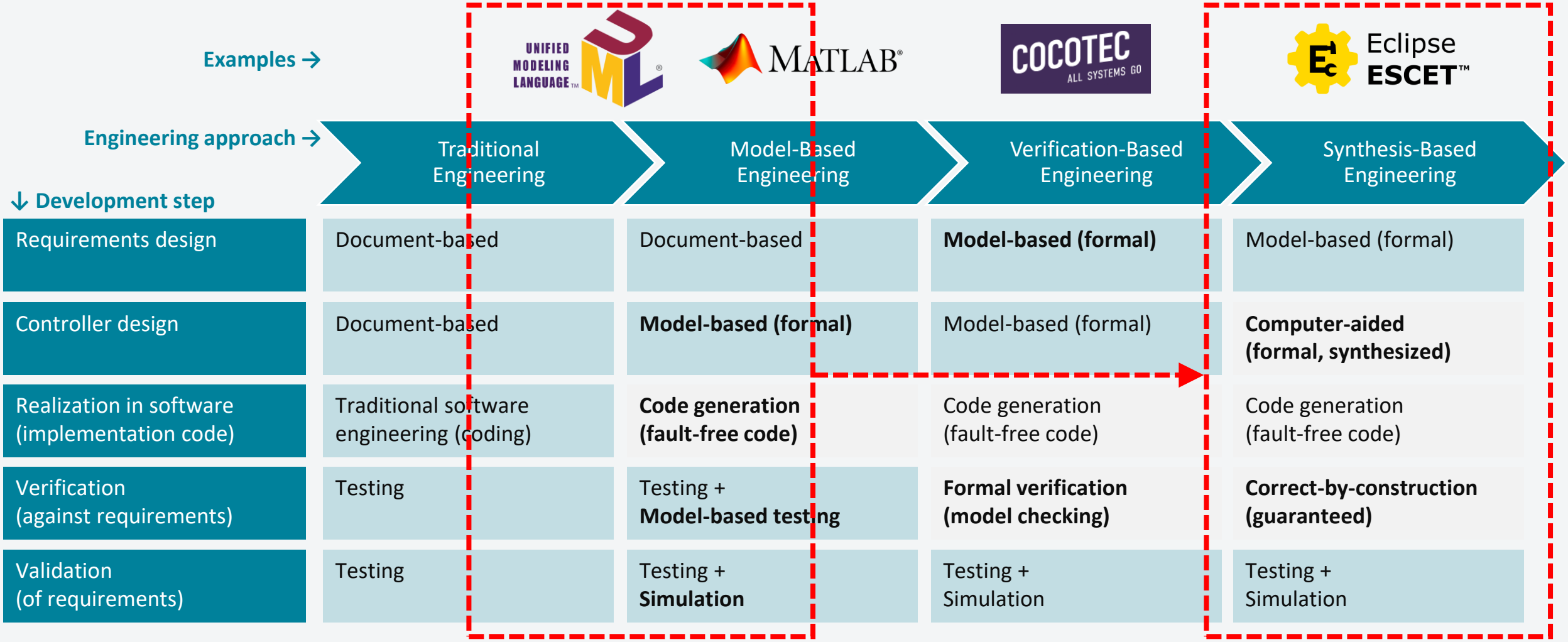
Multiple interfaces

- *Streaming*
- *Calibration*
- *GUI*
- ..

Error handling and recovery

- *Init and terminate*
- *Resync*
- *Abort and clear*
- ..

Controller design process

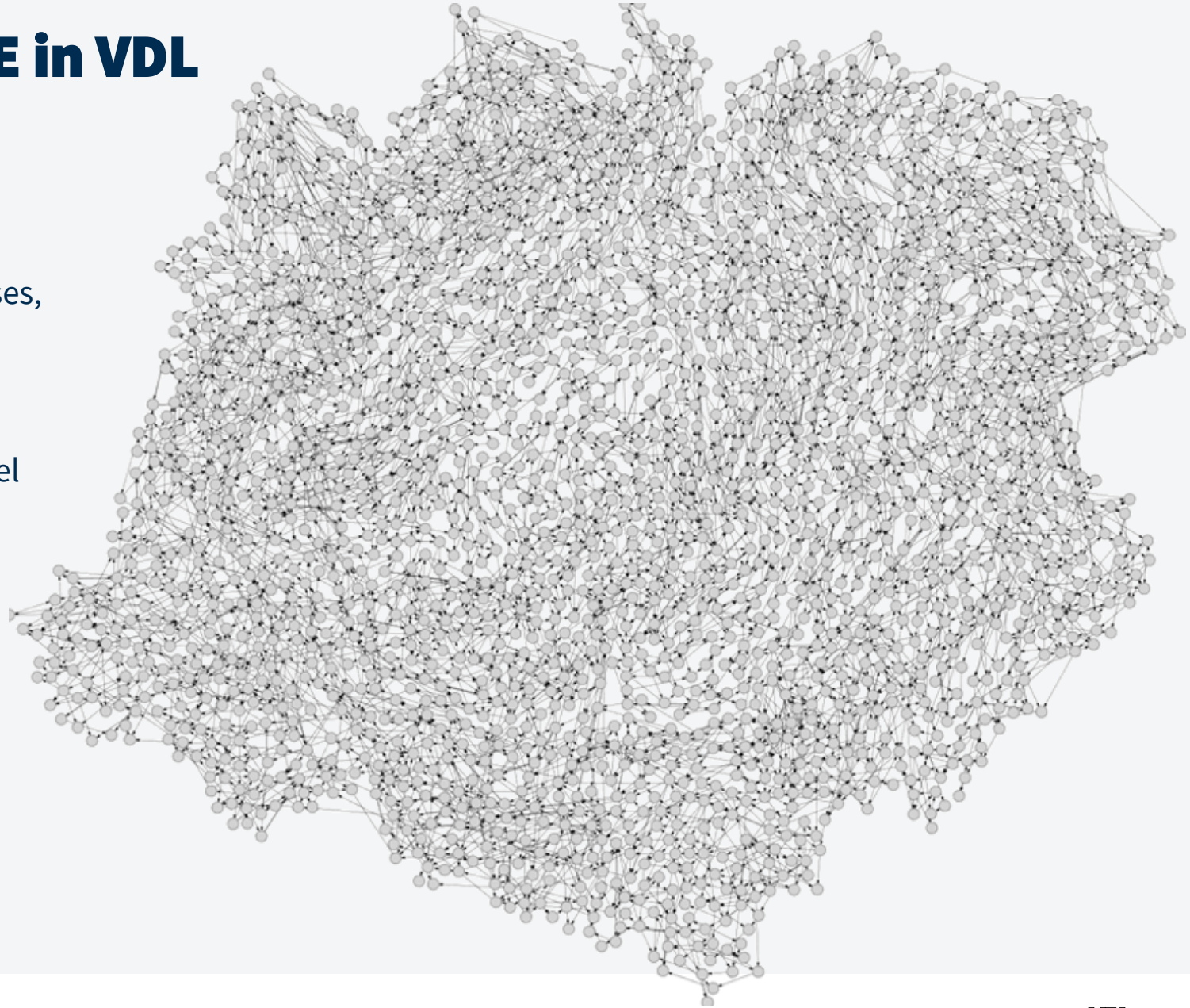


Legend: Manual work / Focus (Semi-)automatic

Challenges in adopting SBE in VDL

How to deal with complexity?

- State space explosion
  - Multiple configurations, clients, and use cases, recovery
- Requirements translation
  - From (incomplete) documents to UML model
- Introduction of new tool
  - Potential disruption in WoW



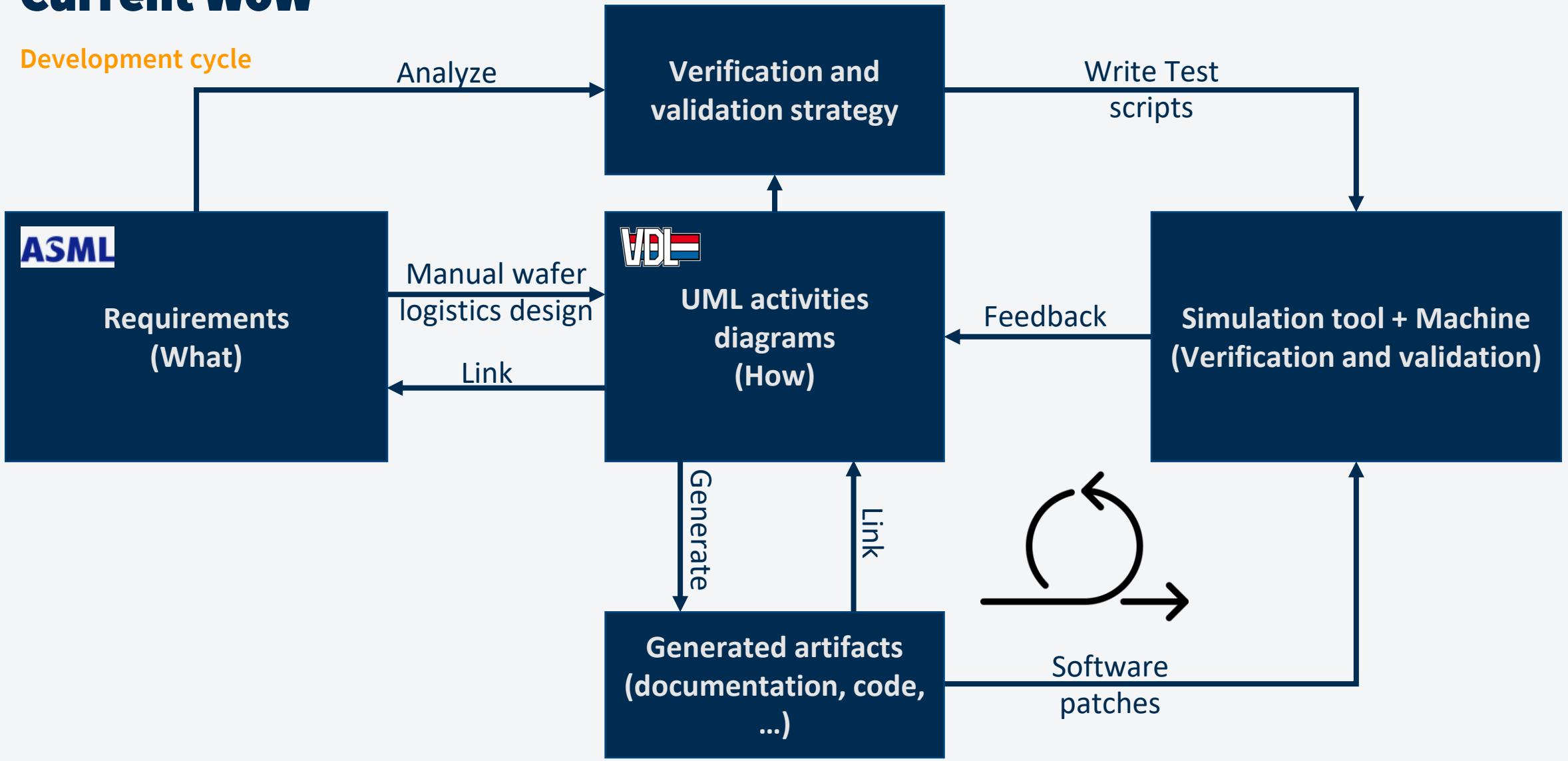
Poka Yoke applied research project (2023—2026)

Introduction

- Poka Yoke: Japanese term meaning 'error-proof' or 'error prevention', used in Lean manufacturing
- Project goal: Investigate to what extent SBE helps to manage the increasing complexity of WH
- Project partners: VDL-ETG, ASML, TNO-ESI

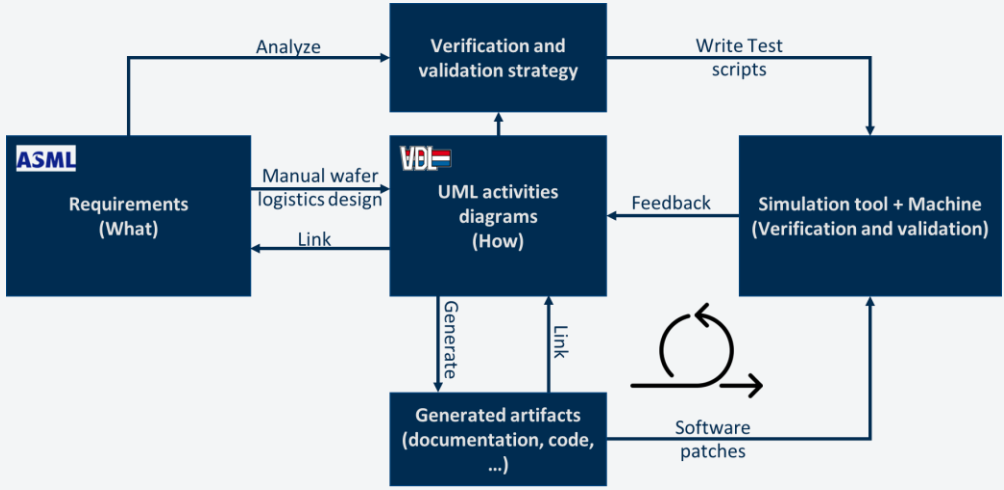
Current WoW

Development cycle



Current WoW

Challenges



Error prone

Long feedback

Difficult to test

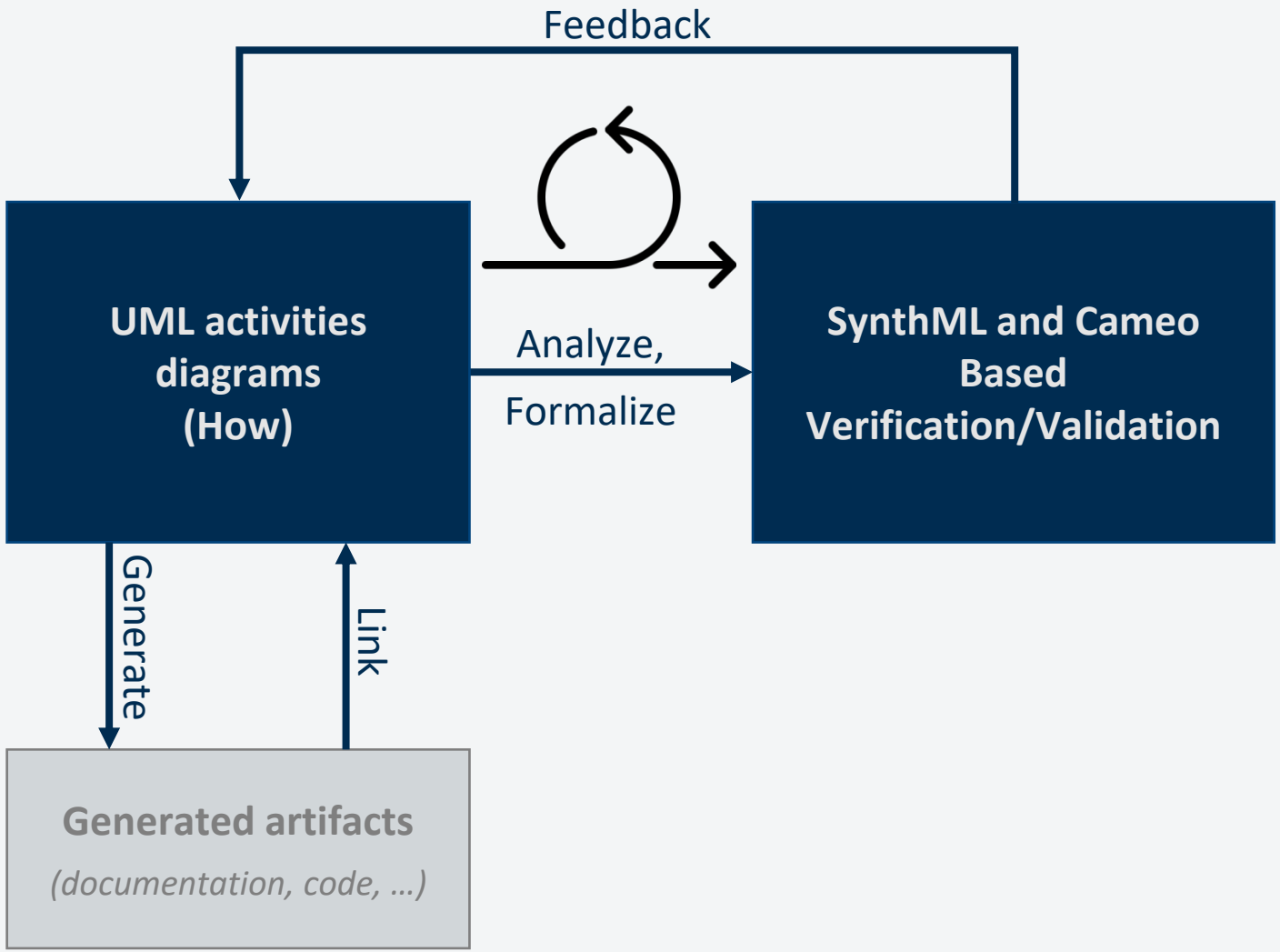
Late detection

Huge regression

Poka Yoke: Approach

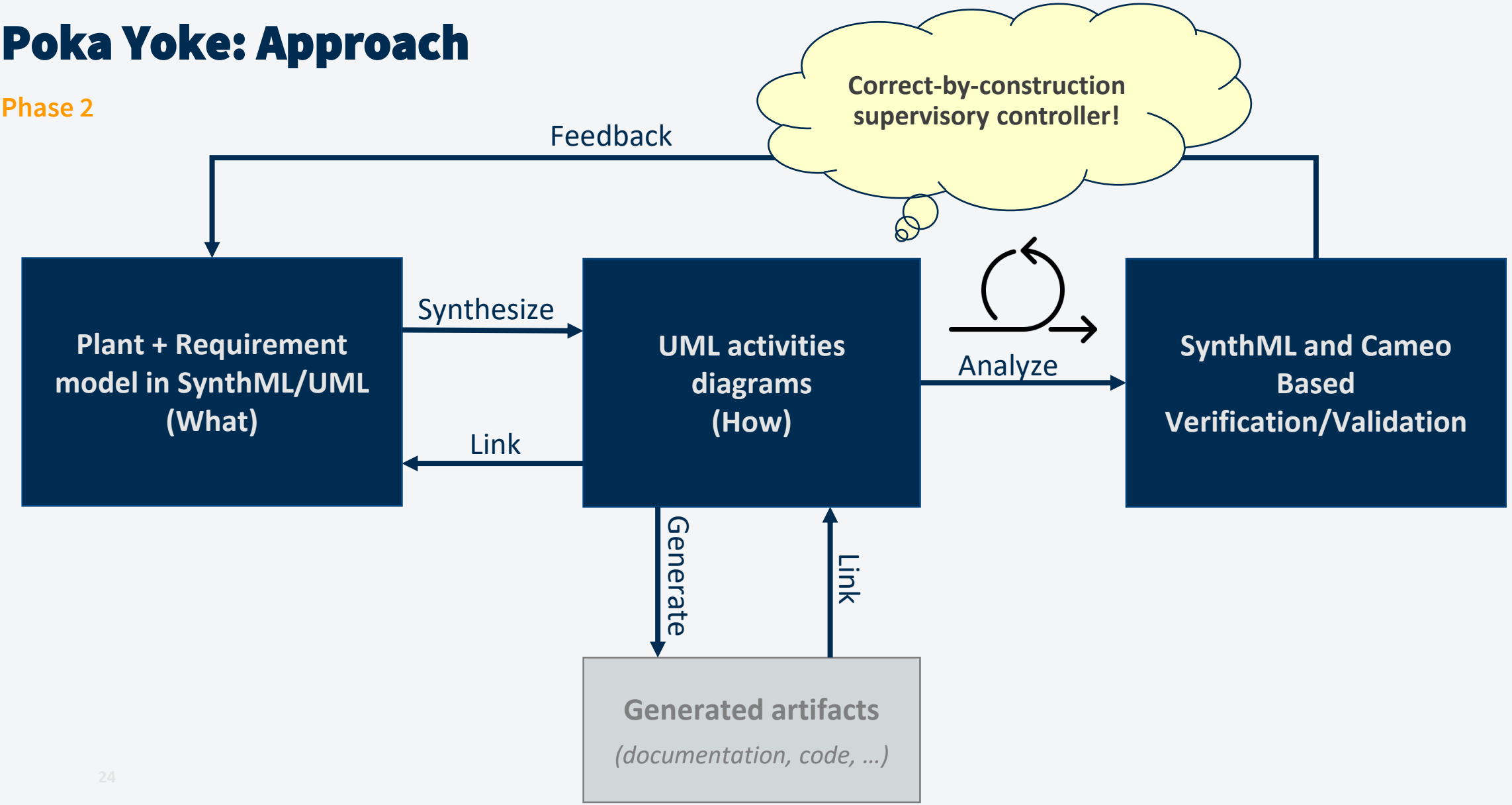
Phase 1

- Shorter design cycle
- Design iterations are software agnostics



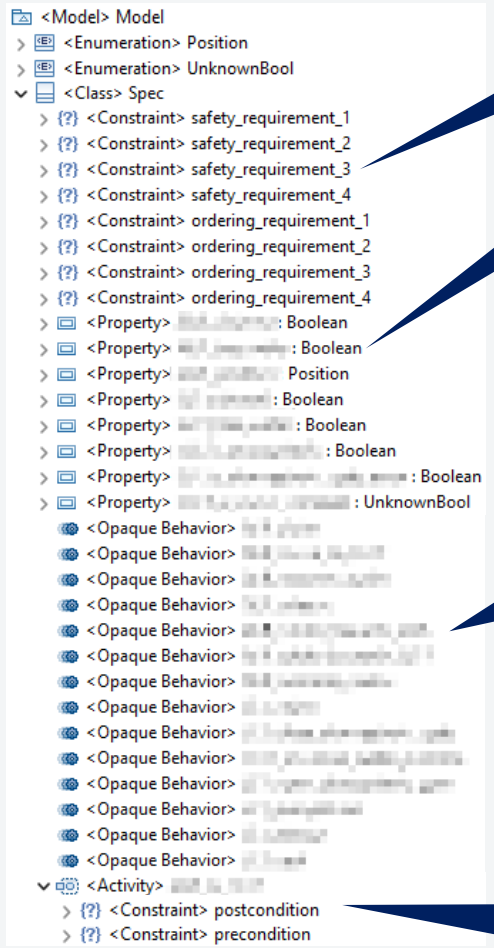
Poka Yoke: Approach

Phase 2



Poka Yoke: Results

Phase 2: Feasibility study of synthesizing controller in UML



III. Requirements

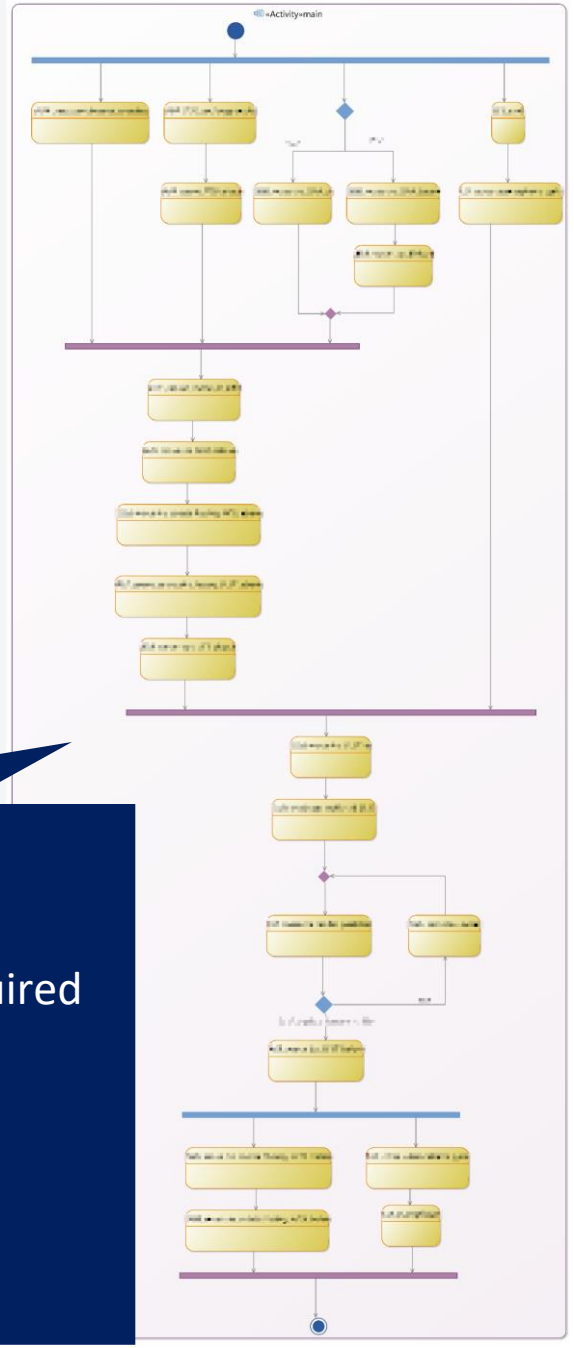
I. System state

II. Available actions

IV. Pre/post conditions

V. Synthesis under the hood, hidden from the user

Activity synthesis



Synthesized UML activity diagram:

- Automatically computed
- Automatically detects rework may be required
- Parallelized as much as possible
- Correct-by-construction with respect to requirements
- Guaranteed to be free of deadlocks
- Result presented in familiar UML notation

Poka Yoke: Results

Moving towards SBE!

Examples →

Engineering approach →

↓ Development step

↓ Development step	Traditional Engineering	Model-Based Engineering	Verification-Based Engineering	Synthesis-Based Engineering
Requirements design	Document-based	Document-based	**Model-based (formal)**	Model-based (formal)
Controller design	Document-based	**Model-based (formal)**	Model-based (formal)	**Computer-aided (formal, synthesized)**
Realization in software (implementation code)	Traditional software engineering (coding)	**Code generation (fault-free code)**	Code generation (fault-free code)	Code generation (fault-free code)
Verification (against requirements)	Testing	Testing + **Model-based testing**	**Formal verification (model checking)**	**Correct-by-construction (guaranteed)**
Validation (of requirements)	Testing	Testing + **Simulation**	Testing + Simulation	Testing + Simulation

Legend: Manual work / Focus (Semi-)automatic

THANK YOU!

Rijkswaterstaat
Ministerie van Infrastructuur en Waterstaat

Synthesis-Based Engineering of Supervisory Controllers

Applications at Rijkswaterstaat

Lars Moormann
lars.moormann@rws.nl

07-10-2025

Rijkswaterstaat

- Ministry of Infrastructure and Watermanagement.
- Design, control, and maintenance of the infrastructural systems in the Netherlands.
- These include locks, bridges, and tunnels.

Challenges at Rijkswaterstaat

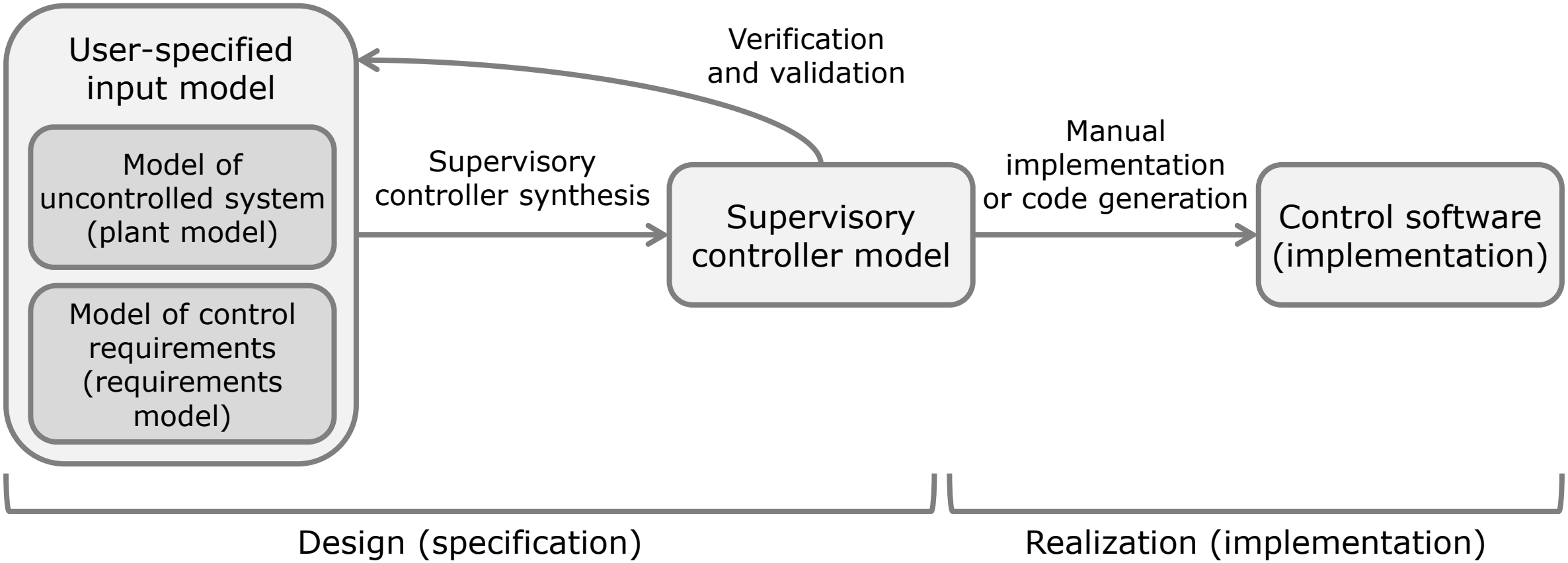
- Many of the infrastructural systems are coming to their end-of-life.
- System knowledge of RWS employees is low due to reliance on contractors.
- Projects are expected to be carried out 4 times as fast without increasing team capacity.
- A significant part of a project pertains the renewal of the supervisory controller.
- Over 300 dynamic infrastructural systems, each renovated every 5-10 years.

Proposed solution: Synthesis-Based Engineering

Synthesis-Based Engineering (SBE) at Rijkswaterstaat

Engineering approach → / **↓ Development step**	Traditional Engineering	Model-Based Engineering	Verification-Based Engineering	Synthesis-Based Engineering
Requirements design	Document-based	Document-based	**Model-based (formal)**	Model-based (formal)
System design	Document-based	**Model-based (formal)**	Model-based (formal)	**Computer-aided (formal, synthesized)**
Realization in software (implementation code)	Traditional software engineering (coding)	**Code generation (fault-free code)**	Code generation (fault-free code)	Code generation (fault-free code)
Verification (against requirements)	Testing	Testing + **Model-based testing**	**Formal verification (model checking)**	**Correct-by-construction (guaranteed)**
Validation (of requirements)	Testing	Testing + **Simulation**	Testing + Simulation	Testing + Simulation

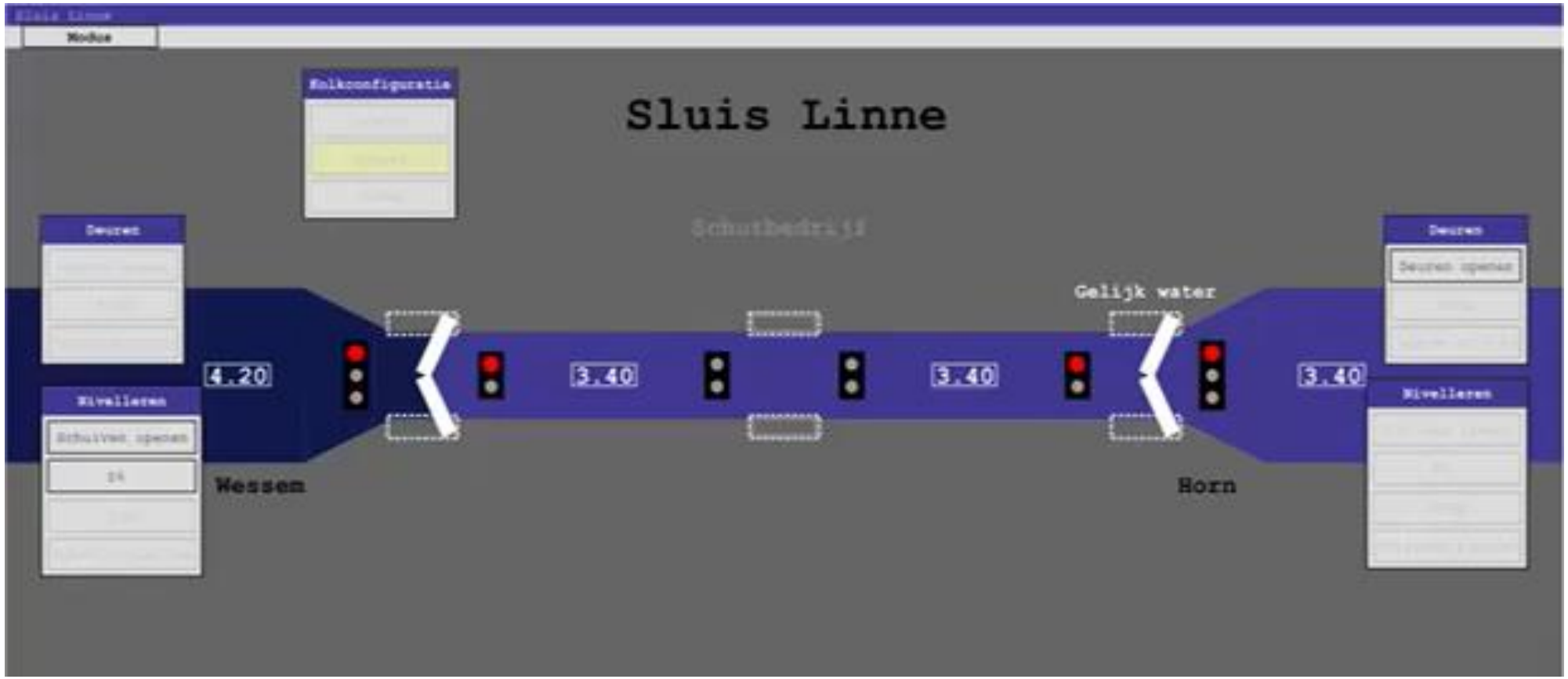
Applications of Synthesis-Based Engineering at Rijkswaterstaat



Lock Linne

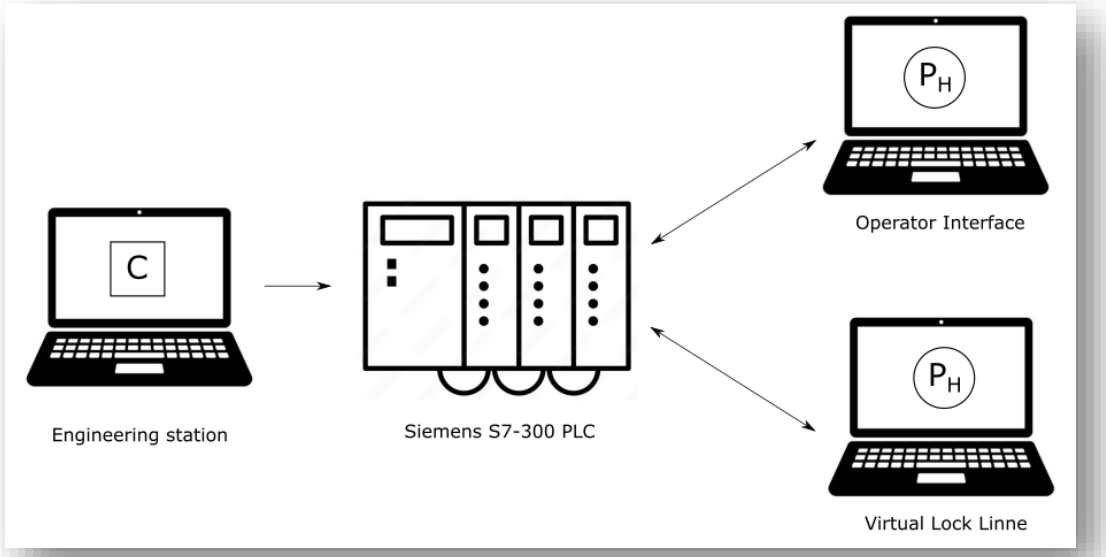
- Internal case study for Lock Linne.
- Creating the SBE model:
  - 121 components
  - 1072 requirements
- Supervisor is synthesized in 6 seconds:
  - $4.33 \cdot 10^{249}$ controlled system states
- Synthesized supervisor is validated through visualized model simulations.

Demo of model simulation



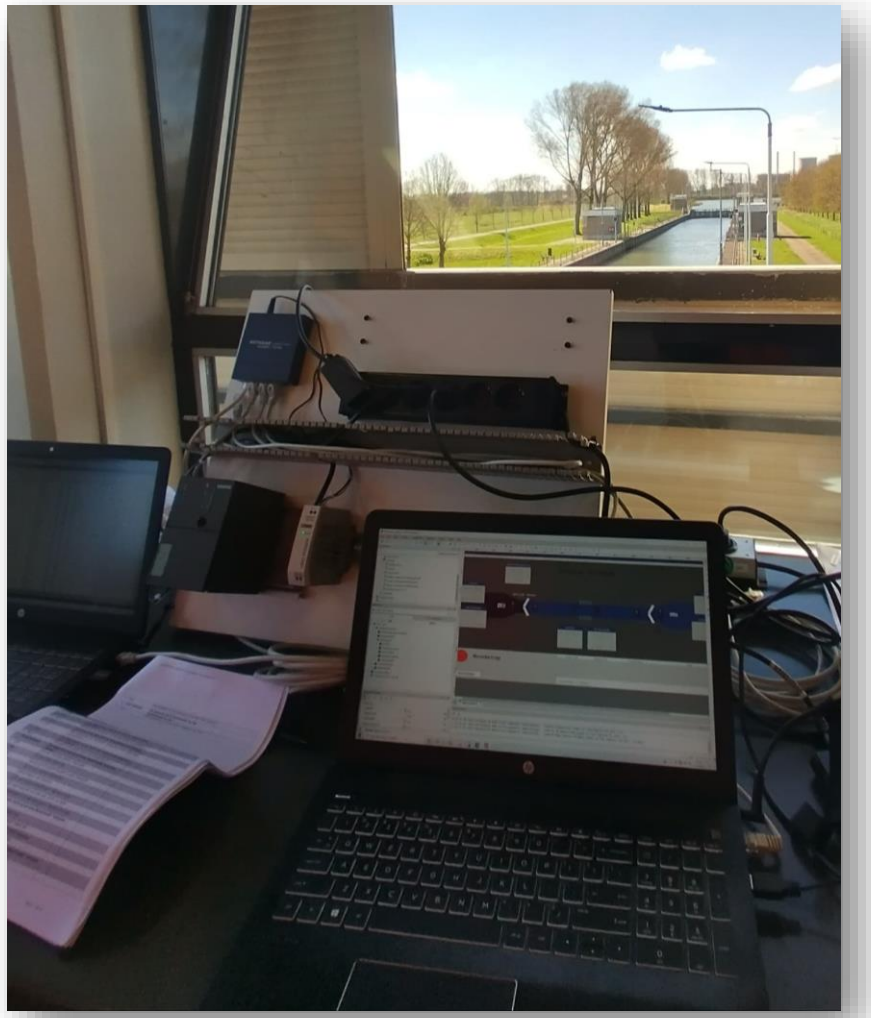
Programmable Logic Controller (PLC) code generation

- PLC code is automatically generated.
- The generated code is implemented in a hardware-in-the-loop (HIL) setup.
- The controlled system behavior is validated through HIL simulations.

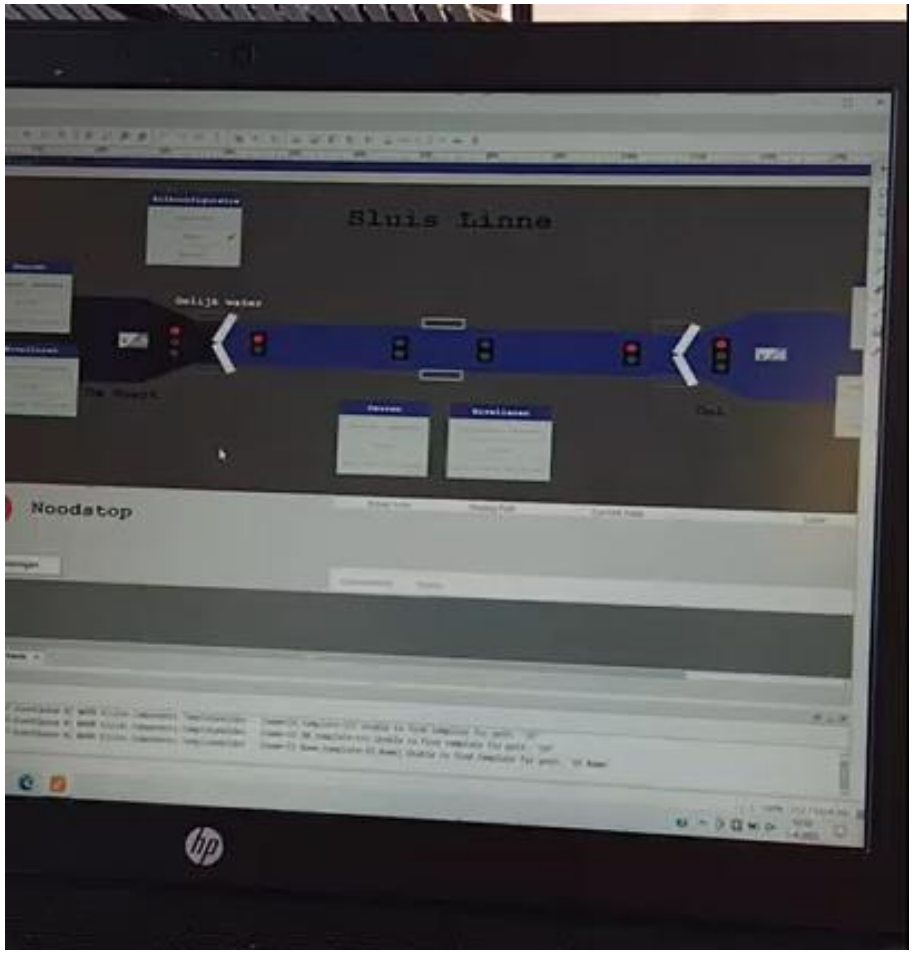


At Lock Linne

- Testing at the physical lock in Linne.
- Regular PLC is unplugged, and the SBE PLC is plugged in.
- Correct functionality has been showcased through extensive testing protocols.



Demo of testing at Lock Linne



Conclusions and key takeaways

By using SBE for the design of the supervisory controller:

- the quality of the controller improves
- the system knowledge of RWS employees increases
- it is estimated that errors are reduced by a factor of 4, and time-to-market by a factor of 10
- we are actually able to tackle these 300 projects every 5-10 years

SBE is actively being used within Rijkswaterstaat and will be further developed and integrated in the future.

Questions & Answers

Synthesis-Based Engineering helps to manage the high complexity

Better understanding

Reducing the engineering effort

Specify the **what** rather than the **how**

Increased efficiency

Improving time to market

Highly automated: ‘push-button’ approach

Improved quality

Avoiding costly engineering mistakes

Correctness by construction

Develop trustworthy systems at reduced effort and cost