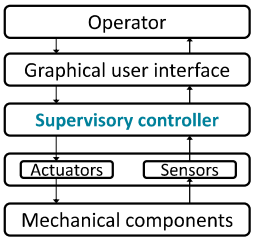
Synthesis-Based Engineering for Cyber-Physical Production Systems

Yuri Blankenstein, Jos Hegge, Dennis Hendriks, Piërre van de Laar, Wytse Oortwijn, Bram van der Sanden, Laura van Veen, Nan Yang



Cyber-Physical Production Systems are systems containing mechatronic components coordinated by control software. Supervisory controllers should orchestrate all resources to work together in the right way to ensure safe, correct, and optimal system behavior.

Engineering reliable supervisory controllers is challenging due to their complex nature

For example:
- High degree of concurrency
- Multi-disciplinary teams & domains
- Many configurations and variation points for product lines
- Continuously evolving functional and safety requirements

Synthesis-Based Engineering (SBE) is the logical next step in controller engineering

Engineering approach → / ↓ Development step	Traditional Engineering	Model-Based Engineering	Verification-Based Engineering	Synthesis-Based Engineering
Requirements design	Documents	Documents	**Models**	Models
Controller design	Documents	**Models**	Models	**Computer-aided (synthesis)**
Realization in software	Traditional coding	**Code generation**	Code generation	Code generation
Verification	Testing	Testing + **Model-based testing**	**Formal verification**	**Correct-by-construction**
Validation	Testing	Testing + **Simulation**	Testing + Simulation	Testing + Simulation

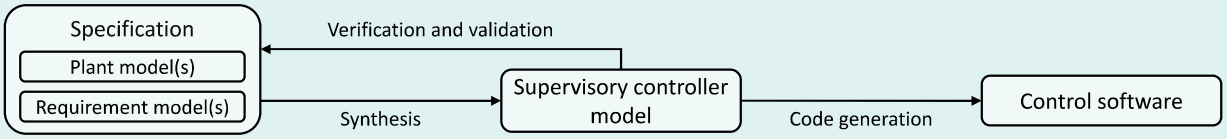
Legend: Manual work / Focus | (Semi-)automatic | **Delta to previous approach**

Better understanding
Specify the *what (requirements)* rather than the *how (implementation)*

Increased efficiency
Highly automated: push-button approach

Improved quality
Correctness by construction

SBE ways of working speeds up the development cycle



High-level specification

1. Specify pre- and postconditions of activity

Pre: Robot is at location 1 and has a product, and location 2 has no product.
Post: Location 2 has a product, and the robot is at location 1.

2. Specify available actions

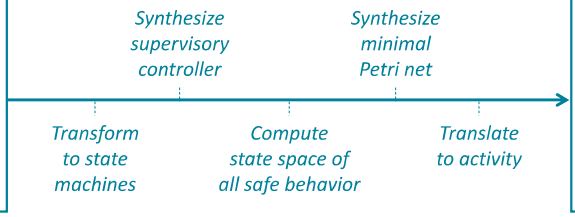
Actions	Precondition	Postcondition
`measure_product_position`	Location 2 is prepared, and product is at location 2	Product position is measured
`prepare_location2`	Location 2 is not prepared	Location 2 is prepared
`prepare_robot`	Robot is not prepared	Robot is prepared
`robot_adjust_product`	Robot is prepared and product is not correctly placed	Product position is adjusted
`robot_move_to_location1`	Robot is prepared	Robot is at location 1
`robot_move_to_location2`	Robot is prepared	Robot is at location 2
`robot_release_product_at_location2`	Robot is at location 2 while holding a product, and location 2 has no product	The robot holds no product, and location 2 has a product

3. Specify requirements

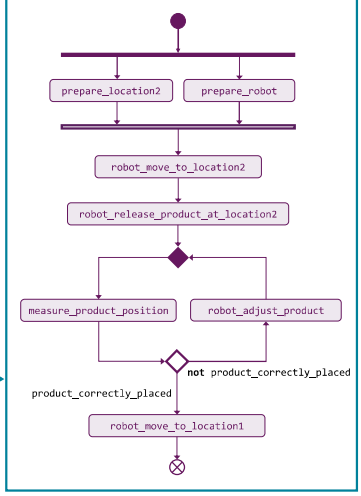
All actions except `measure_product_position` and `robot_adjust_product_position` happen at most once.

Poka Yoke applied research project: synthesize activities from requirements

TNO-ESI, ASML, and VDL-ETG are working together in an applied research project to synthesize activities for the supervisory control of wafer handlers. This example synthesizes an activity where a robot moves a product from one location to another.



Activity diagram



This research is carried out as part of the Poka Yoke project under the responsibility of TNO-ESI in co-operation with ASML and VDL-ETG. The research activities are supported by the Netherlands Ministry of Economic Affairs and Climate Policy, and TKI-HTSM.