Modular Model-based Supervisory Controller Design for Wafer Logistics in Lithography Machines

Bram van der Sanden, Michel Reniers, Marc Geilen, Twan Basten, Johan Jacobs, Jeroen Voeten, Ramon Schiffelers

MODELS 2015

ASML

Embedded Systems Innovation BY TNO

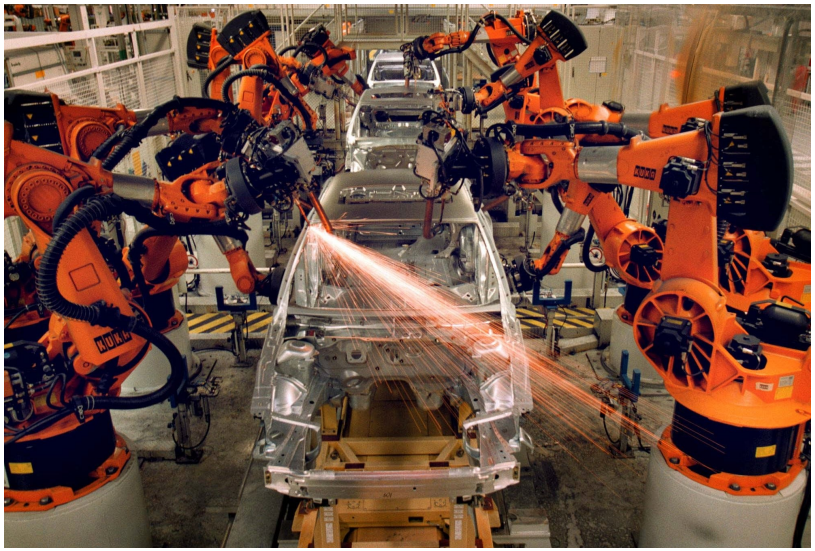
STW

Enabling new technology

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

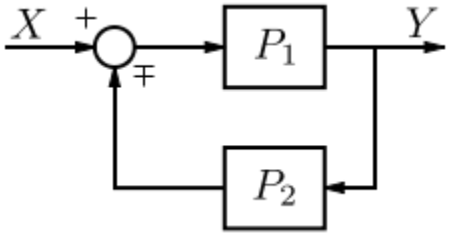
Supervisory Controllers



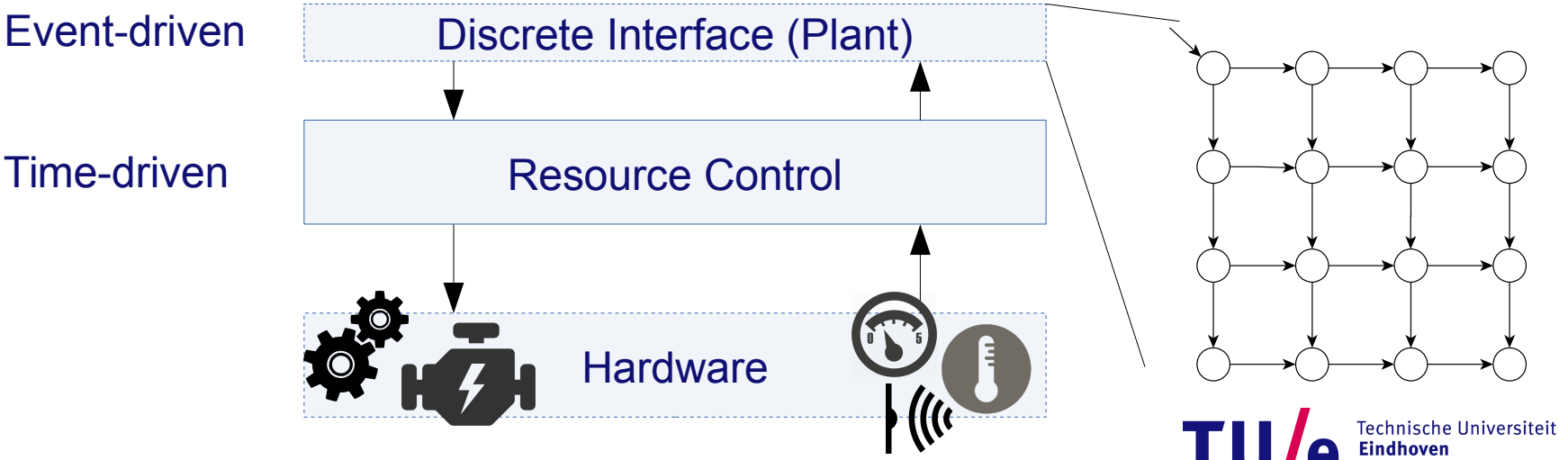
Positioning Supervisory Control

Positioning Supervisory Control

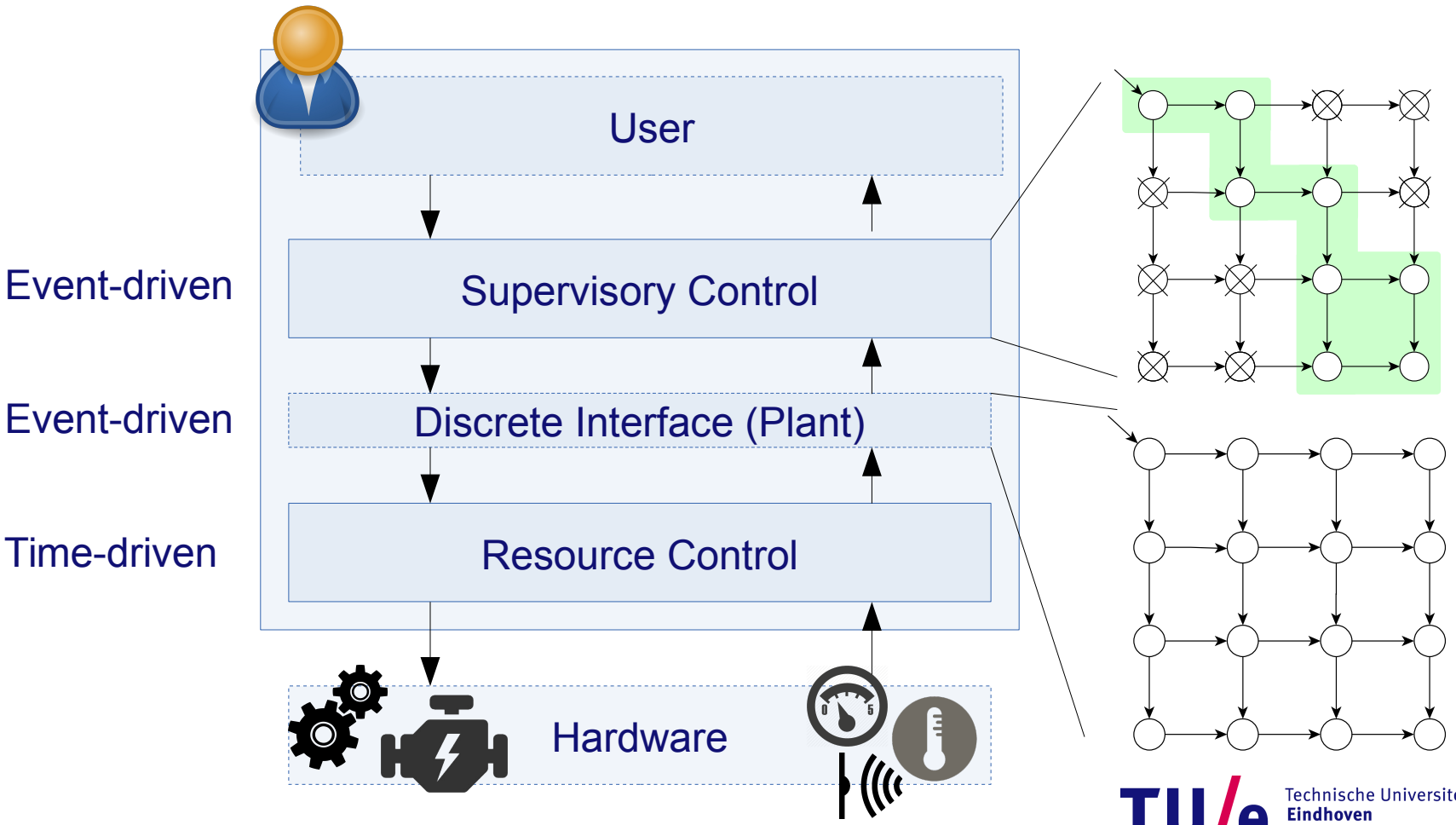
Time-driven



Positioning Supervisory Control



Positioning Supervisory Control



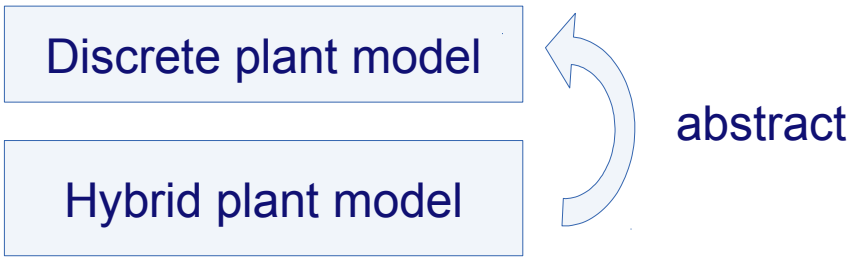
Supervisory controllers

How to *design* supervisory controllers?

How to Design Supervisory Controllers?

Hybrid plant model

How to Design Supervisory Controllers?



How to Design Supervisory Controllers?

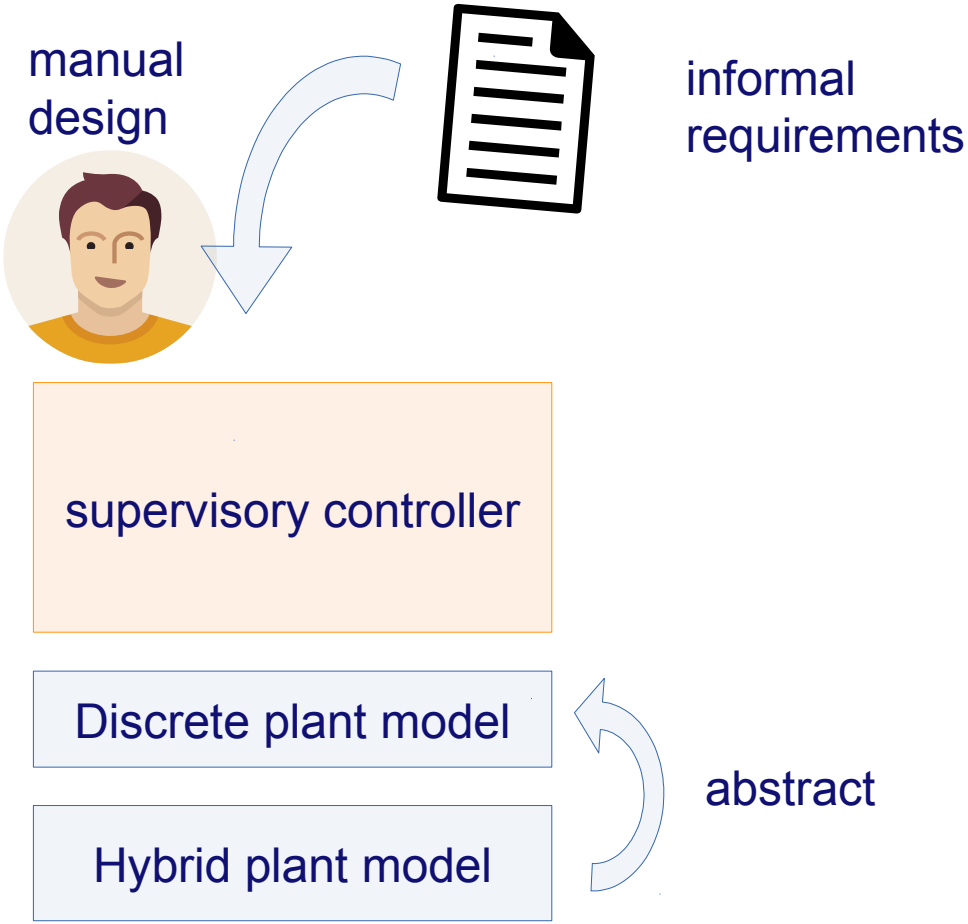
informal
requirements

Discrete plant model

Hybrid plant model

abstract

How to Design Supervisory Controllers?



How to Design Supervisory Controllers?

manual design

informal requirements

What about:
- requirement traceability?
- system is evolving: changing requirements?

supervisory controller

Discrete plant model

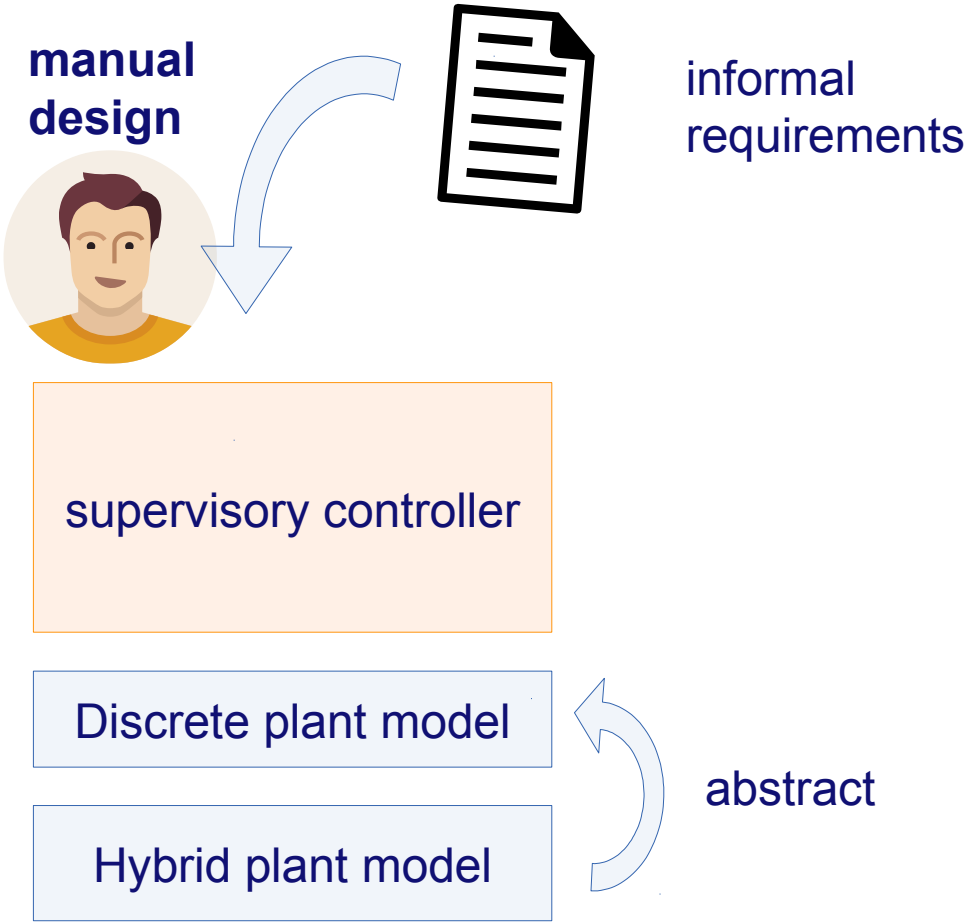
Hybrid plant model

abstract

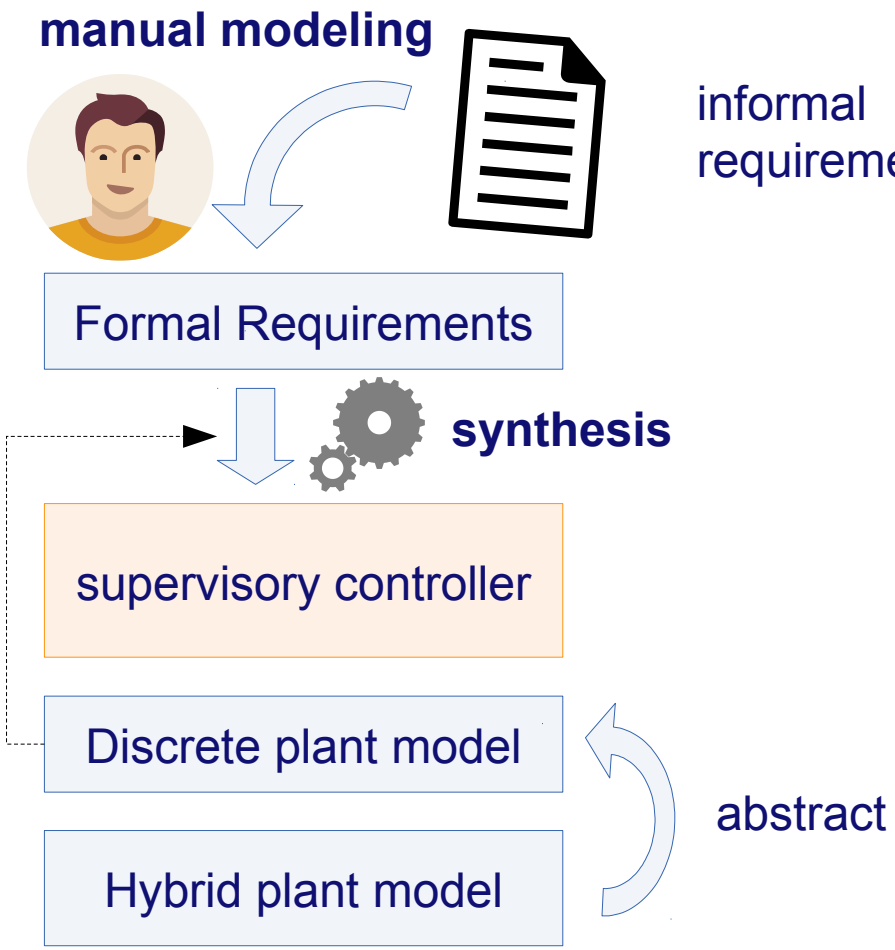
How to Design Supervisory Controllers?

Why not use **formally specified requirements** as part of the model?

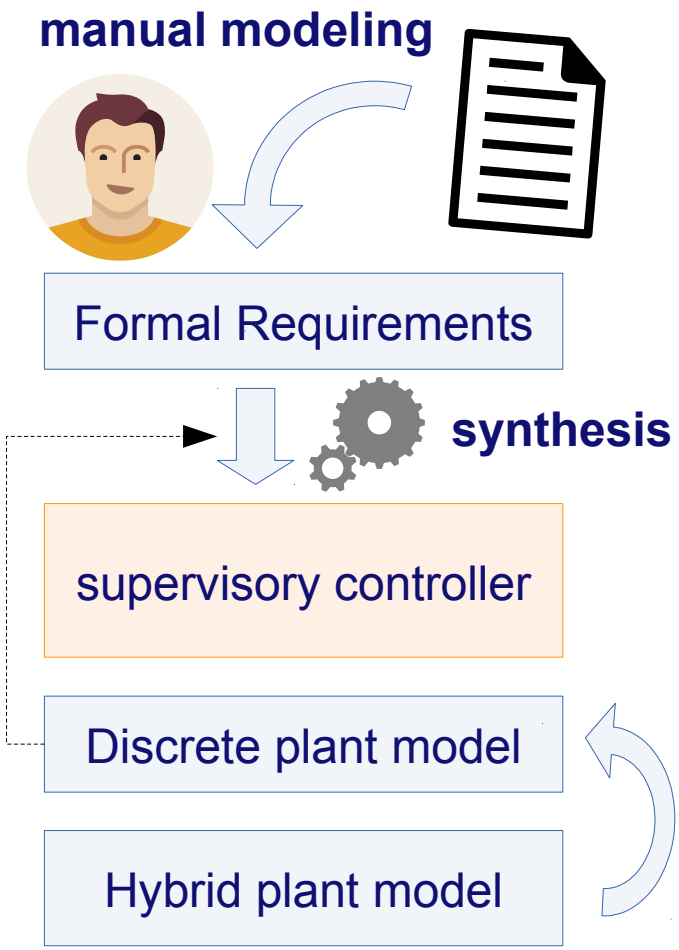
How to Design Supervisory Controllers?



How to Design Supervisory Controllers?

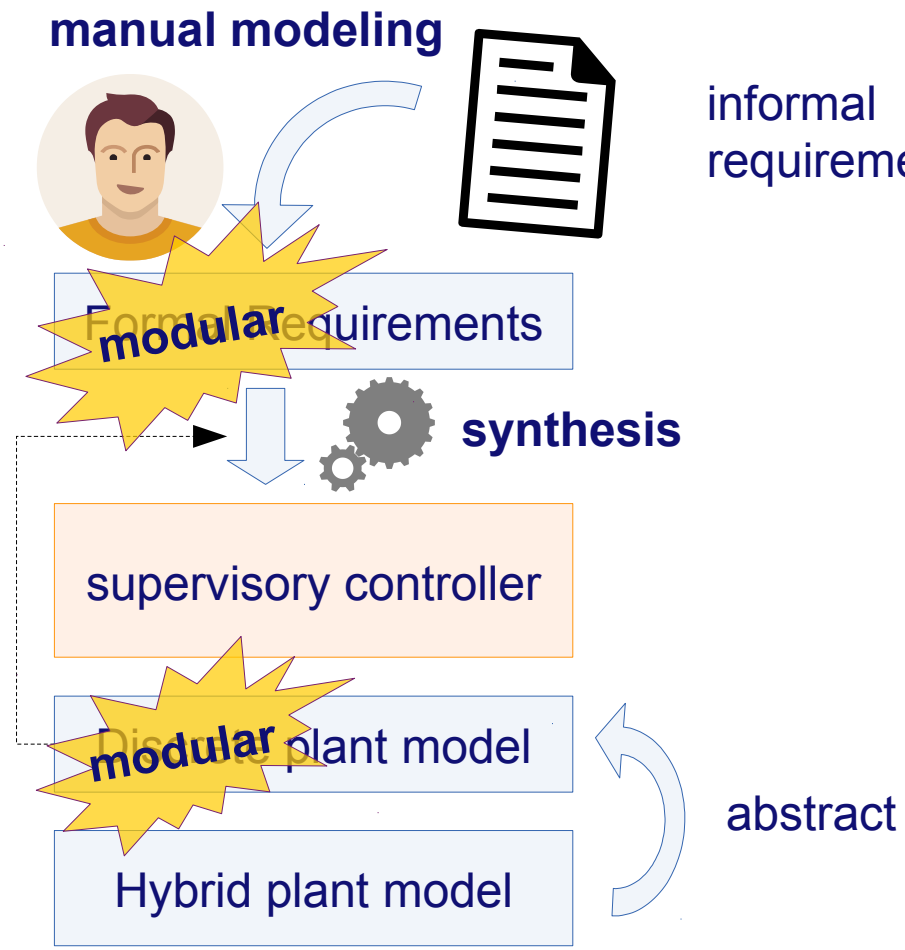


How to Design Supervisory Controllers?



Synthesis ensures controller:
- is deadlock-free
- is maximally permissive
- does not block environment actions

How to Design Supervisory Controllers?



Modular modeling

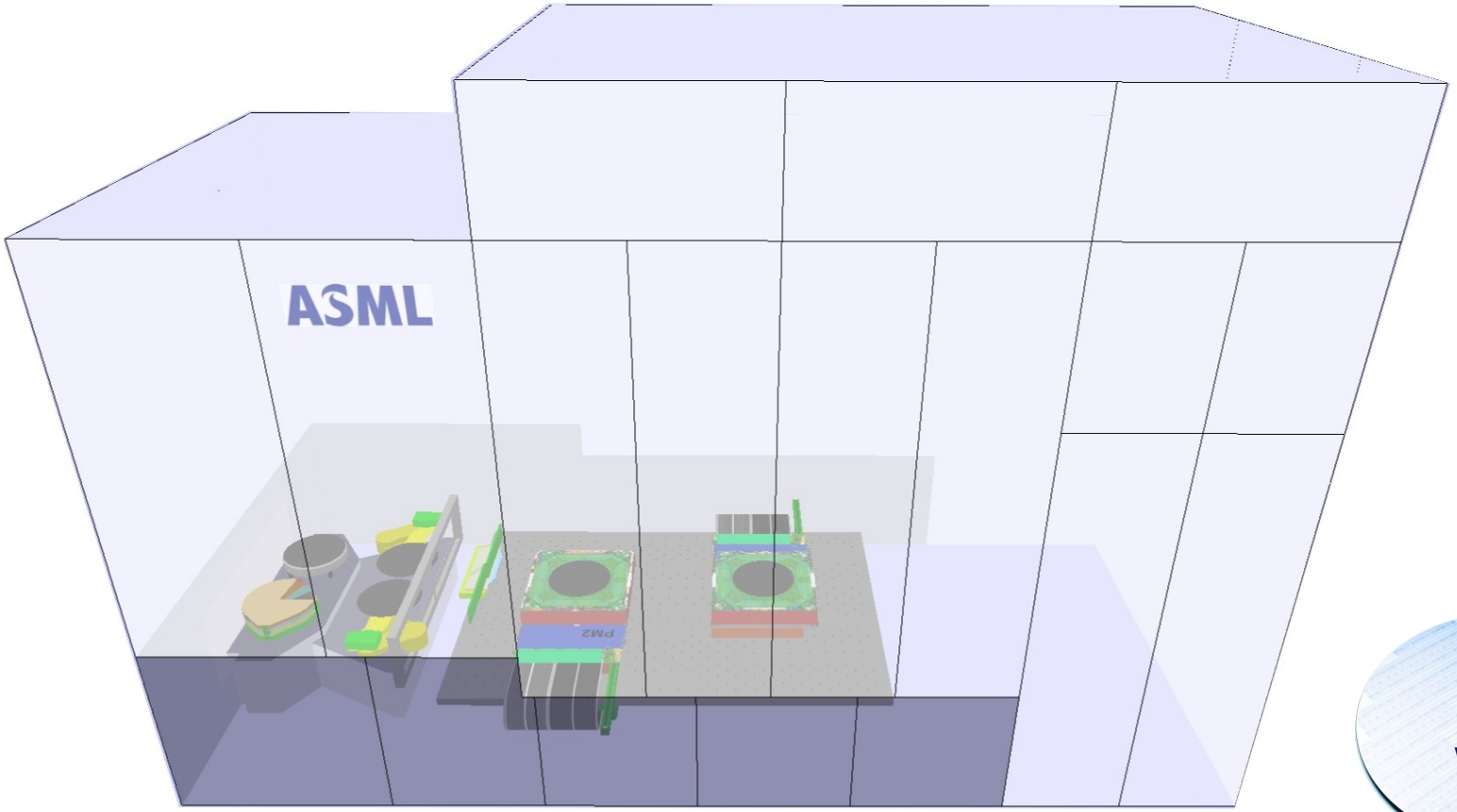
Synchronize models using
Multi-party event synchronization

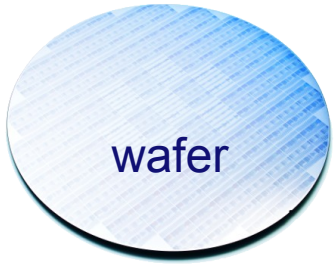
Model-based Development Process

Apply this development process to an

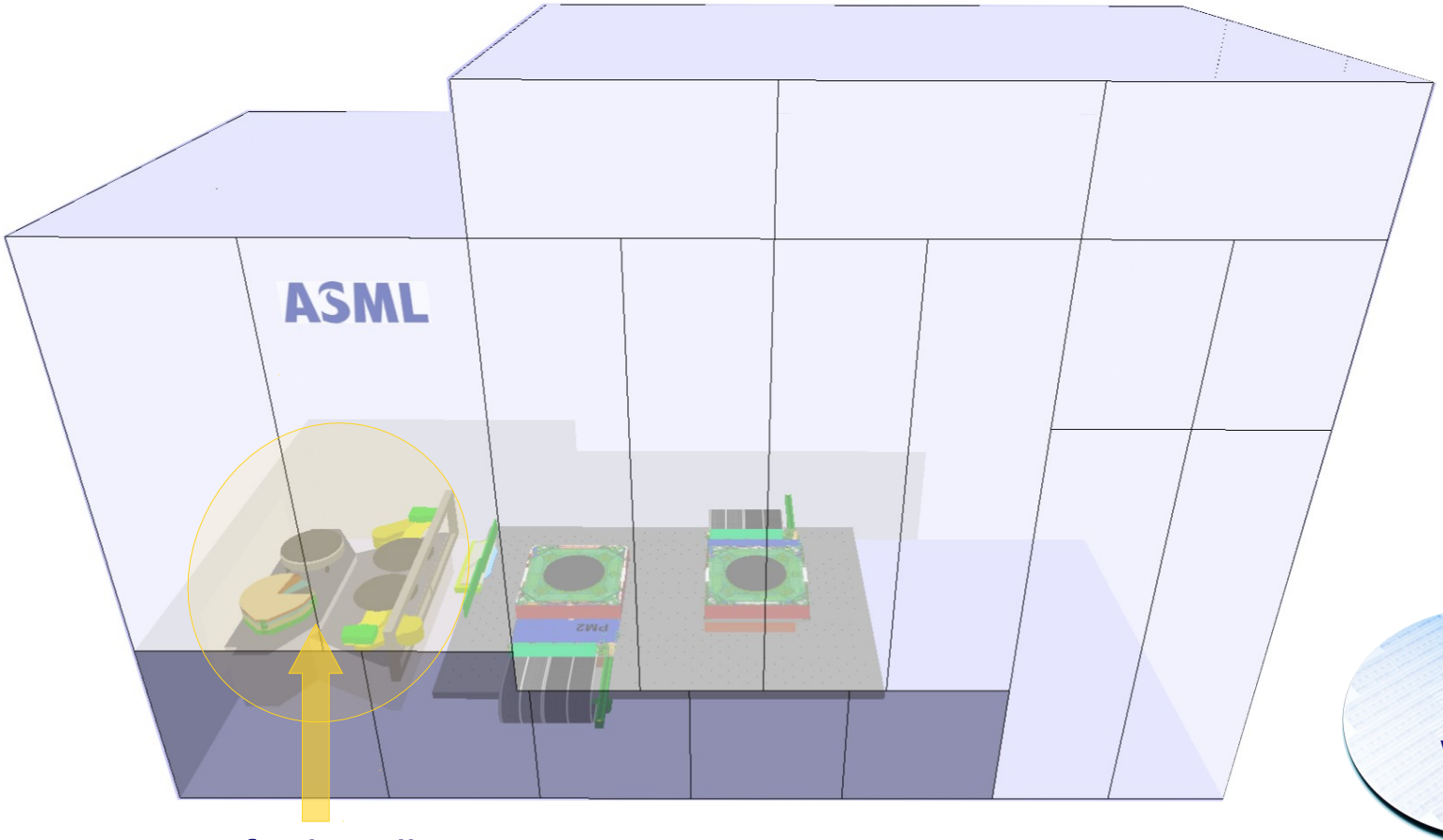
industrial case study

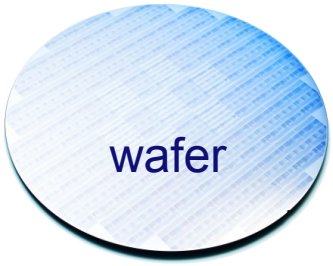
Case Study: Modeling the Wafer Logistics in a Lithography Machine



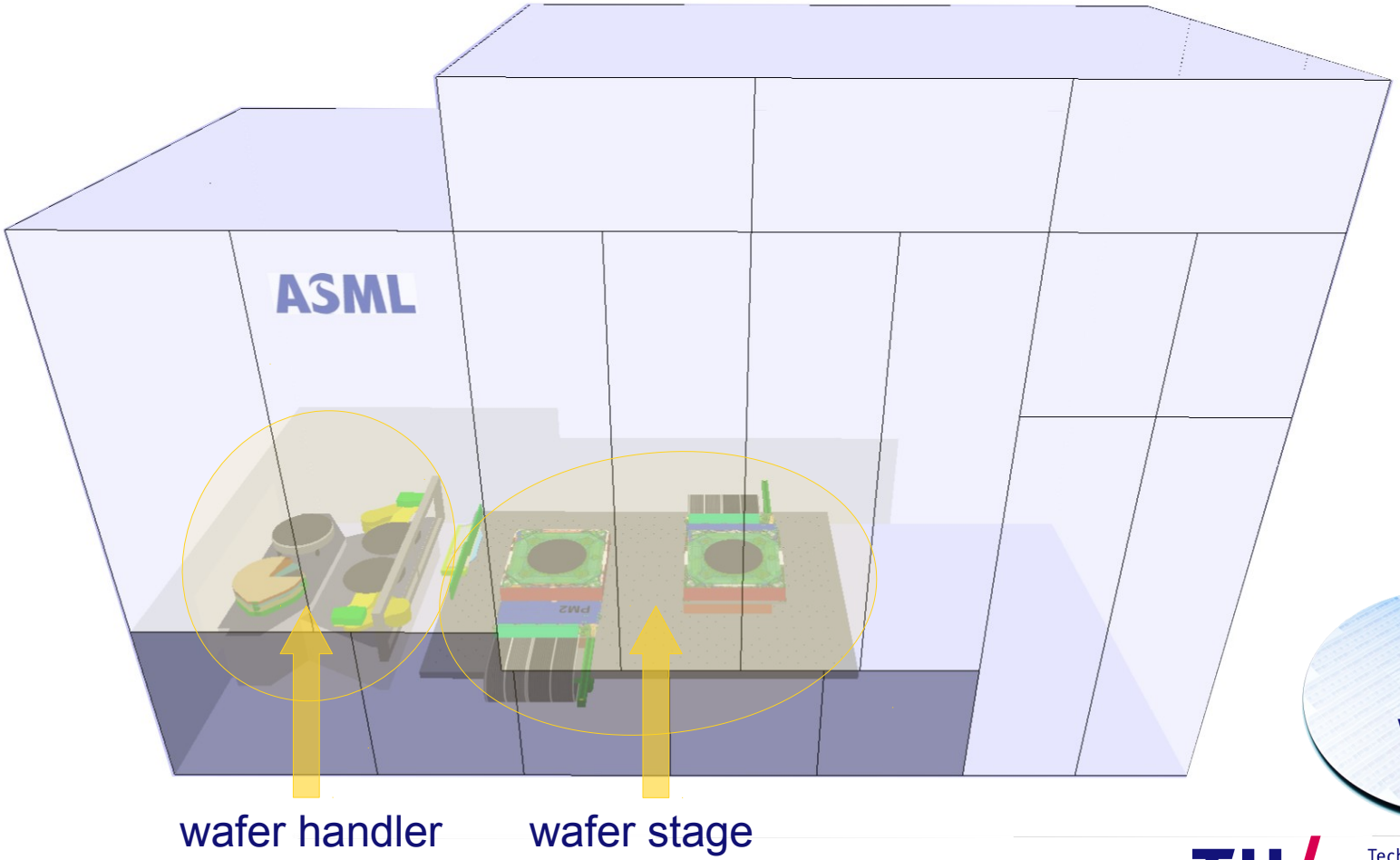


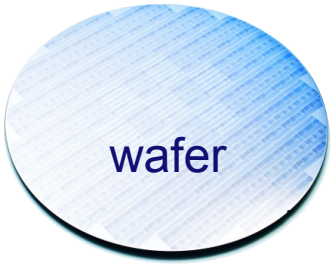
Case Study: Modeling the Wafer Logistics in a Lithography Machine





Case Study: Modeling the Wafer Logistics in a Lithography Machine



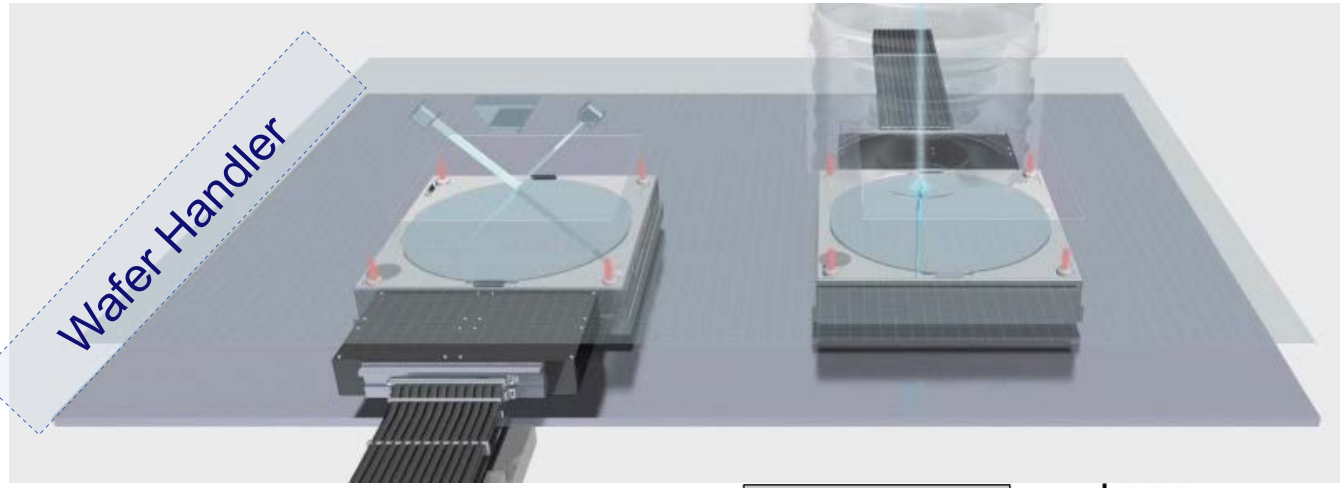


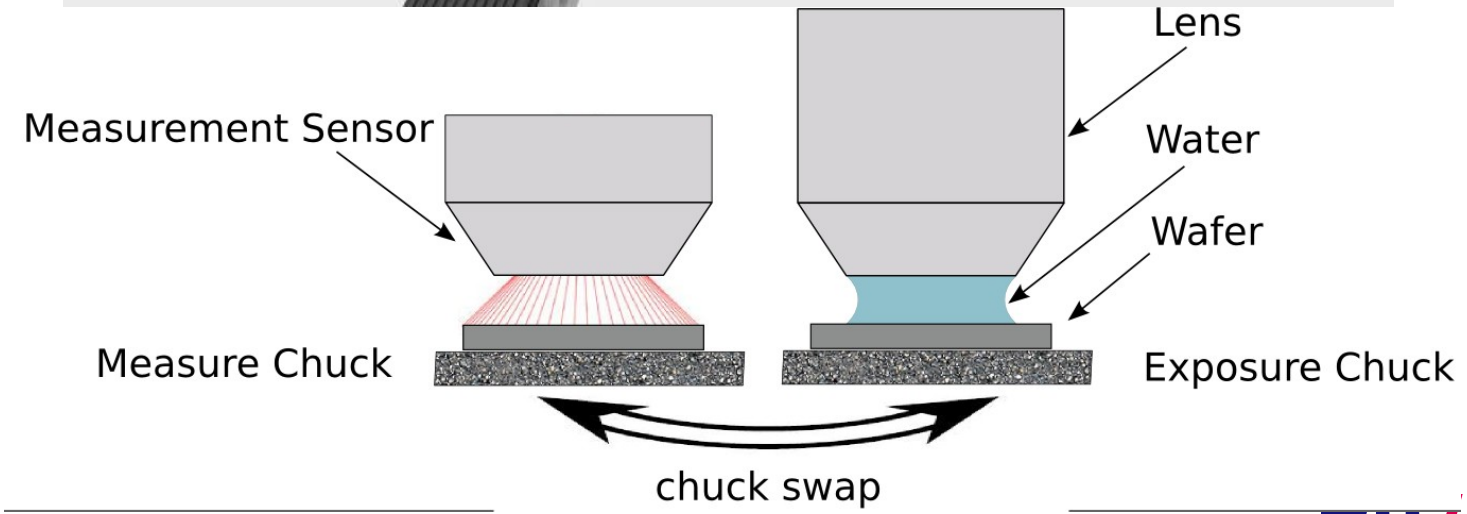
Case Study: Modeling the Wafer Logistics in a Lithography Machine

- What we have modeled in the paper:
  - Wafers
    - Positions in system
    - Life cycle requirement
  - System capabilities
  - Global flow restrictions (FIFO ordering)

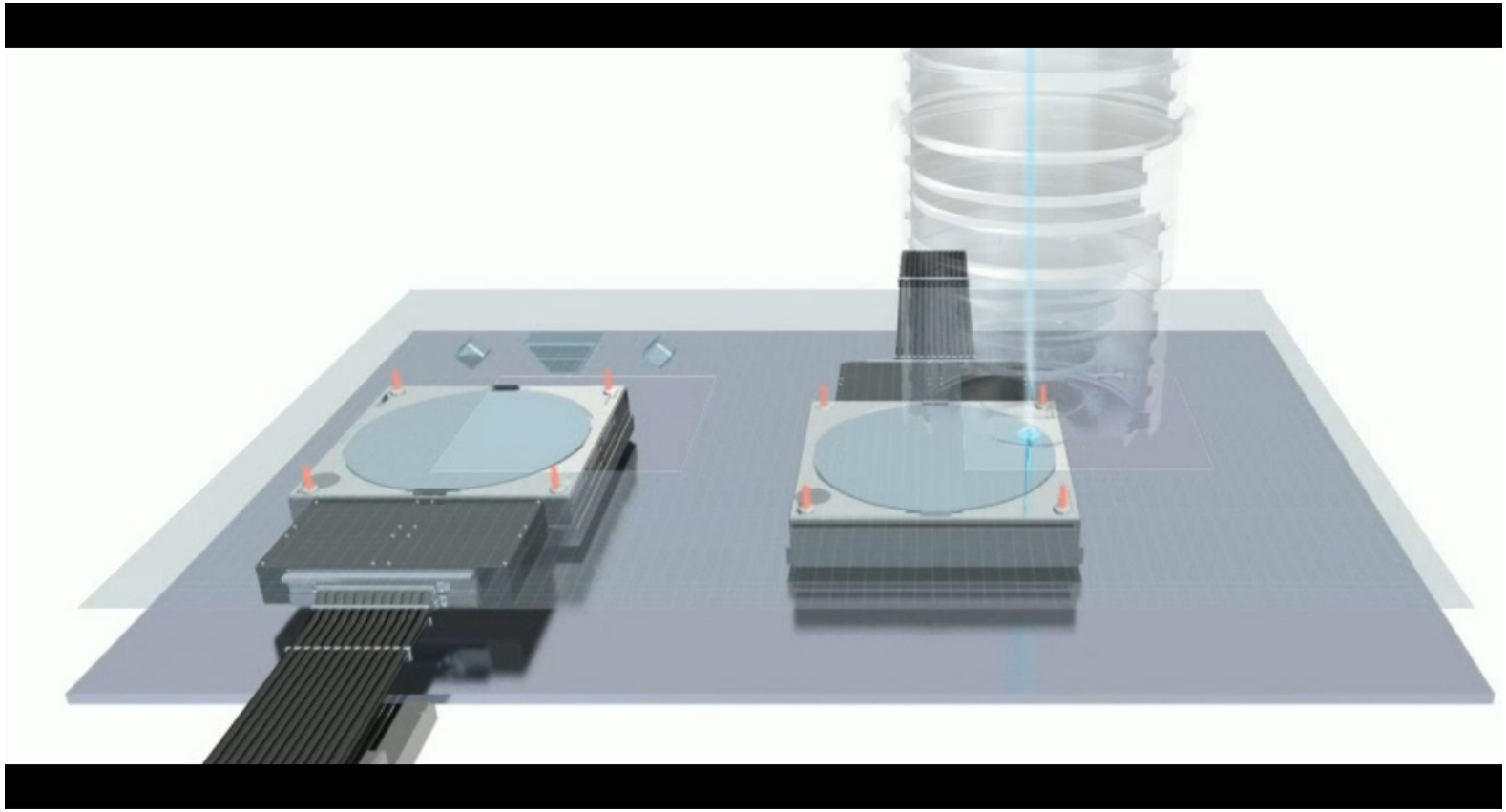
This presentation: focus on modeling the wafer stage

Wafer Stage: Measure, Expose and Chuck Swap

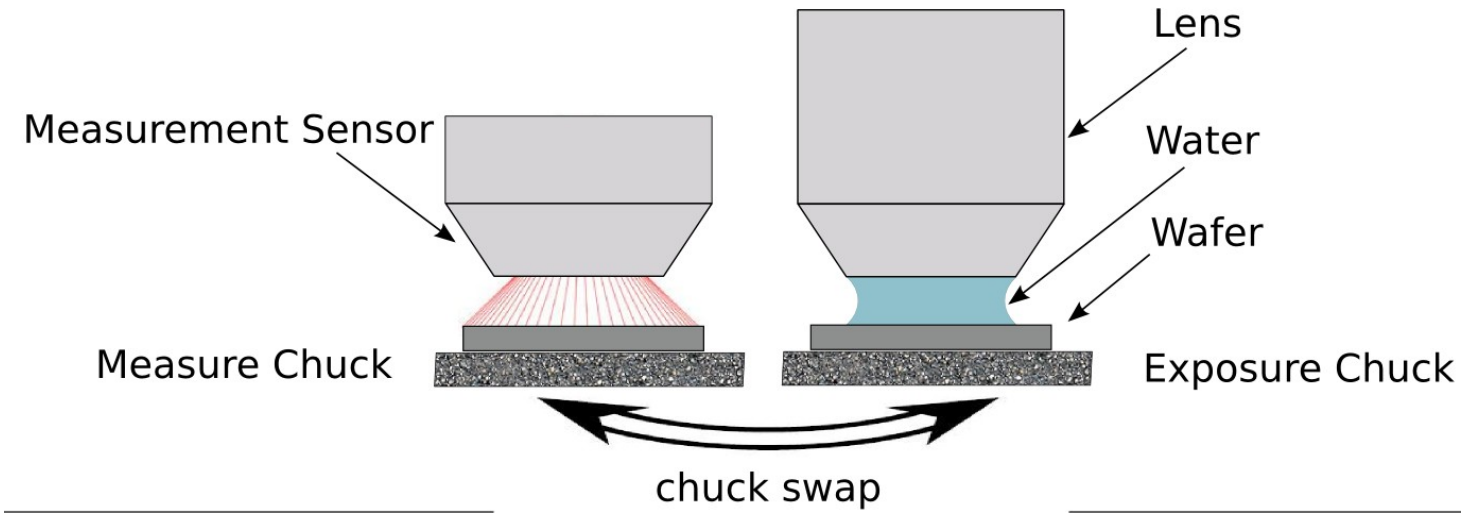




Wafer Stage: Measure, Expose and Chuck Swap



Modeling the Wafer Stage



(1) Chucks

Positions, allowed actions

Can hold at most one wafer

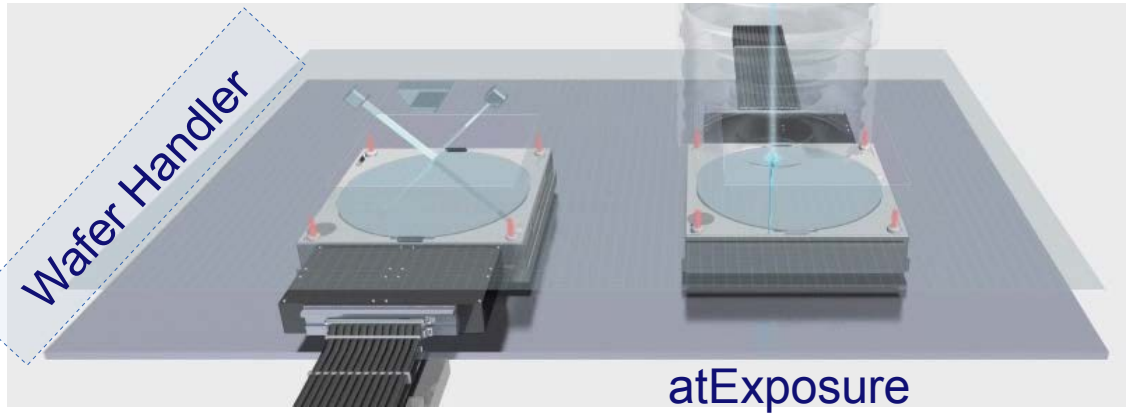
(2) Water layer

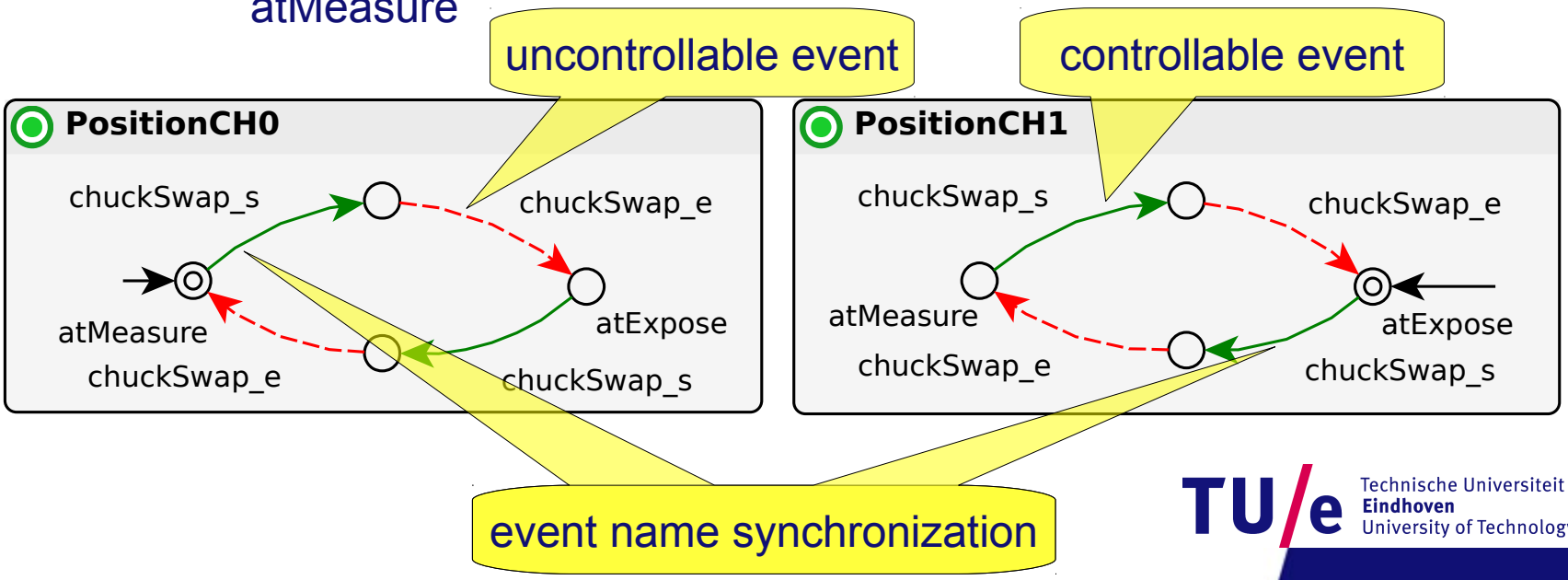
Below exposure lens

(3) Wafers

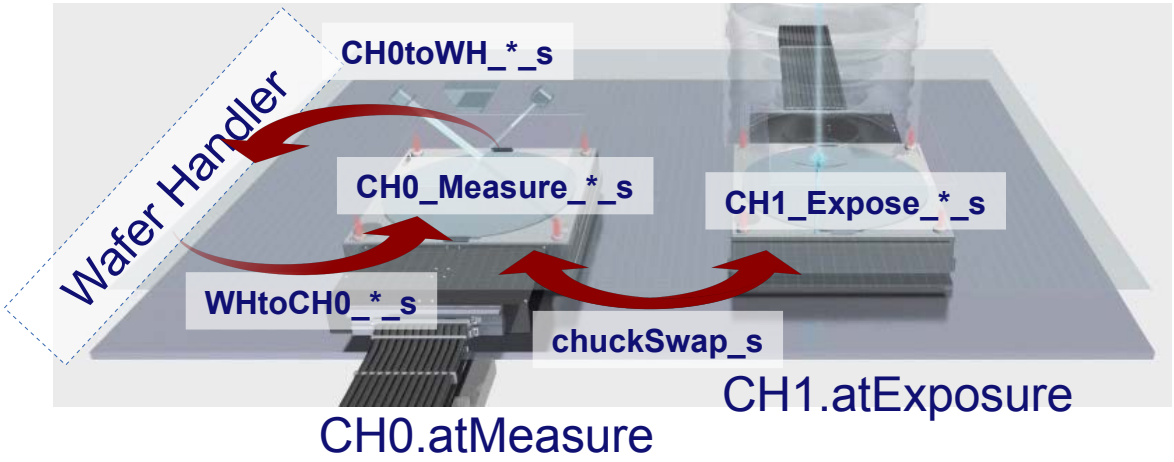
Position and life cycle

Modeling the Wafer Stage (1/3): Chuck Position





Modeling the Wafer Stage (1/3): Chuck Actions



ActionsCH0

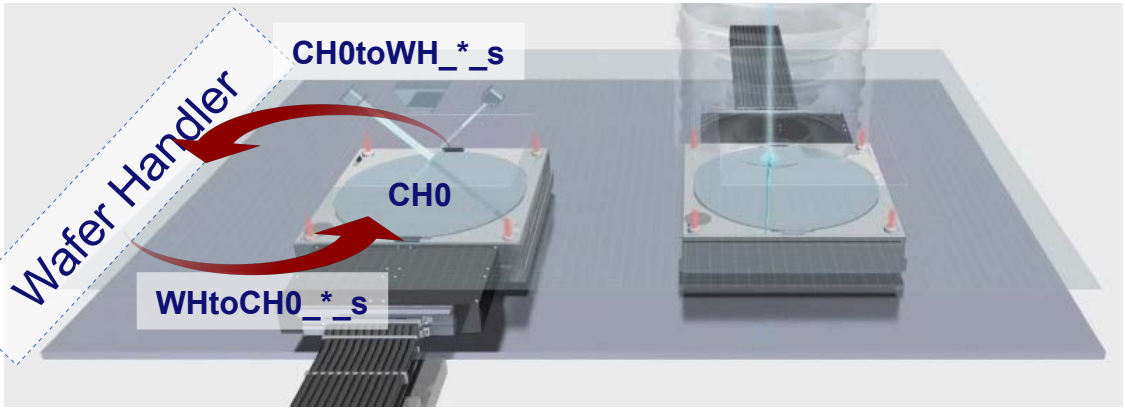
CH0_Measure_*_s,
CH0toWH_*_s,
WHtoCH0_*_s ⇒
PositionCH0.atMeasure
CH0_Expose_*_s ⇒
PositionCH0.atExpose

ActionsCH1

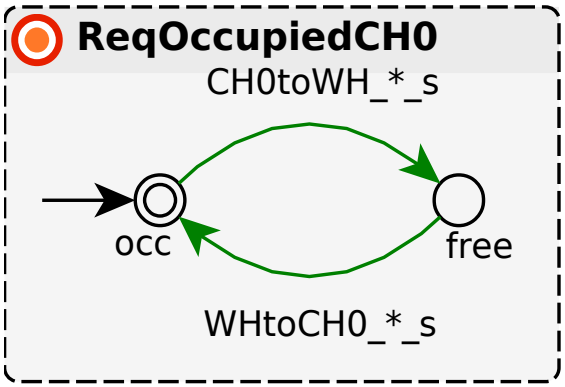
CH1_Measure_*_s,
CH1toWH_*_s,
WHtoCH1_*_s ⇒
PositionCH1.atMeasure
CH1_Expose_*_s ⇒
PositionCH1.atExpose

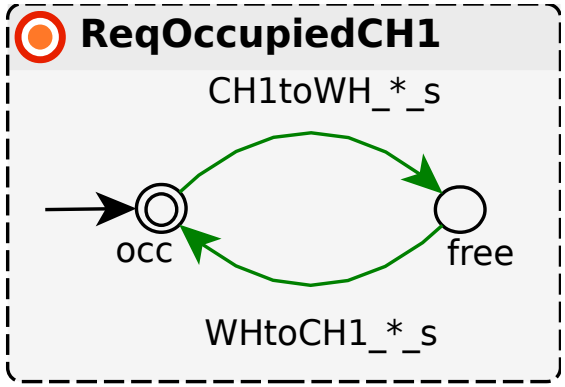
refer to states using variables

Modeling the Wafer Stage (1/3): Chuck Capacity



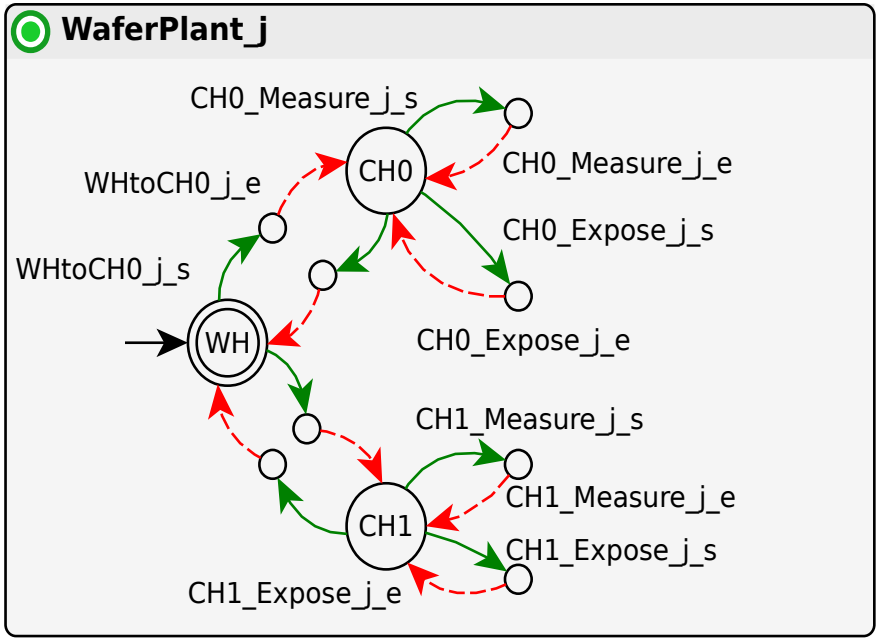
Chuck can hold at most one wafer



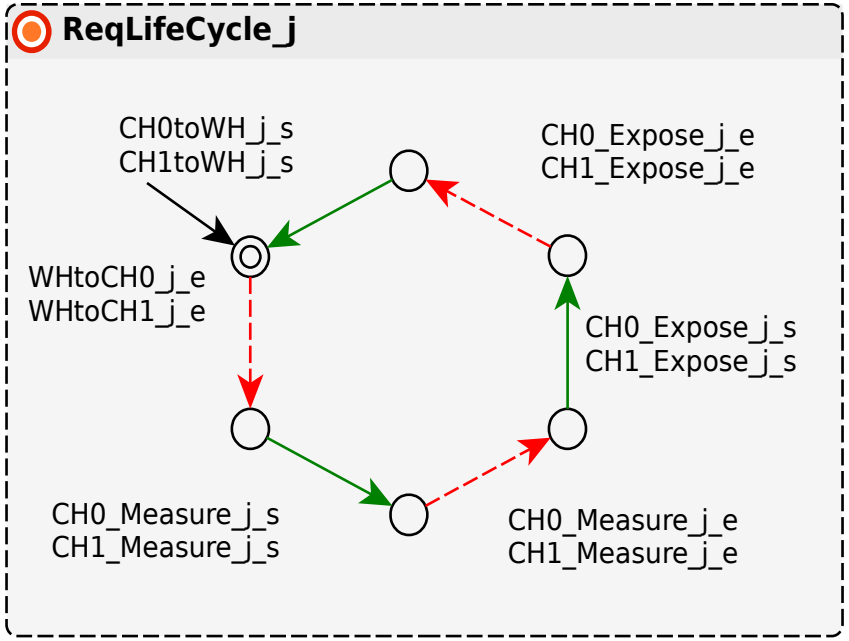


Modeling the Wafer Stage (2/3): Wafers Position and Life Cycle

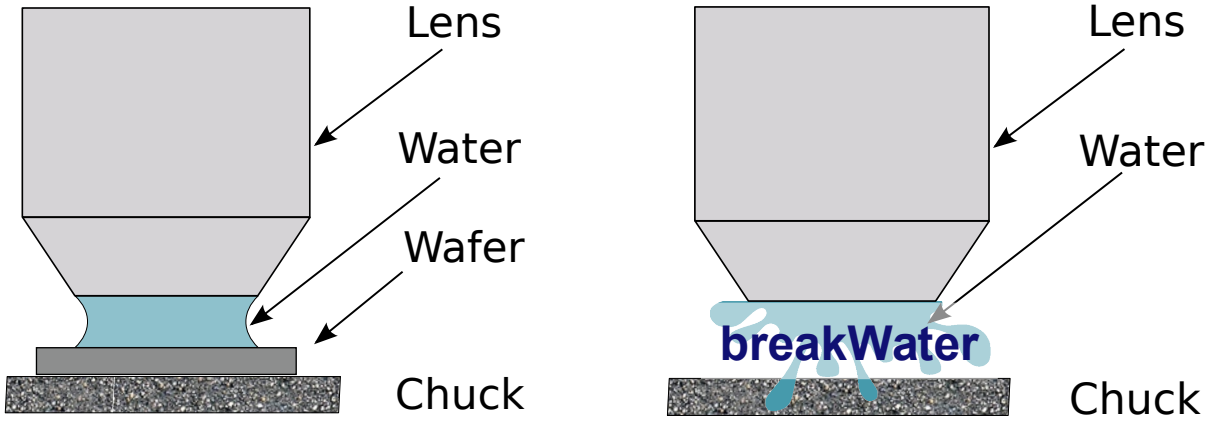
Position of wafer *j*



Life cycle of wafer *j*



Modeling the Wafer Stage (3/3): Water Layer



BreakWater

breakWater ⇒
(PositionCH0.atExpose ∧ StatusCH0.free)
∨ (PositionCH1.atExpose ∧ StatusCH1.free)

When no wafer is present below the lens, the water layer will break

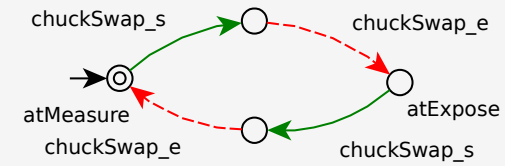
ReqBreakWater

breakWater ⇒
false

Breaking the wafer layer is never allowed

Modeling the Wafer Stage: Complete Specification Model

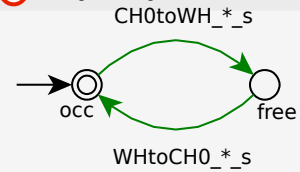
PositionCH0



ActionsCH0

CH0_Measure_*_s,
CH0toWH_*_s,
WHtoCH0_*_s ⇒
PositionCH0.atMeasure
CH0_Expose_*_s ⇒
PositionCH0.atExpose

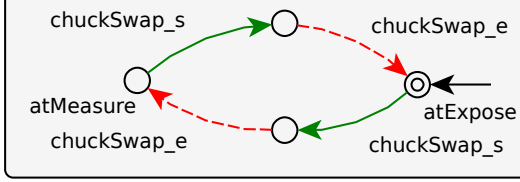
ReqOccupiedCH0



BreakWater

breakWater ⇒
(PositionCH0.atExpose ∧
StatusCH0.free)
∨ (PositionCH1.atExpose ∧
StatusCH1.free)

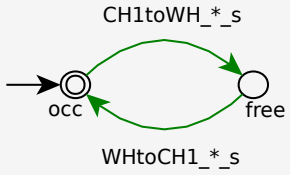
PositionCH1



ActionsCH1

CH1_Measure_*_s,
CH1toWH_*_s,
WHtoCH1_*_s ⇒
PositionCH1.atMeasure
CH1_Expose_*_s ⇒
PositionCH1.atExpose

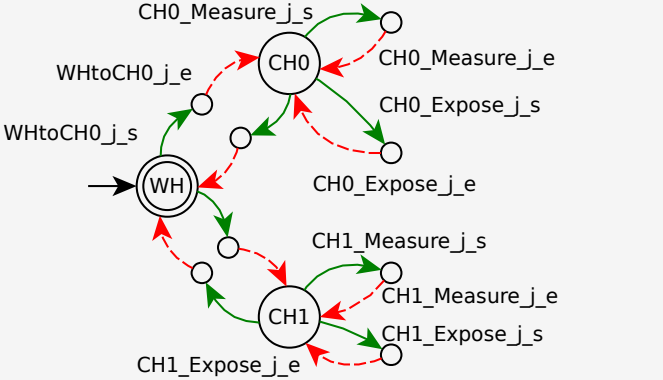
ReqOccupiedCH1



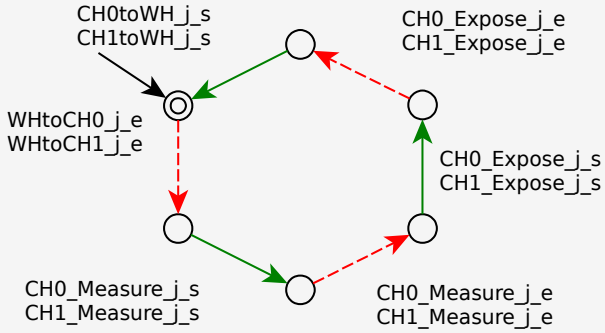
ReqBreakWater

breakWater ⇒
false

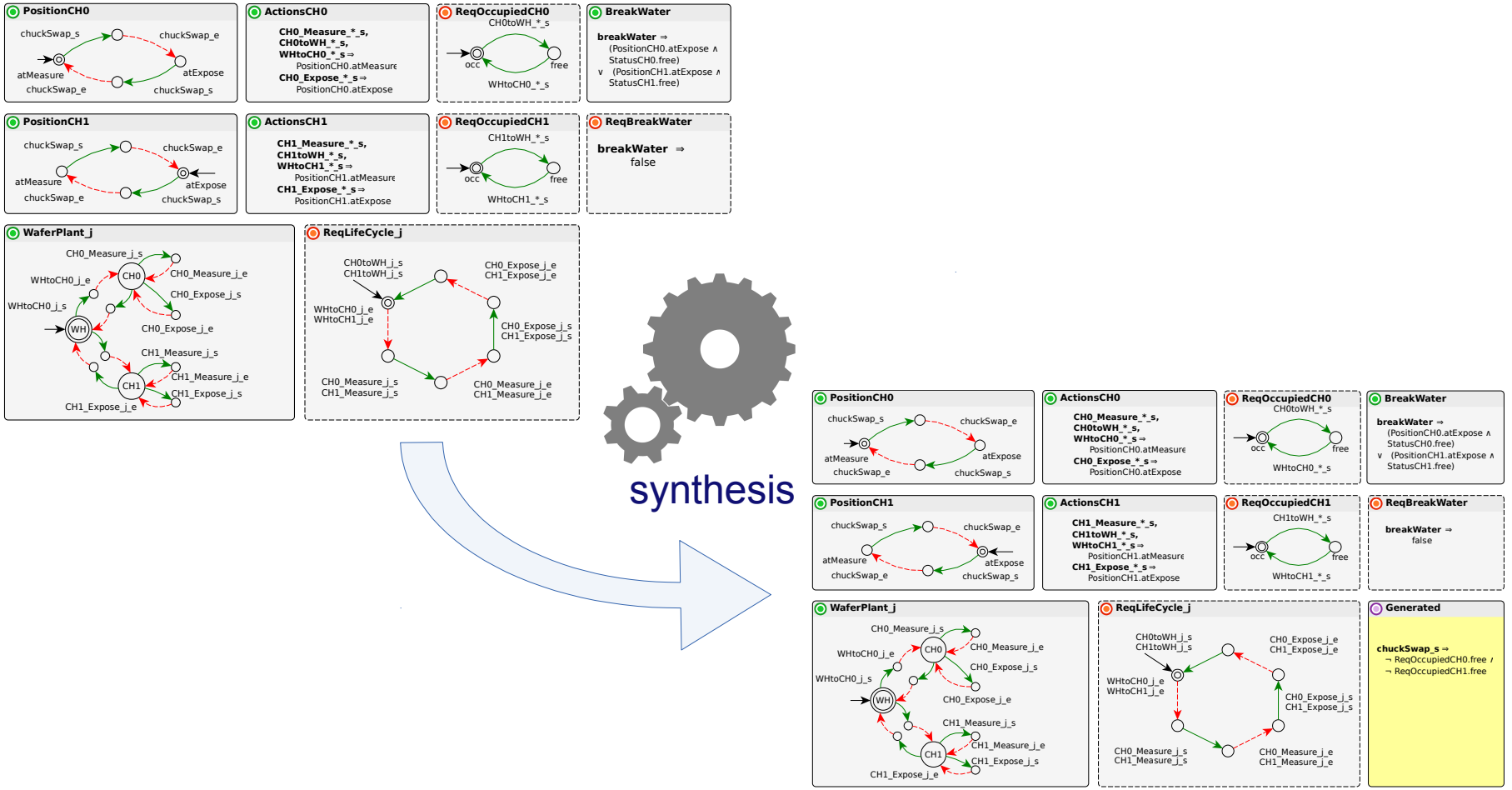
WaferPlant_j



ReqLifeCycle_j

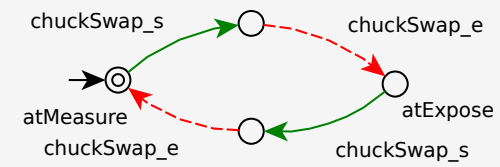


Modeling the Wafer Stage: Apply Synthesis on Specification Model



Modeling the Wafer Stage: Complete Model After Synthesis

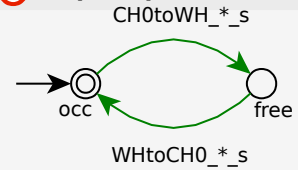
PositionCH0



ActionsCH0

CH0_Measure_*_s,
CH0toWH_*_s,
WHtoCH0_*_s ⇒
PositionCH0.atMeasure
CH0_Expose_*_s ⇒
PositionCH0.atExpose

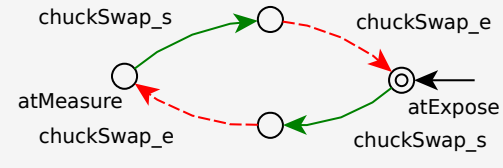
ReqOccupiedCH0



BreakWater

breakWater ⇒
(PositionCH0.atExpose ∧
StatusCH0.free)
∨ (PositionCH1.atExpose ∧
StatusCH1.free)

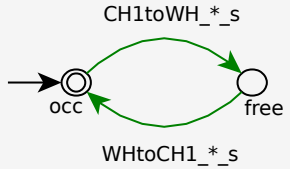
PositionCH1



ActionsCH1

CH1_Measure_*_s,
CH1toWH_*_s,
WHtoCH1_*_s ⇒
PositionCH1.atMeasure
CH1_Expose_*_s ⇒
PositionCH1.atExpose

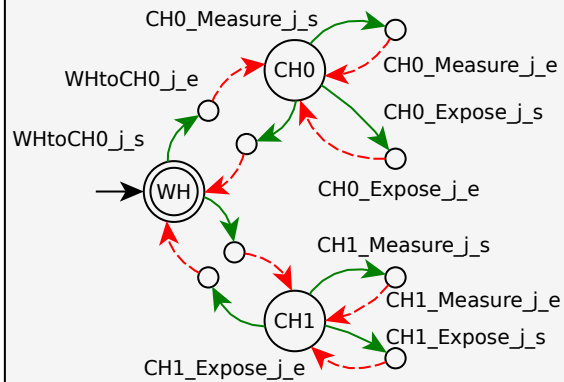
ReqOccupiedCH1



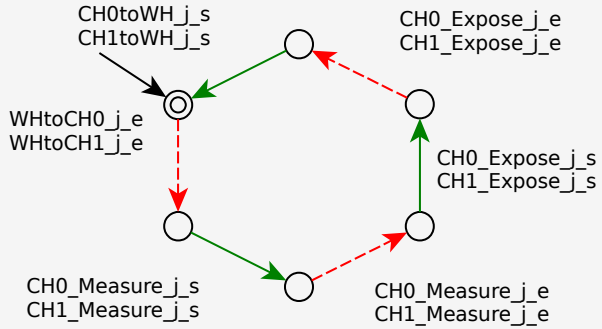
ReqBreakWater

breakWater ⇒
false

WaferPlant_j



ReqLifeCycle_j



Generated

chuckSwap_s ⇒
¬ ReqOccupiedCH0.free ∧
¬ ReqOccupiedCH1.free

Business Case

- **Why does ASML want this?**
  - **Controller design is now done manually:**
    - **Time consuming**
    - **Going from requirements to a supervisory controller design is difficult**
  - **Tight coupling between concepts: hard to make proper decomposition**
  - **High-tech systems evolve rapidly and adapting supervisory controllers take a lot of time**
  - **Exploring impact of changes is hard**

Take-away points and lessons learned

- **Compositional modeling of plant and requirements: divide and conquer.**
- **When requirements are formally part of the specification, requirement traceability is trivial.**
- **Multiparty synchronization provides advantages in terms of modularity and adaptability/maintainability.**

Challenges: road towards industrial adaptation

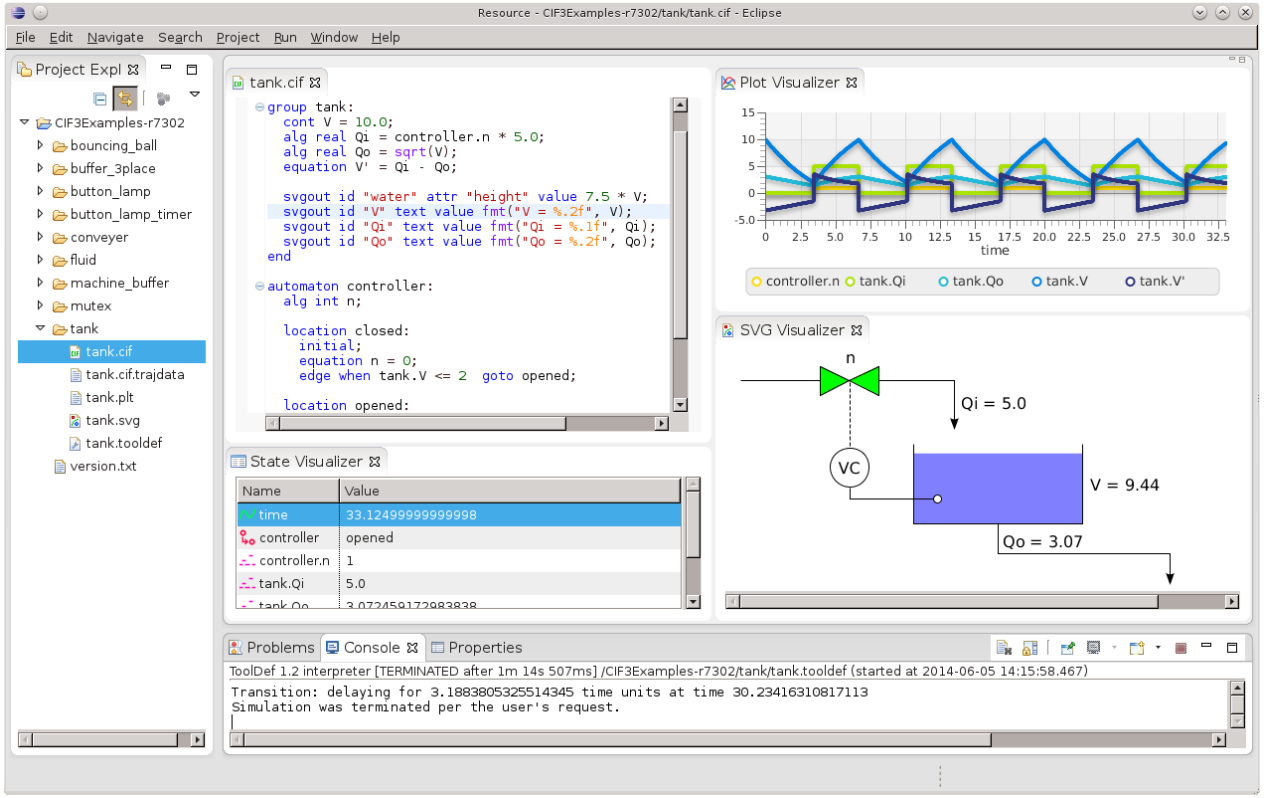
- Synthesis step using monolithic synthesis does not scale well enough
  - Evaluation of modular synthesis techniques
- Completeness: specifying liveness goals, timing requirements
- Performance optimization: maximize throughput, minimize makespan

Available tooling

- **CIF3: http://cif.se.wtb.tue.nl/**
  - **Modeling using (extended) finite (hybrid) automata**
  - **Synthesis of supervisors**
  - **(Graphical) simulation of controlled system with a discrete/hybrid plant**

Available tooling: CIF3

- **CIF3: http://cif.se.wtb.tue.nl**



bramvandersanden.com/models2015

b.v.d.sanden@tue.nl