Supervisory control of a patient support table

Rolf Theunissen
Ramon Schiffelers
Bert van Beek
Koos Rooda

May 28, 2008

Embedded Systems INSTITUTE

TU/e Technische Universiteit Eindhoven University of Technology

Outline

Introduction

Supervisory control

Case: Patient support system

Implementation

Conclusions

Rationale

Trends in current high-tech multidisciplinary system development:

- Increasing number of features
- Increasing complexity
- Decreasing time-to-market

Resulting effects:

- Decreasing quality
- Increasing effort
- Increasing costs

Darwin project

Objective: to develop architectures, methods and tools for optimizing system evolvability.

Result: faster time to market for product iterations whilst maximizing technology reuse.

Partners:

- Philips Healthcare – MR division
- Embedded Systems Institute (ESI)
- Philips Research
- 5 Universities

Introduction

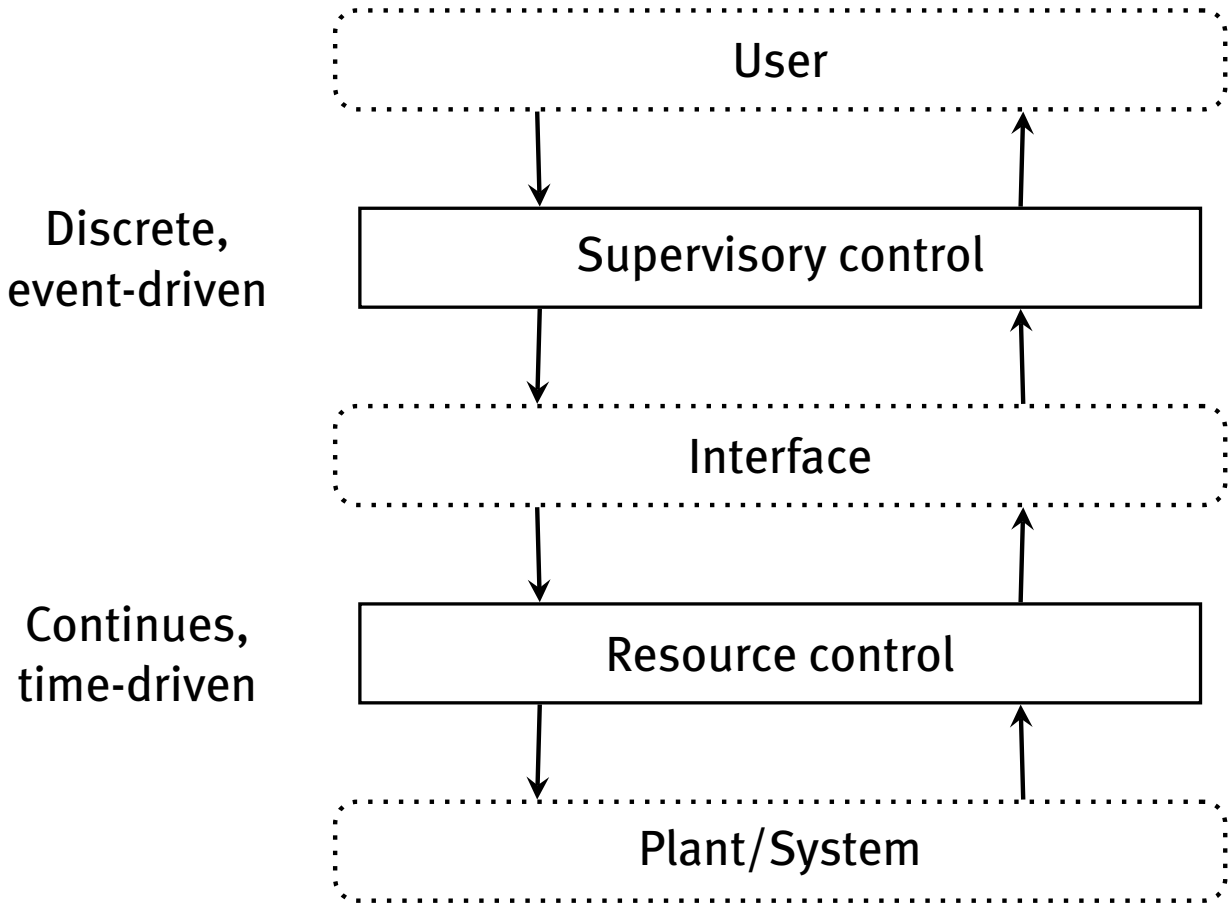
Supervisory control

Case: Patient support system

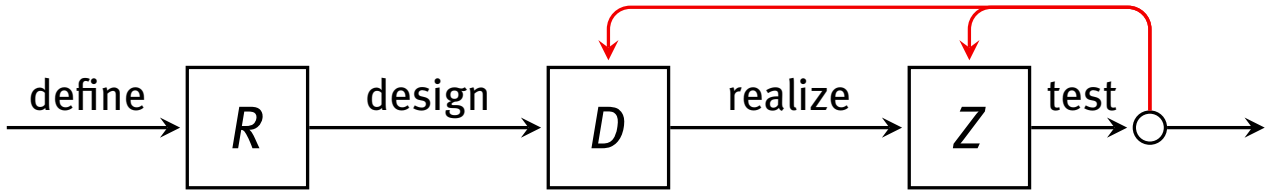
Implementation

Conclusions

Supervisory control

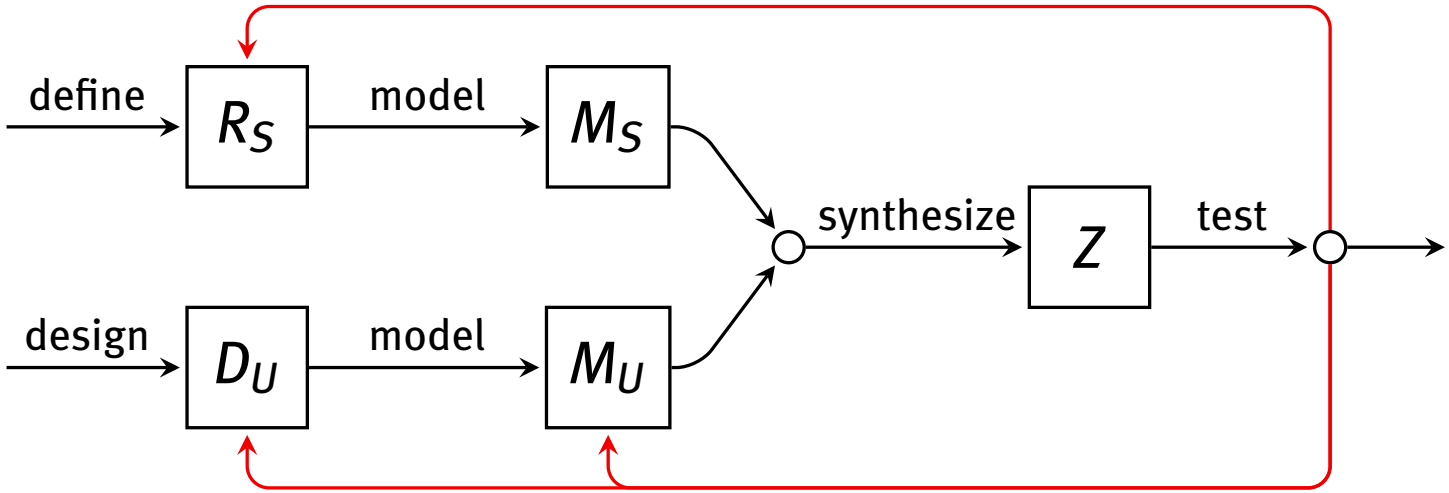


Current supervisory control design



- Define requirements
- Design the supervisor by hand
- Realize the supervisor by hand
- Test whether the system meet the requirements
- Iterate (change design and/or implementation)

Supervisory control synthesis



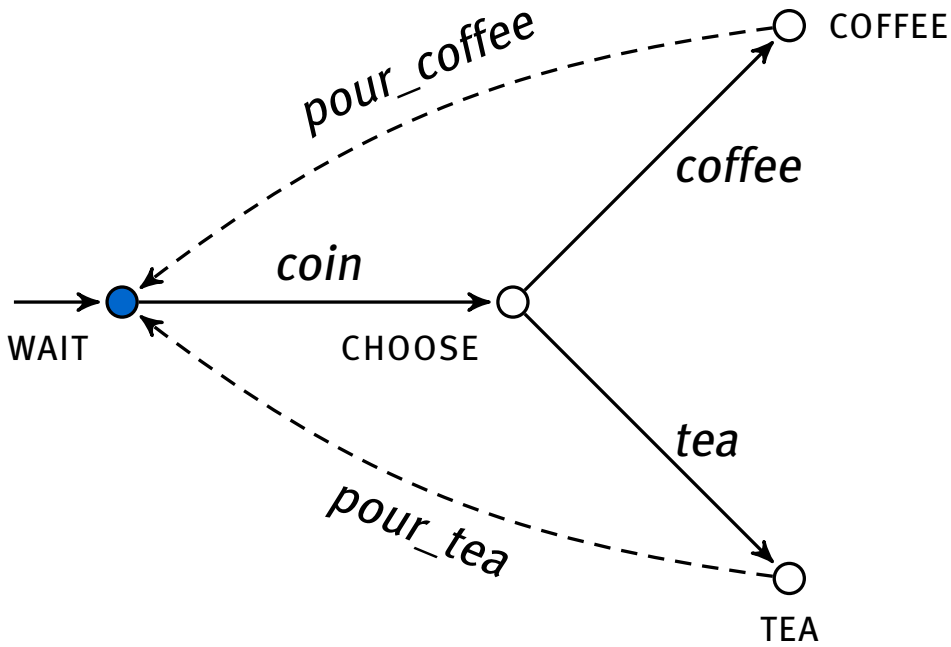
- Model the uncontrolled system
- Define control requirements
- Model/formalise the control requirements
- Synthesize (generate) the supervisor implementation
- Test whether the system meets the requirements
- Iterate (change control requirements and/or uncontrolled system)

Wonham Supervisory Control Theory

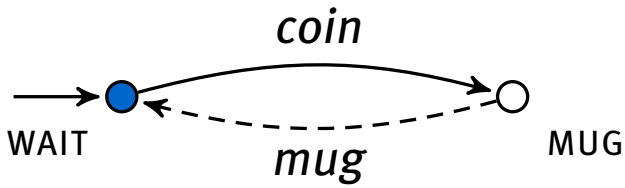
- Behaviour of a plant is a sequence of events (formal language)
- Desired behaviour (specification) is described as allowed order of events:
  - safety
  - liveness (deadlock free)
- Distinction between two types of events:
  - controllable events
  - uncontrollable events
- Generates an controller which is correct by contstruction:
  - within safety specification
  - controllable
  - deadlock free

Example: Coffee Machine

After a coin is inserted a choice can be made for coffee or tea. Then the machine pours either coffee or tea:

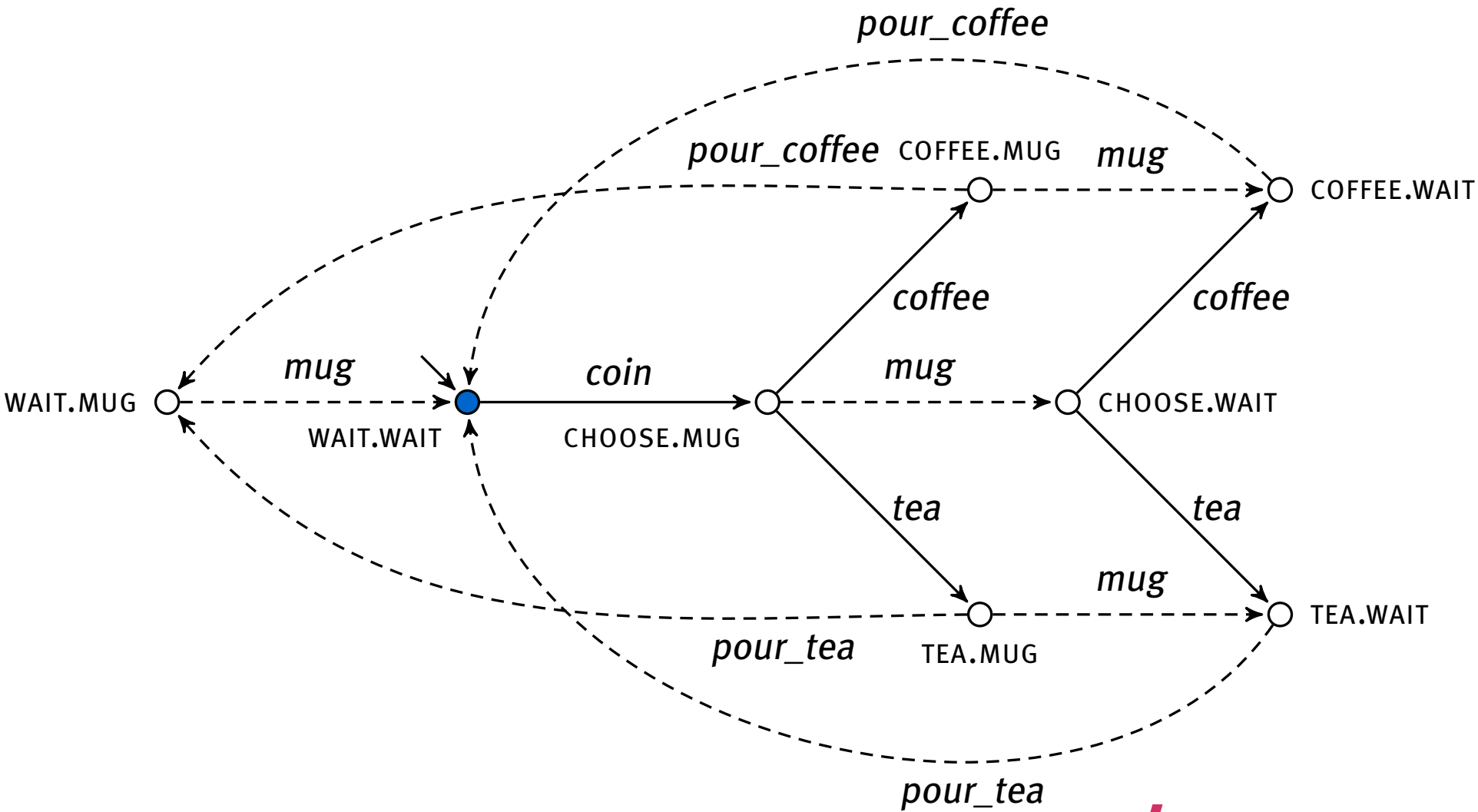


After a coin is inserted, the machine dispenses a mug:



Example: Coffee Machine

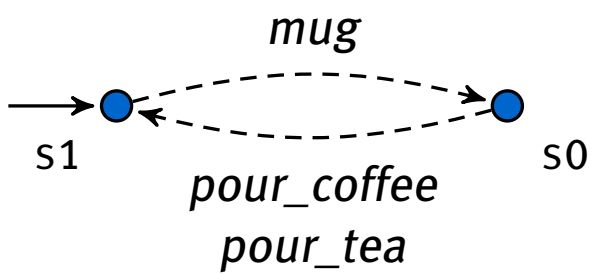
Total machine behaviour:



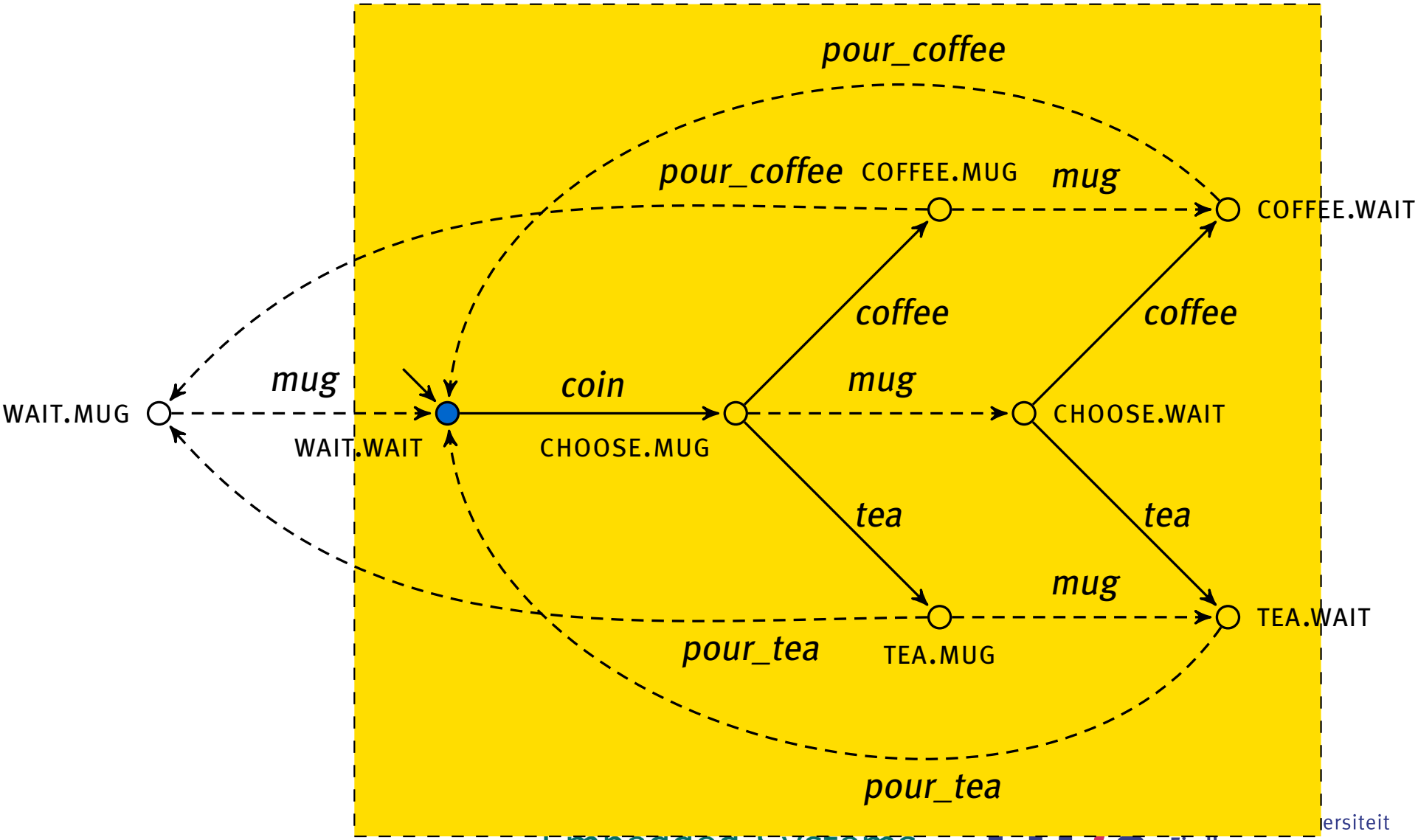
Example: Coffee Machine

Control requirement:

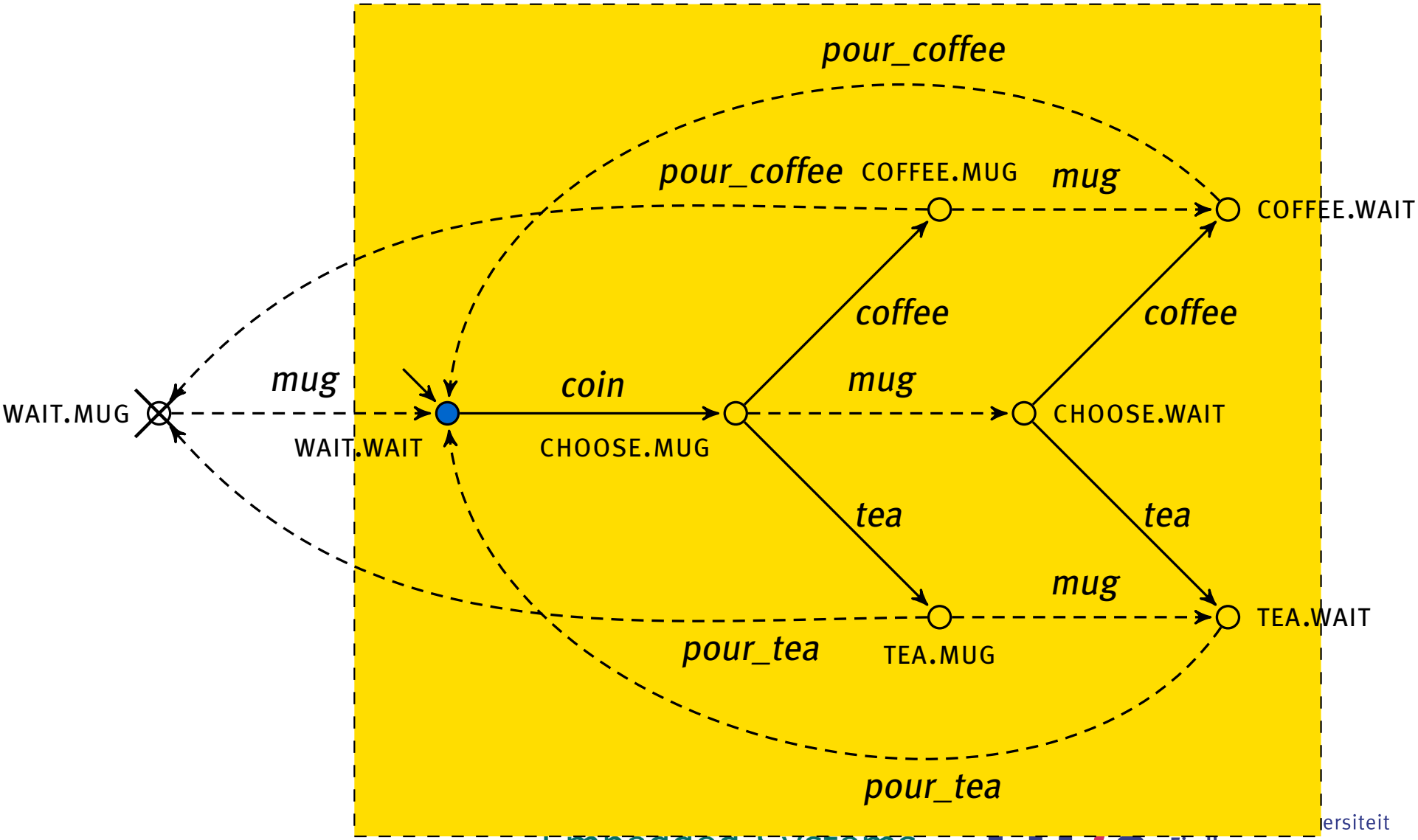
- Prevent coffee or tea being poured before a mug is dispensed.



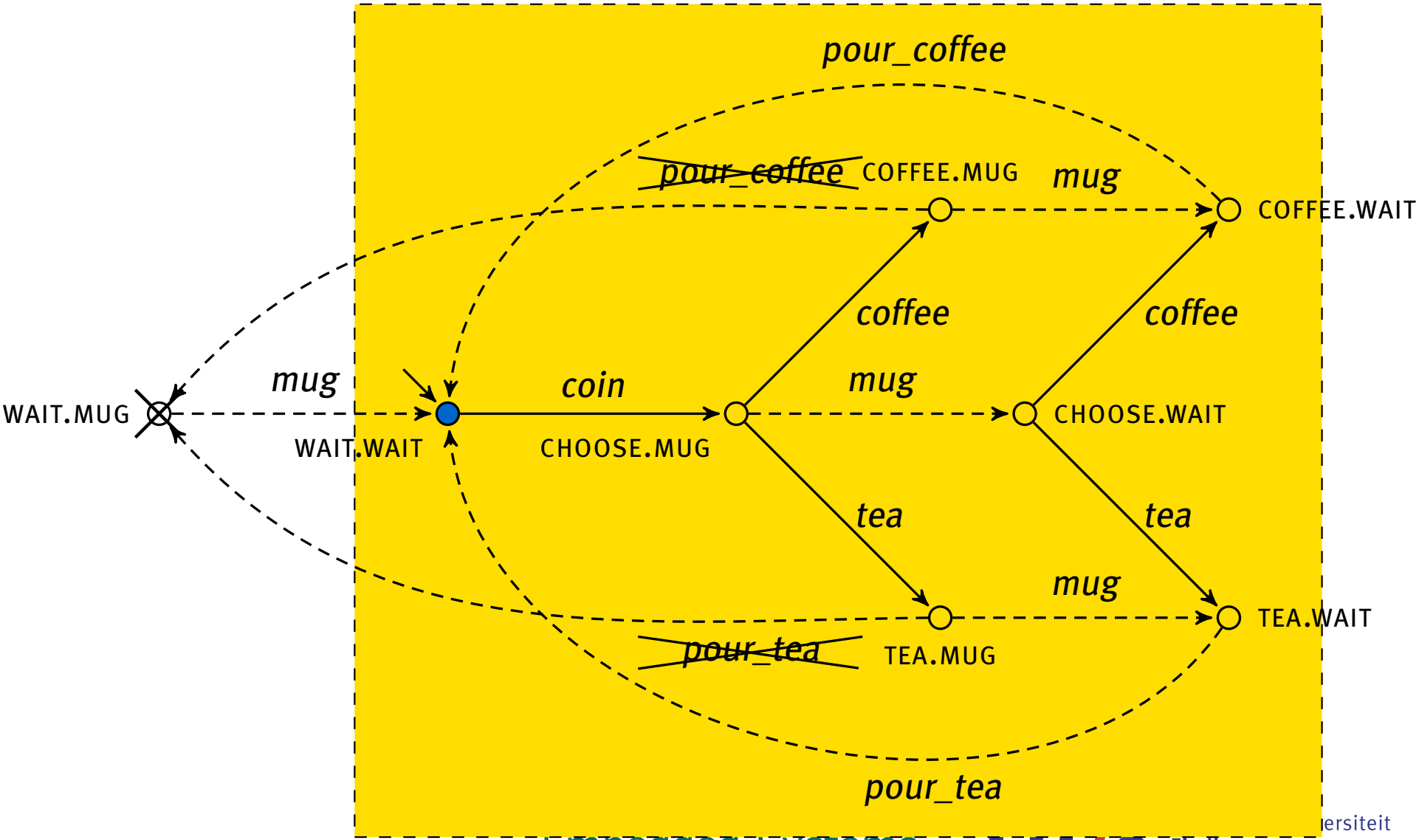
Example: Coffee Machine



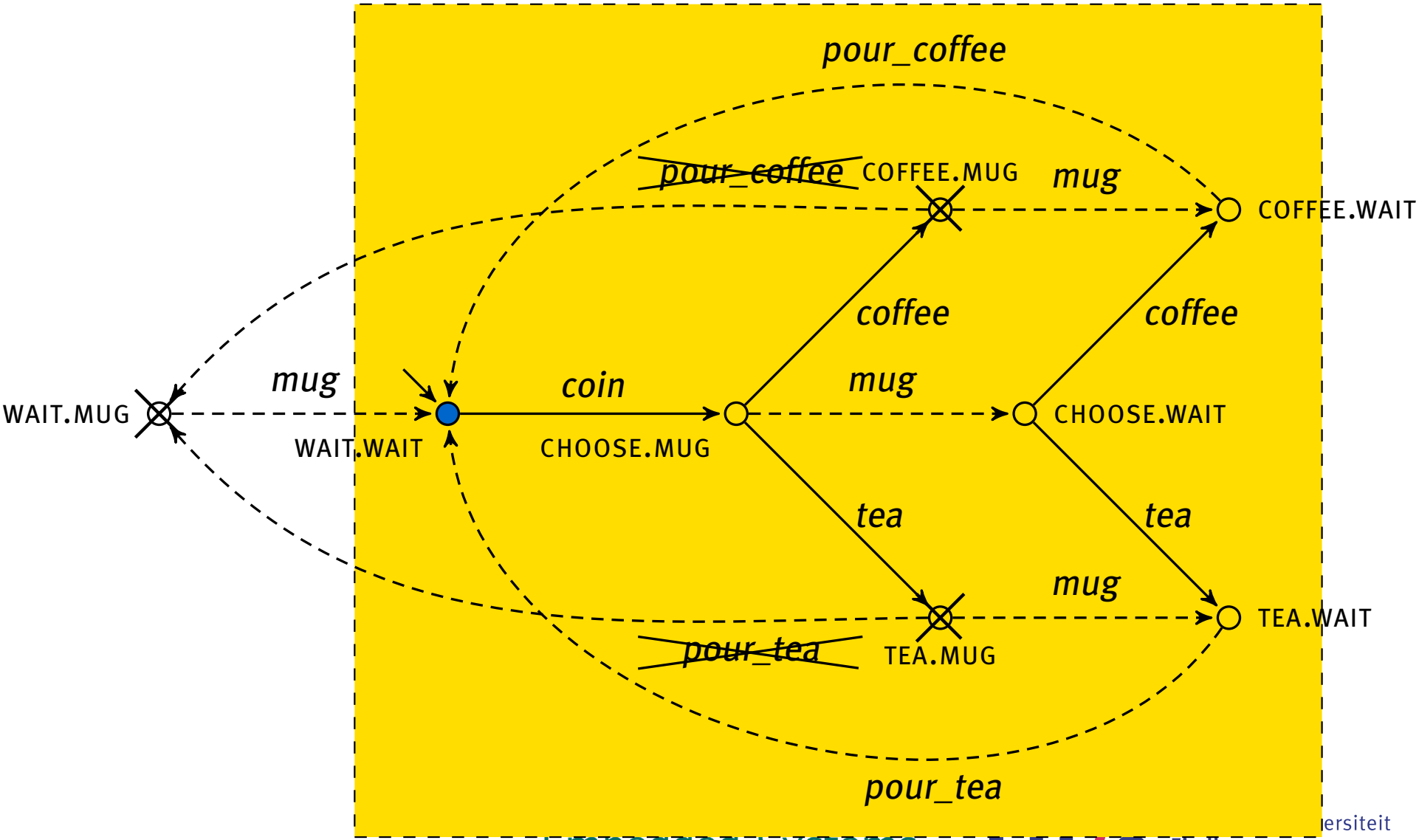
Example: Coffee Machine



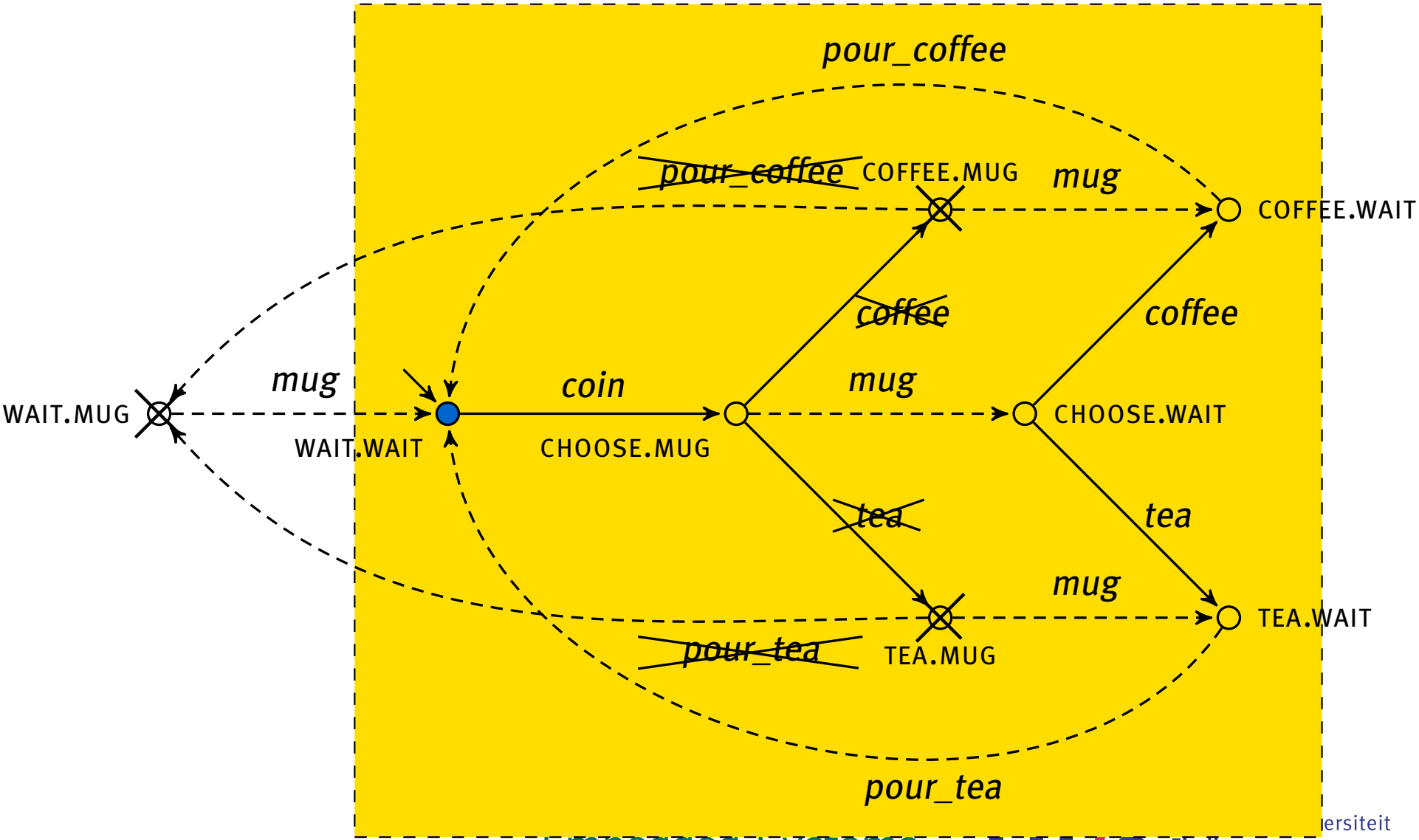
Example: Coffee Machine



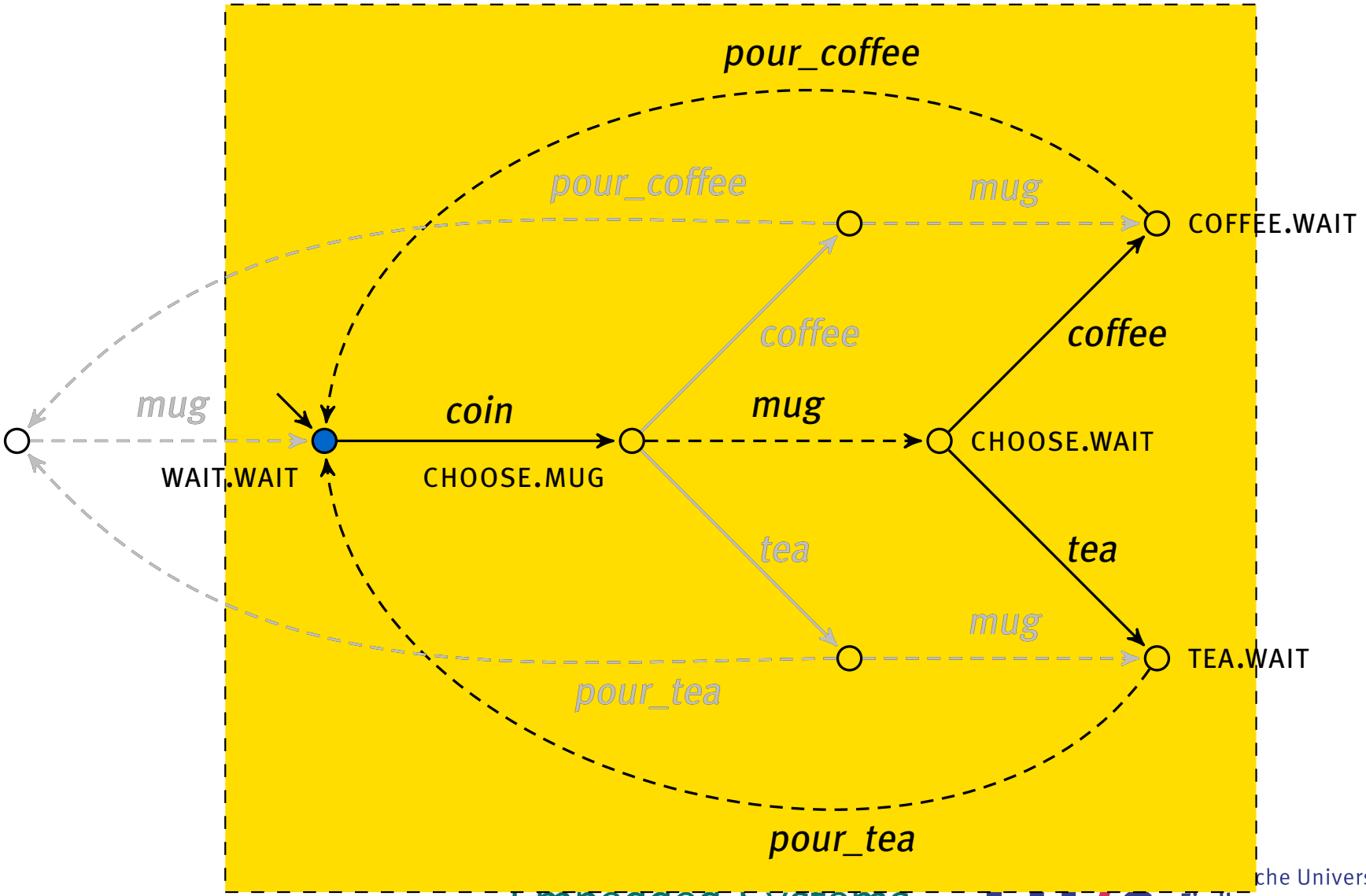
Example: Coffee Machine



Example: Coffee Machine



Example: Coffee Machine



Introduction

Supervisory control

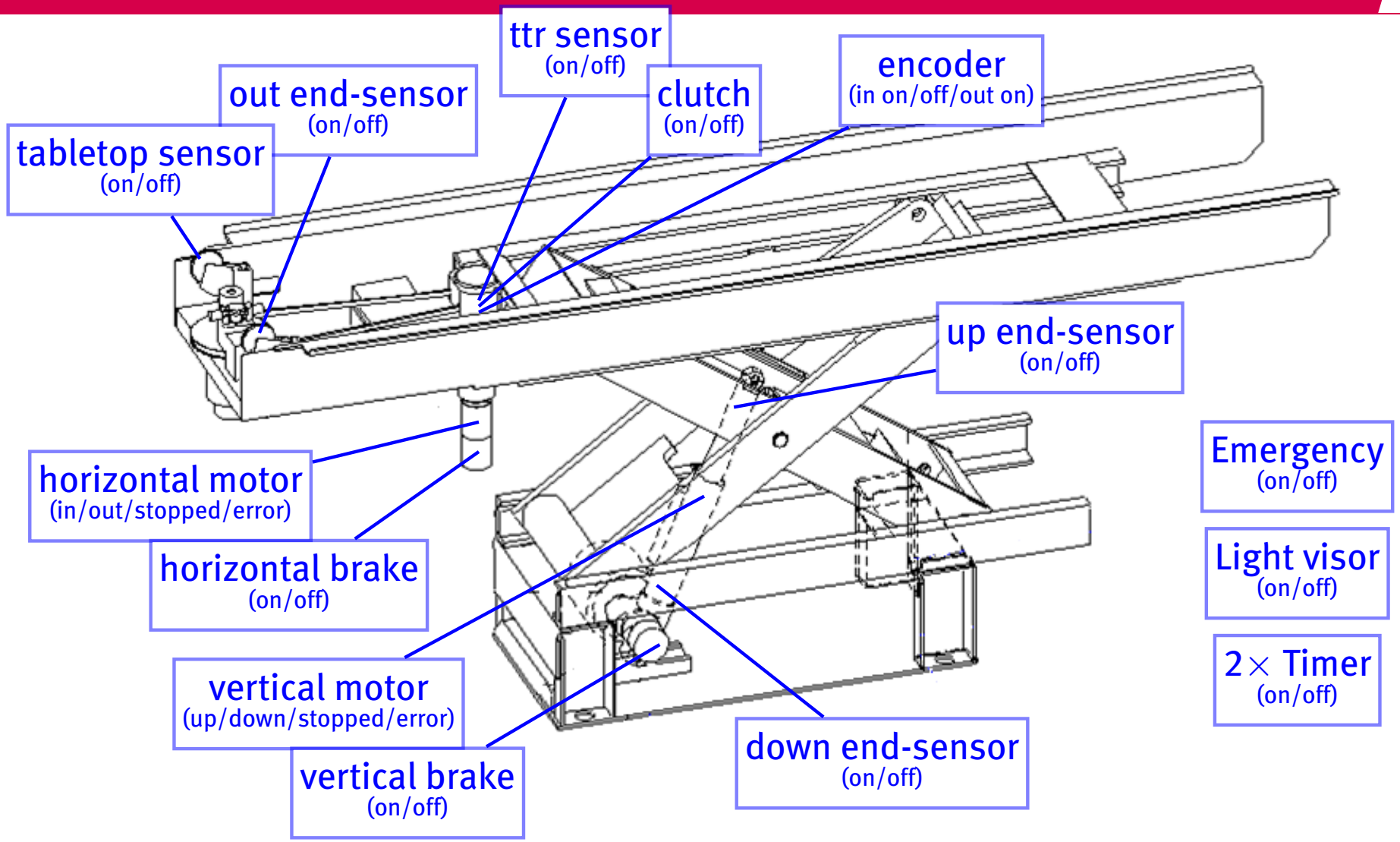
Case: Patient support system

Implementation

Conclusions

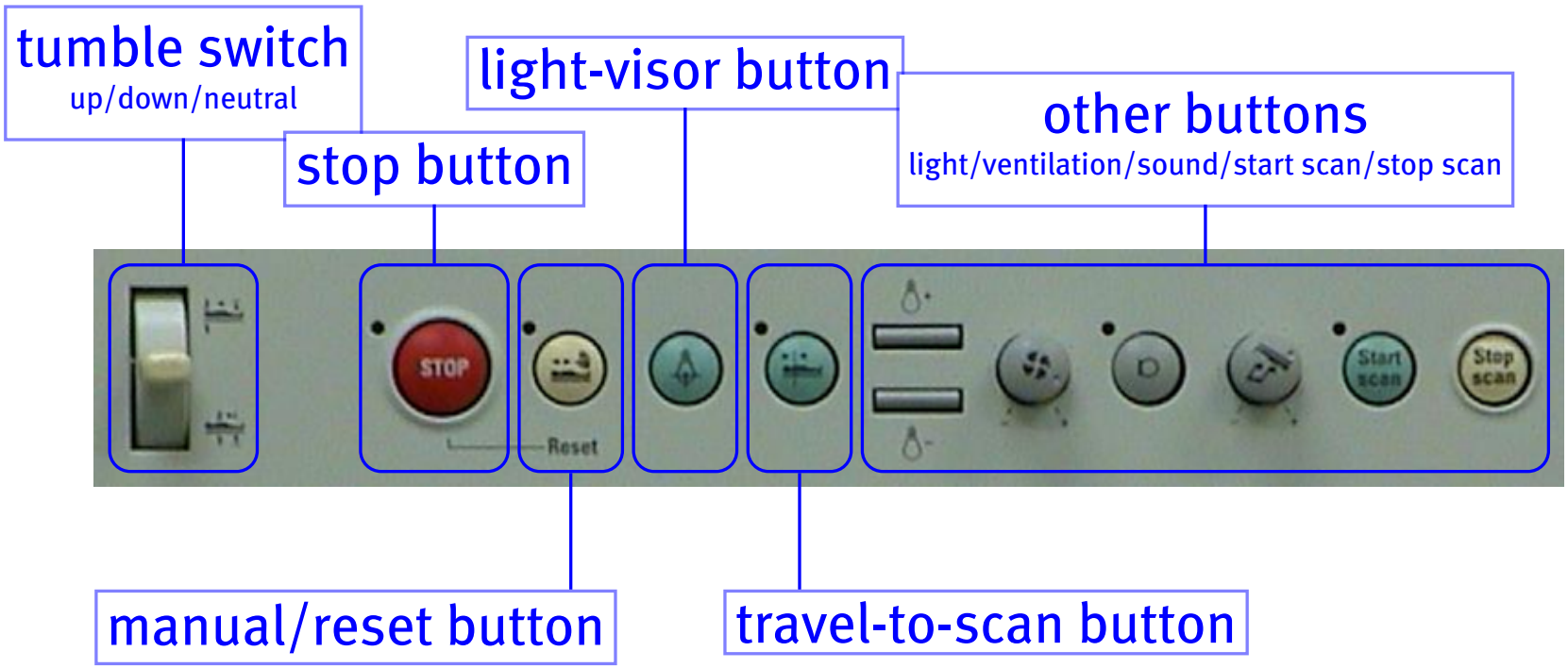
Case: Patient support system

Case: Patient support system



⟹ Total 1.7 million states and 64 million transitions for the uncontrolled system.

Case: Patient support system



Case: Patient support system, Goal

Desired system design process:

- Proven correct w.r.t. control requirements
- Proven non-blocking behaviour
- Easier to adapt to a new environment
  - New user interface
  - Other hardware
- Less effort to add new features

Supervisory control synthesis and evolvability

Re-synthesize the supervisor when:

- the control requirements change
- the uncontrolled system changes

Decompose the system and requirements into *small*, *loosely coupled* and *minimal restrictive* models.

- Small:
  - easier to understand
  - easier to modify
- Loosely coupled:
  - no entangling of requirements
  - independent modifications
- Minimally restrictive:
  - maximal freedom for future modifcations/extentions

Modeling uncontrolled system

Components modelled with 27 small automata:

- Horizontal
- Vertical
- Travel-to-scan and light-visor
- Emergency
- User interface buttons
- User interface LED's

Total state-space uncontrolled system: 870 million states

Modeling the control requirements

55 control small, loosely coupled, requirements automata:

- Do not move beyond end sensors
- Only motorized movement if clutch is active
- No motorized movement if Table-Top-Release active
- Only move vertically if horizontal in maximal out position
- Tumble switch moves table up and down, or in and out
- ...

Introduction

Supervisory control

Case: Patient support system

Implementation

Conclusions

State space explosion

- Resulting supervisor is to big to implement in real hardware
- The problem is to big to solve all at once

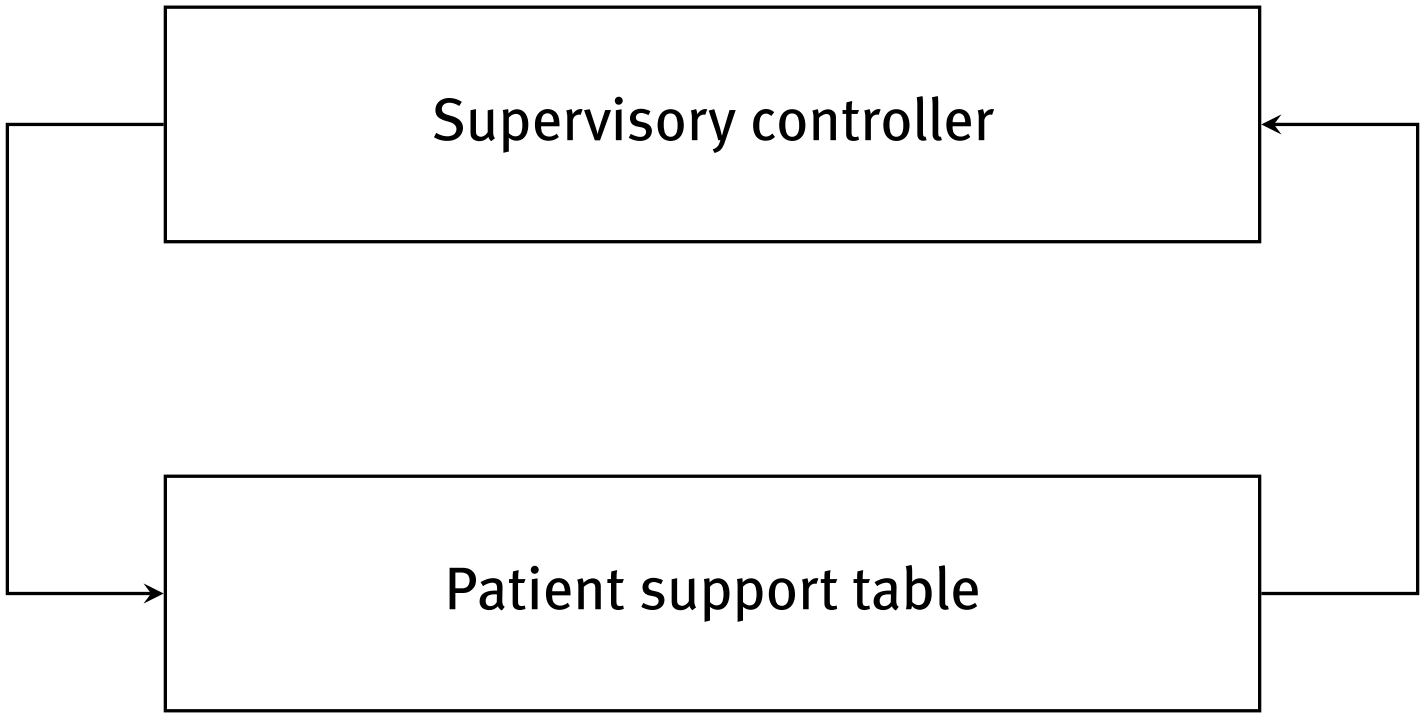
State space explosion

- Resulting supervisor is to big to implement in real hardware
- The problem is to big to solve all at once

Construct the supervisor in a modular way:

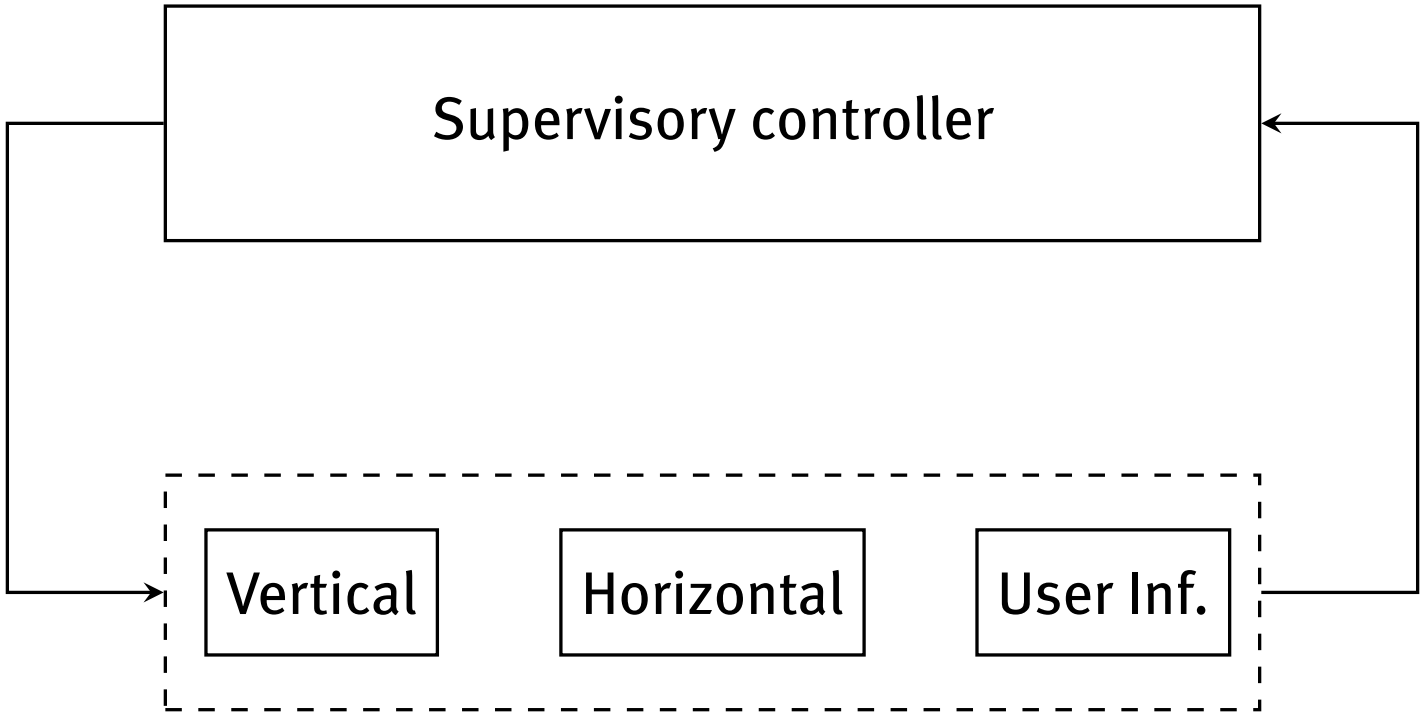
- Exploit locality of the control requirements
- Create abstractions of the plant models

Modular supervisors



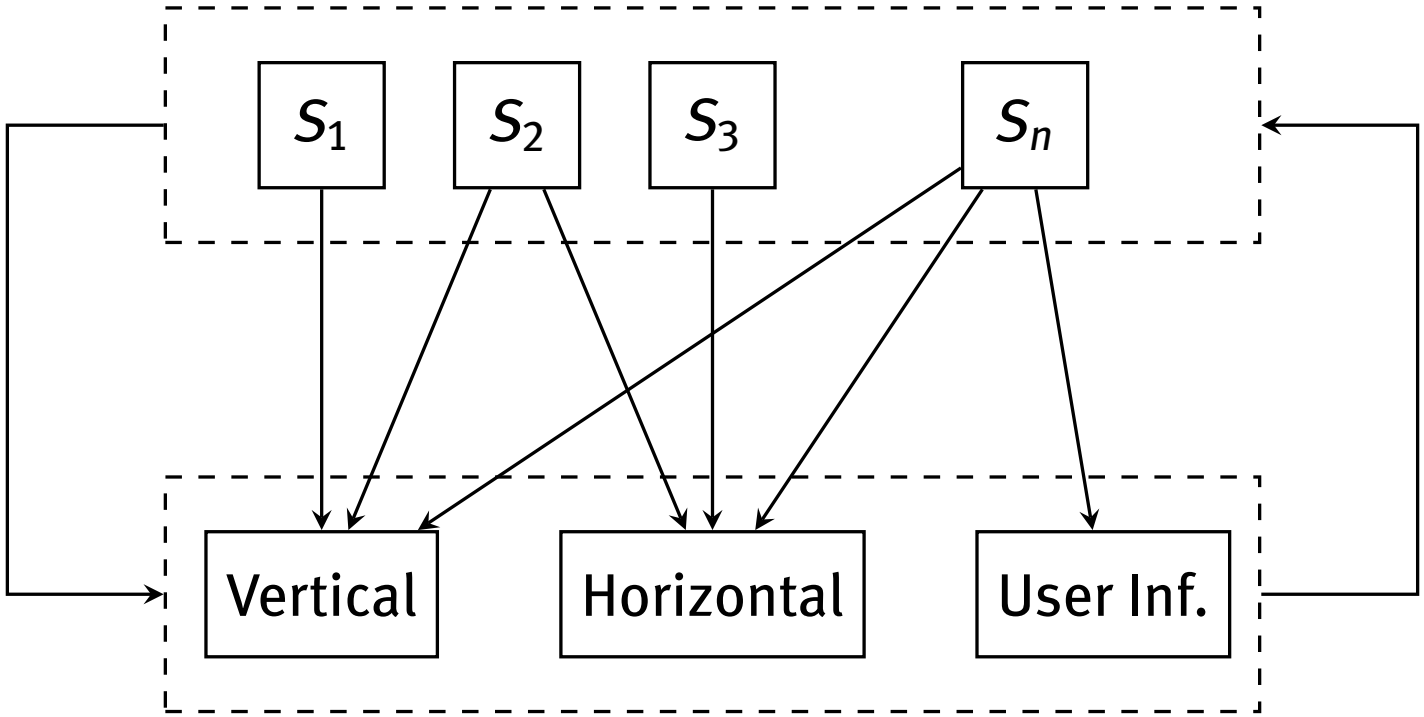
Modular supervisors

The patient support system consists of multiple components:



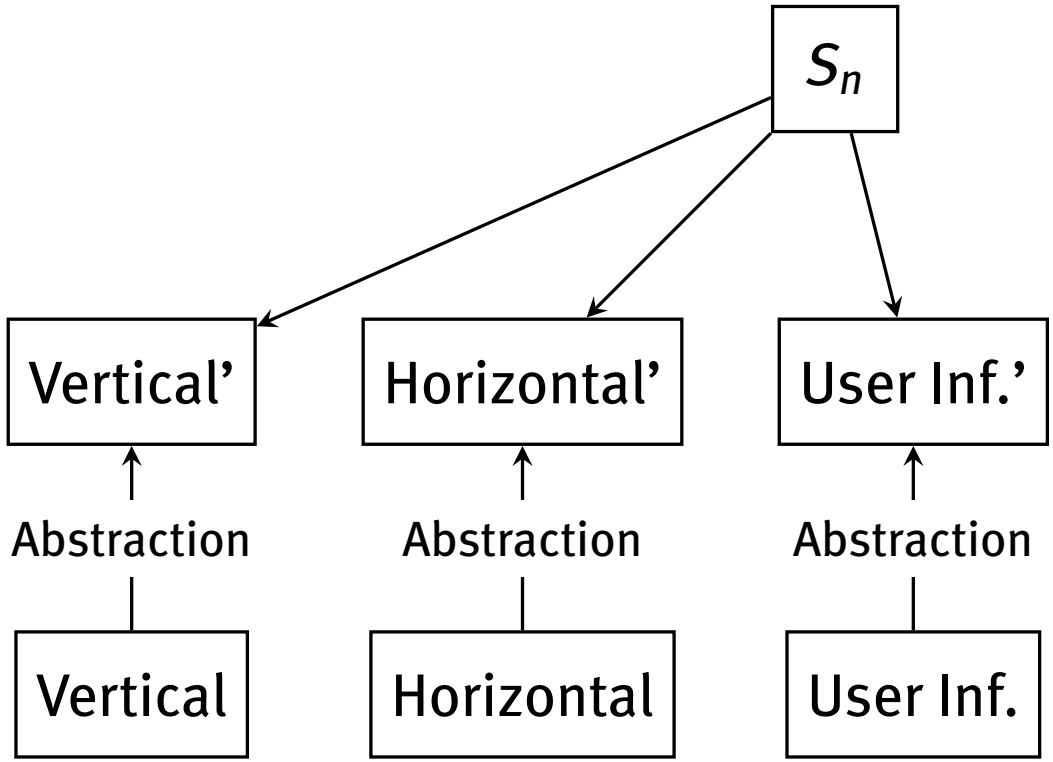
Modular supervisors

Divide the control requirements over the components they belong to, and generate multiple supervisors to implement these requirements:



Plant abstractions

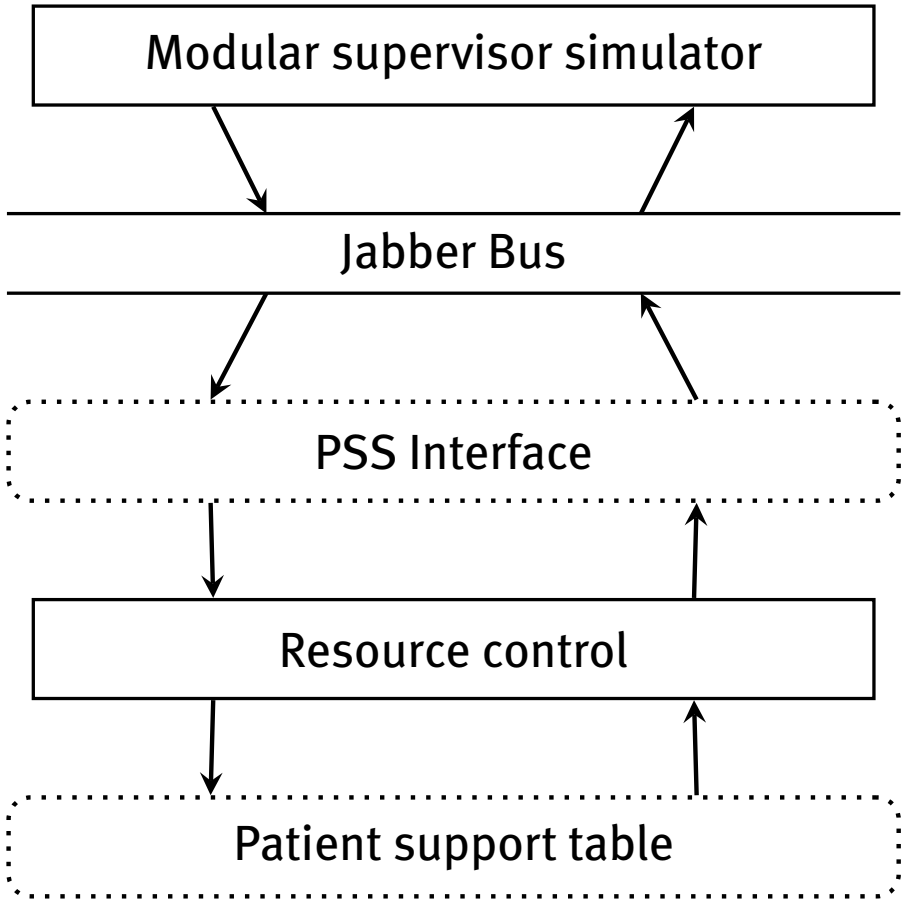
Hide all events which are not in the control requirements implemented by supervisor S_n:



Modular supervisor with abstractions

- Divided the 55 control requirements over 17 supervisors
- All supervisors relatively small
- Automatic abstractions
- Automatic compatibility check of the supervisors
- Globally non-blocking

Implementation on real hardware



Future work

- Extensions to the patient support case:
  - Host behavior (interface)
  - Bad weather behavior
  - Change requirements
- The implications of a-synchronous communication

Conclusion

- Unambiguous specification of the uncontrolled system
  - Small and loosely coupled models

    $\Longrightarrow$ Evolvable models
- Unambiguous specification of the control requirements
  - Small, loosely coupled and minimal restrictive models

    $\Longrightarrow$ Evolvable models
- Generated correct modular supervisor:
  - 17 supervisors
  - deadlock free
  - automated abstractions
- Hardware-in-the-loop simulation with the real hardware
- Changed specification within hours

Supervisory control of a patient support table

Rolf Theunissen
Ramon Schiffelers
Bert van Beek
Koos Rooda

May 28, 2008

Embedded Systems INSTITUTE

TU/e Technische Universiteit Eindhoven University of Technology