Embedded Systems INSTITUTE

PHILIPS

# Supervisory control synthesis for a patient support table

Rolf Theunissen, Ramon Schiffelers,
Bert van Beek, Koos Rooda

Systems Engineering Group
Dept. of Mechanical Engineering

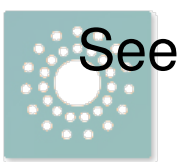
TU/e Technische Universiteit Eindhoven University of Technology

December 4, 2008

# Background

The work presented is carried out in the Darwin project:

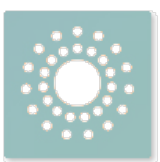
- **Objective**
  Develop architectures, methods and tools for optimizing system evolvability. i.e. the ability of a system to evolve easily in the face of changing requirements.
- **Industrial case**
  MRI scanners: complex systems, about $10^7$ lines of code
- **Organization**
  - *Academic partners* Delft University of Technology, Eindhoven University of Technology, University of Groningen (RuG), University of Twente, and the Vrije Universiteit Amsterdam
  - *Industrial partners* Philips Healthcare, Philips Research
  - *Project Management* Embedded Systems Institute (ESI)



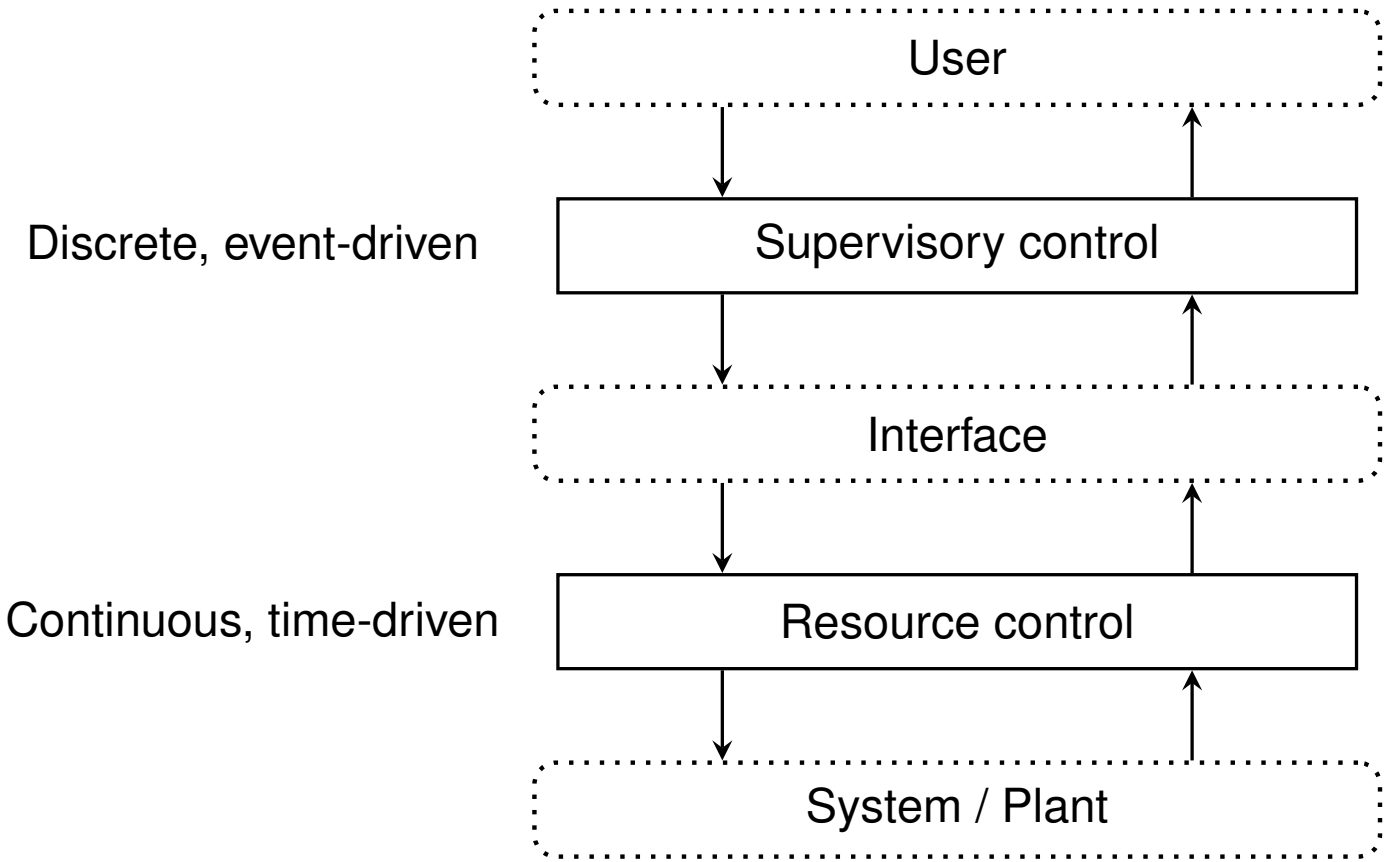
See http://www.esi.nl/projects/darwin

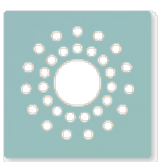
# Outline

- Supervisory control
- Patient support system
- Supervisory control design
  - Conventional
  - Model-based Engineering
  - Synthesis
- Concluding remarks



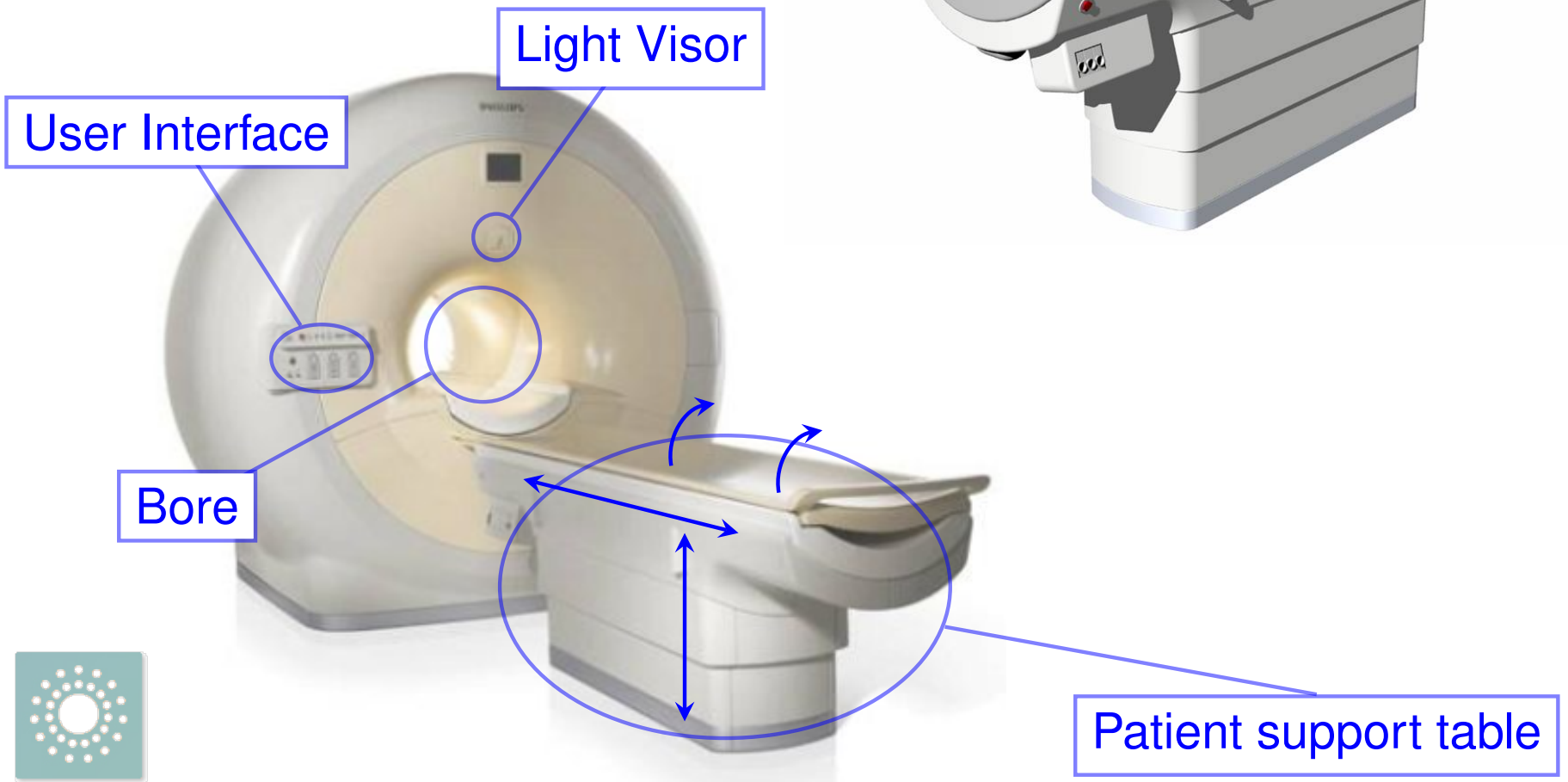
# Supervisory control



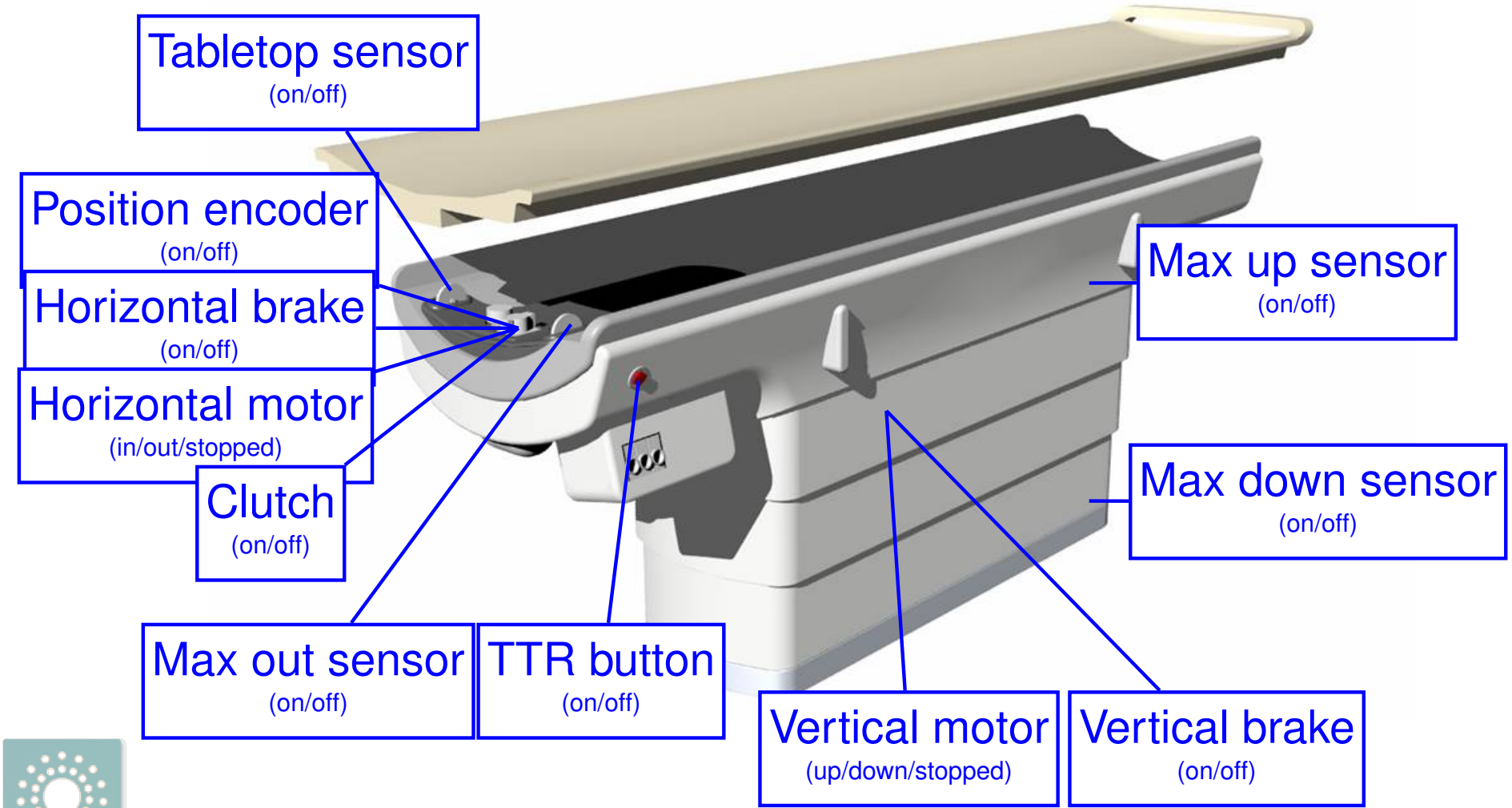
# Patient support system

## Overview



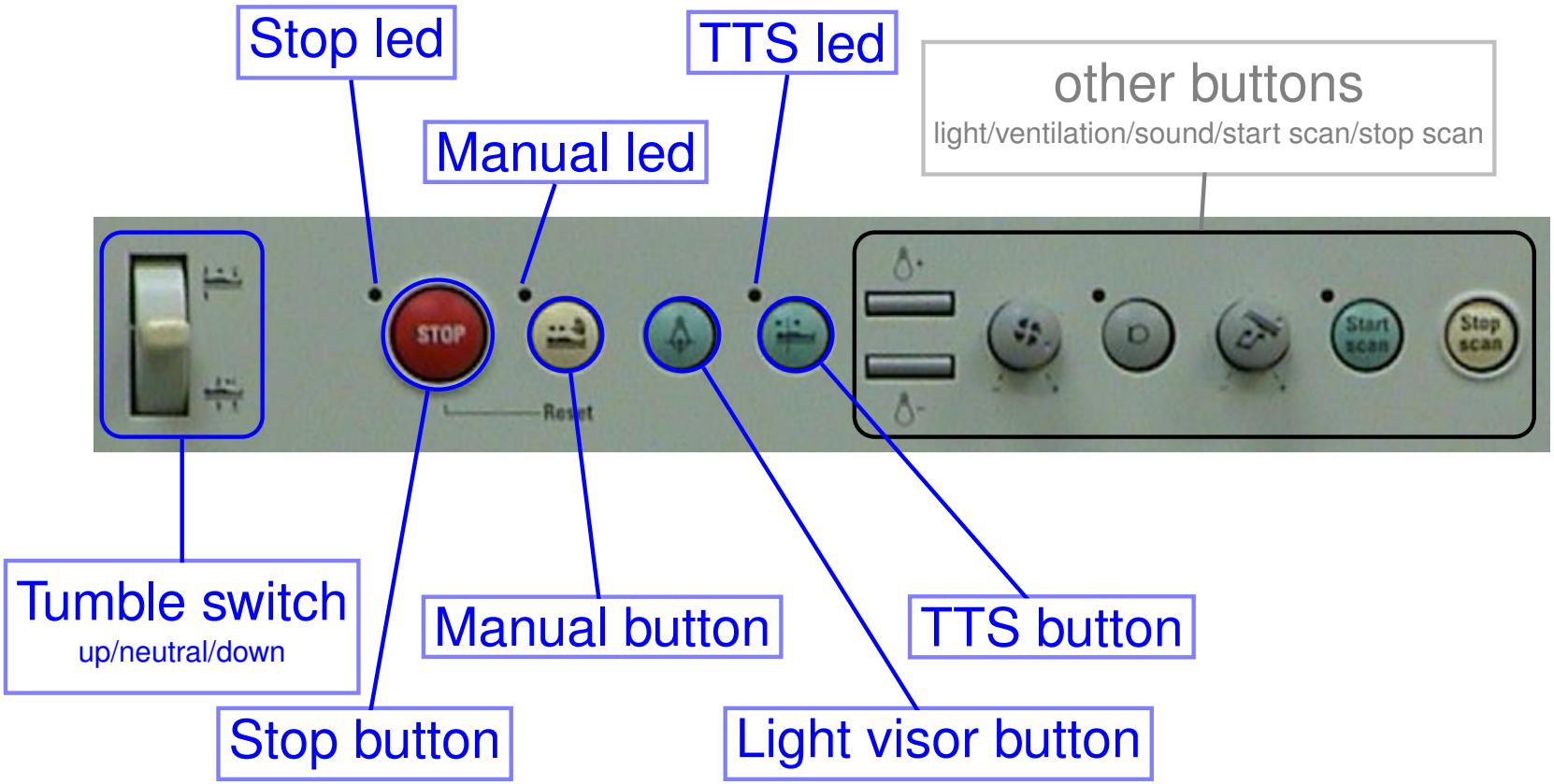
# Patient support system

## Patient table



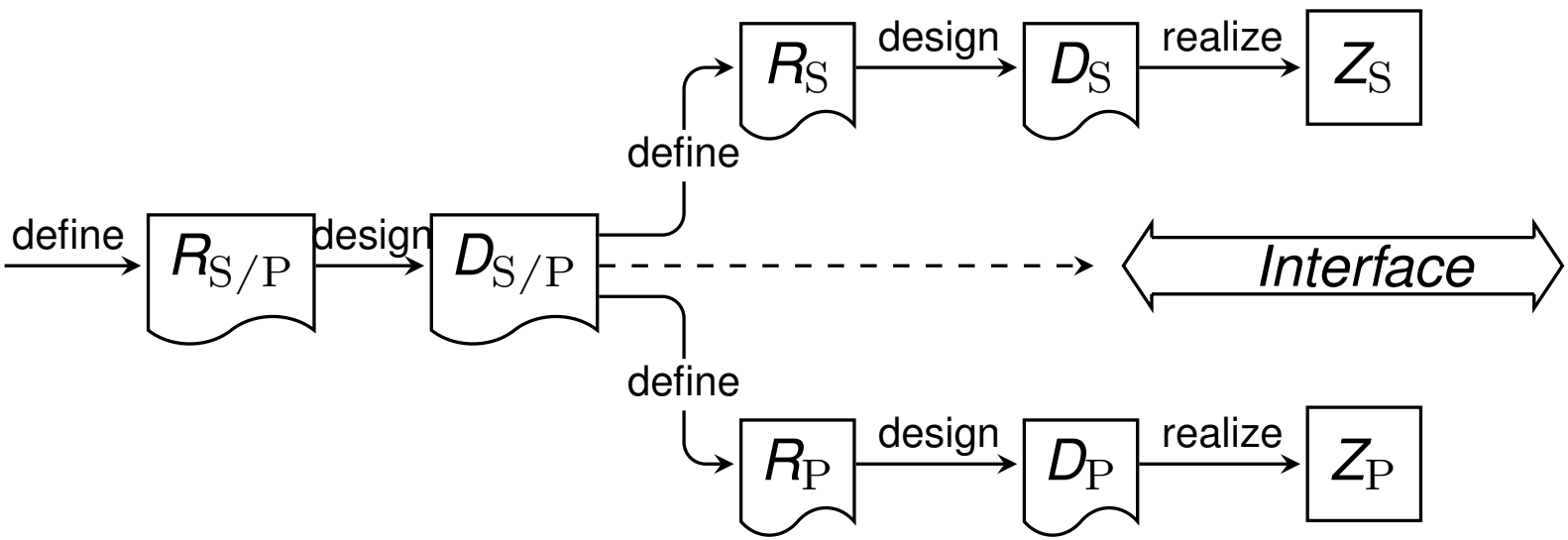
# Patient support system

PICU (user interface)



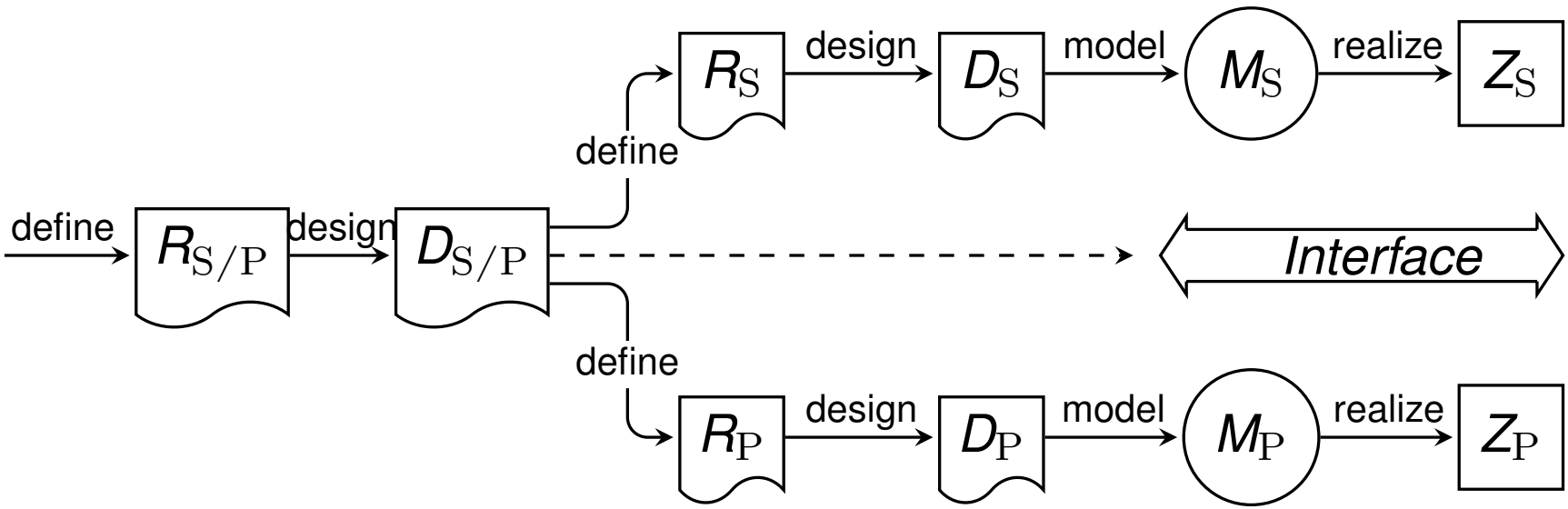
# Supervisory control design

## Conventional design



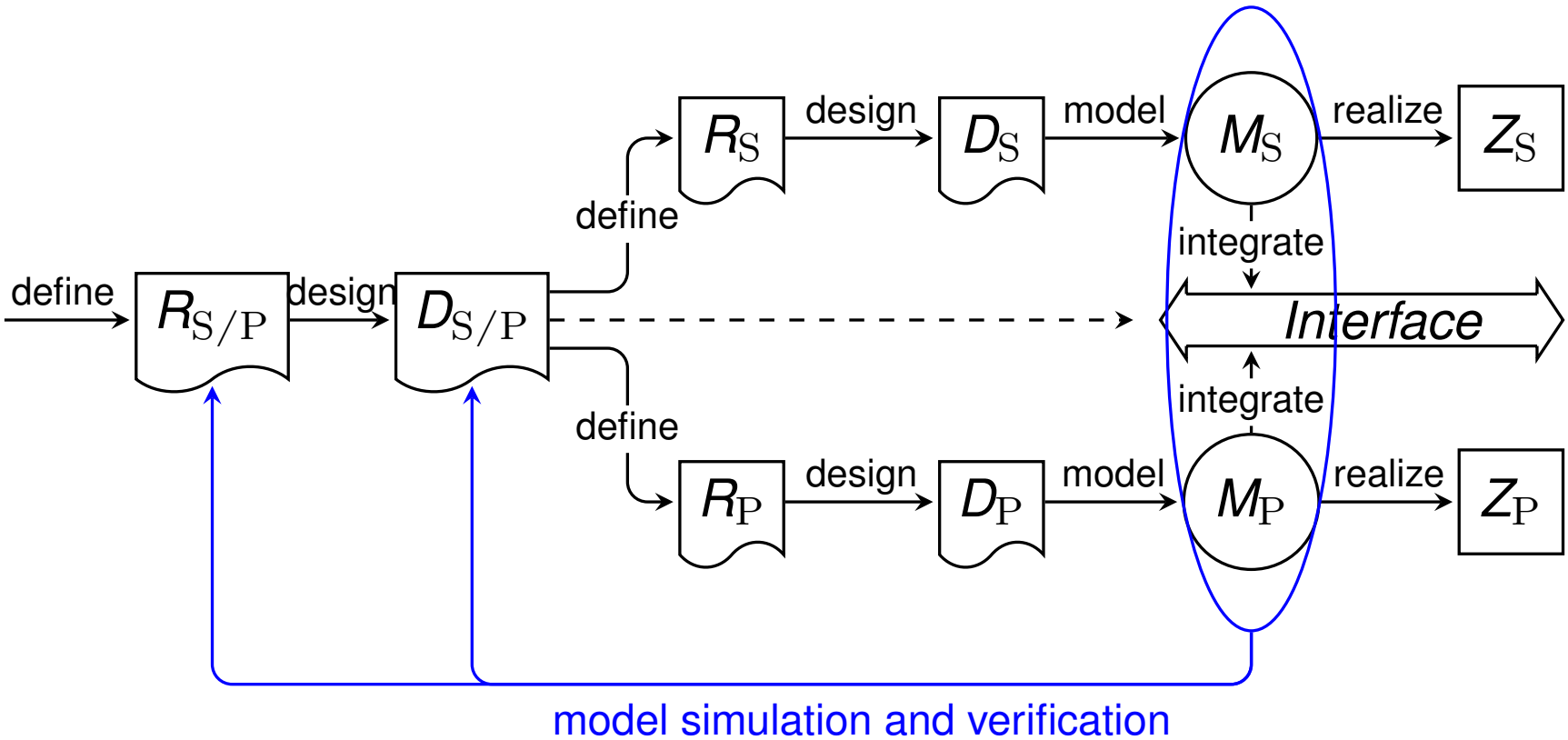
# Supervisory control design

## Model-based Engineering



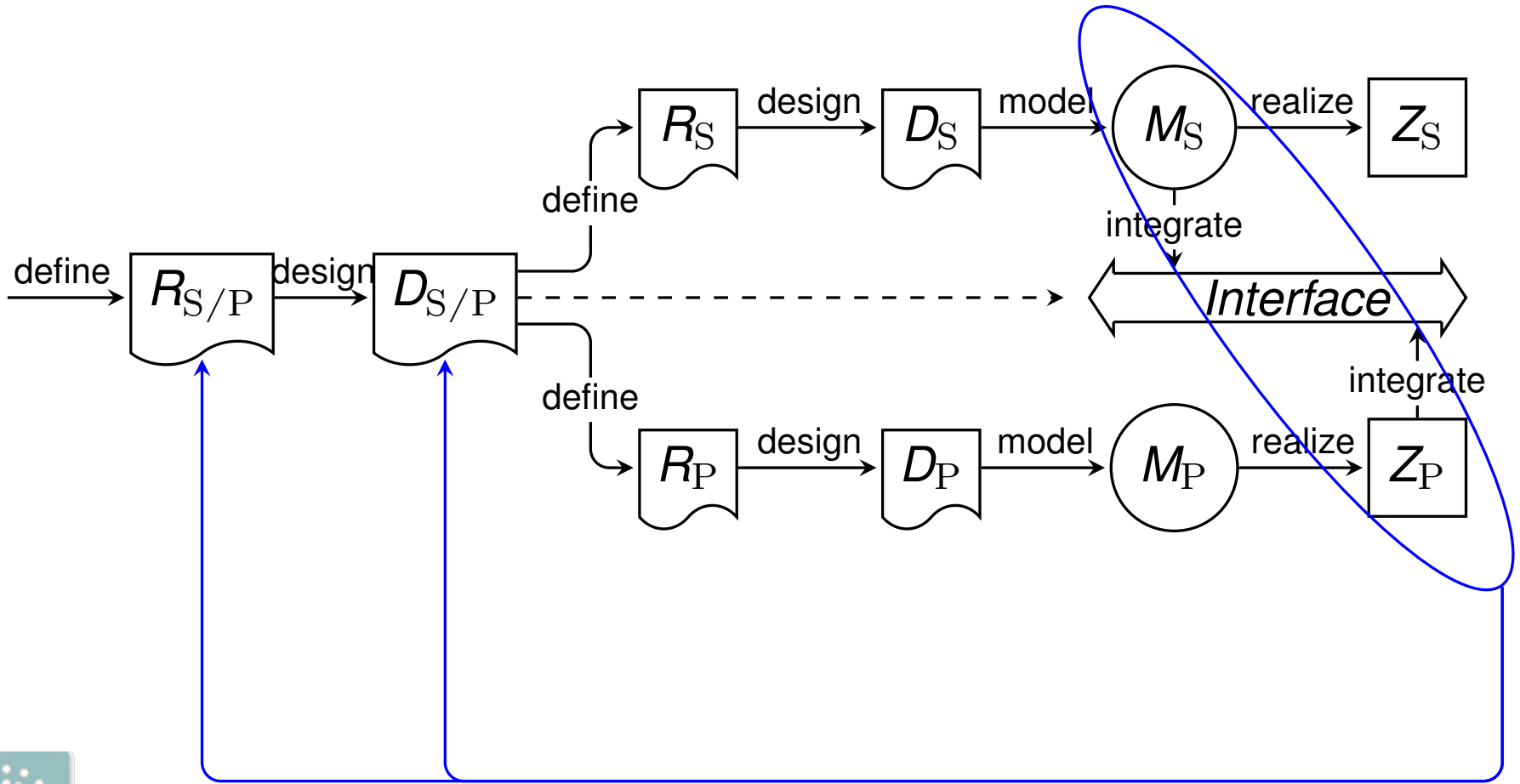
# Supervisory control design

## Model-based Engineering



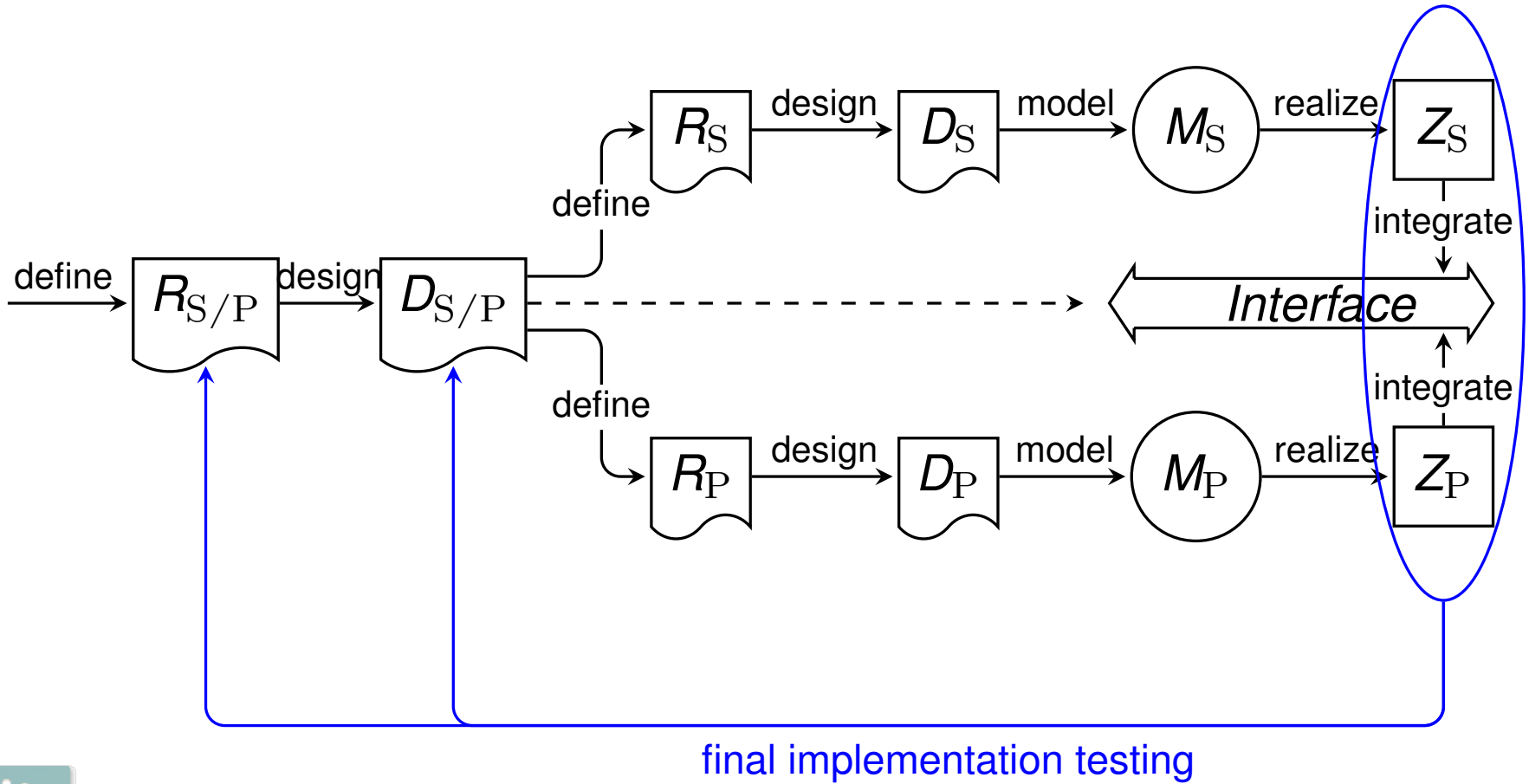
# Supervisory control design

## Model-based Engineering



# Supervisory control design

## Model-based Engineering



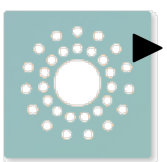
# Supervisory Control Synthesis

The resulting supervisor is:

- by construction mathematically correct w.r.t. $M_{R_S}$
- controllable
- non-blocking (deadlock and livelock free)
- maximally permissive allowing selection of 'optimal' sequence of events

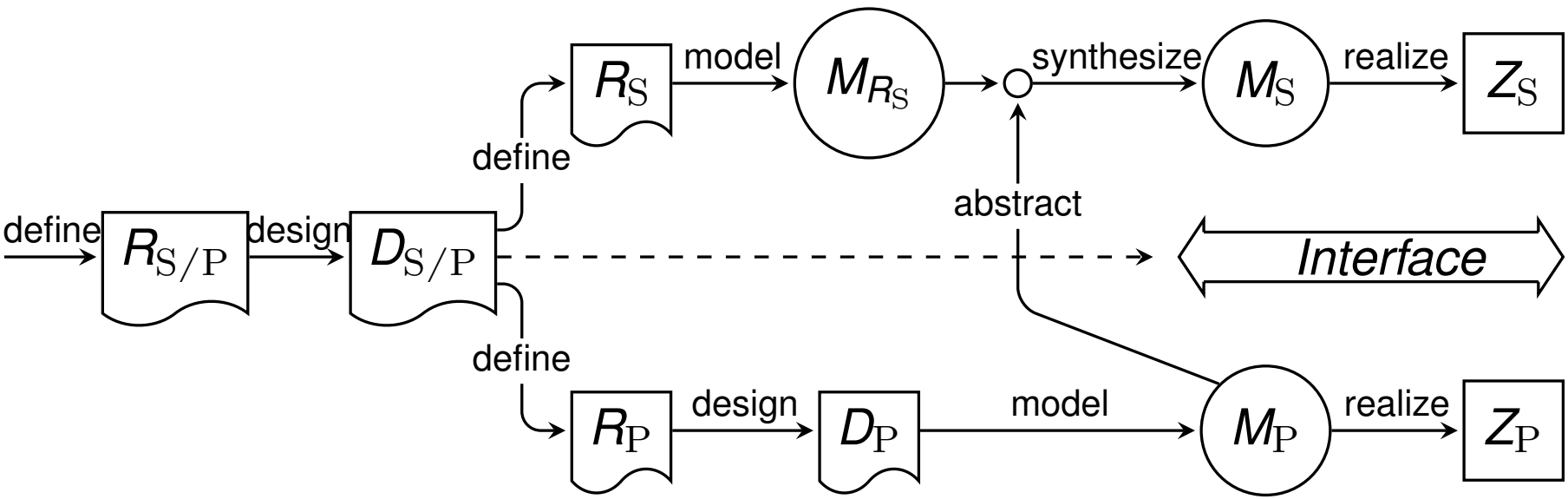
Approach:

- Model (uncontrolled) system $\Longrightarrow$ $M_P$ (hybrid model)
- Abstract from $M_P$ (hybrid model) $\Longrightarrow$ $M_P$ (discrete-event model)
- Model control requirements $R_S$ that determine when events may happen $\Longrightarrow$ $M_{R_S}$ (formal requirements)
- Synthesize the supervisor $\Longrightarrow$ $M_S$ (discrete-event model)



# Supervisory control design

## Model-based Engineering and Supervisory Control Synthesis



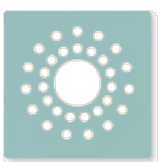
# Supervisory control synthesis and evolvability

## Advantages of the method

In case of changes in the required (control) functionality $R_{\mathrm{S}}$ only:

- the control requirements $M_{R_{\mathrm{S}}}$ have to be updated
- the uncontrolled system $M_{\mathrm{P}}$ might change (new sensors/actuators)

the supervisor and its implementation are regenerated.

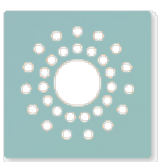


# Supervisory control synthesis and evolvability

## Model requirements

The system and requirements are decomposed into *small*, *loosely coupled* and *minimally restrictive* models:

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



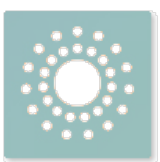
# Patient support system

## Uncontrolled system $M_P$

Uncontrolled system $M_P$ consists of 17 small automata describing:

- Horizontal axis
- Vertical axis
- User interface buttons

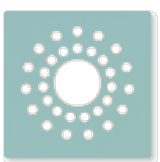
In total 1296 states and 27360 transitions for the uncontrolled system.



# Patient support system

## Control requirements $M_{R_S}$

- The model of the control requirements $M_{R_S}$ consists of 16 small automata
- Examples of requirements:
  - 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 horizontally in maximal out position
  - Tumble switch moves table up and down, or in and out
  - ...



# Patient support system

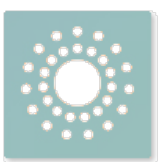
## Supervisor synthesis

Synthesis:

- The model of the supervisor $M_S$ consists of 2816 states and 21672 transitions
- Supervisor synthesis takes a minute on a desktop pc

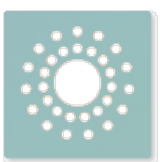
Implementation:

- The synthesized supervisor has been simulated in parallel with the (hybrid) model of the system
- The synthesized supervisor has been simulated in real-time with the actual patient support system (hardware-in-the-loop simulation)



# Concluding remarks

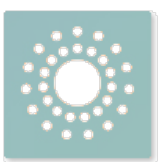
- Eliminated manual design of the supervisor
- Unambiguous specifications of the uncontrolled system and the control requirements
- Supports evolvability
- Implemented supervisor $Z_S$ on the real hardware
- Different theories available for supervisory control synthesis:
  - monolithic / modular / decentralized / hierarchical / interface-based supervisors
  - supervision under partial observation
  - event-based / state-based supervision



# Concluding remarks

*Q*-*T*-*C* triangle

- *Quality*: $Q \uparrow$
  The synthesized supervisor is by construction mathematically correct w.r.t. the modeled requirements
- *Time-to-market*: $T \downarrow$
  A change in required functionality leads to re-modeling of the requirements only
- *Costs*: $C \approx$
  The costs remain more or less the same



Embedded Systems INSTITUTE

PHILIPS

# Supervisory control synthesis for a patient support table

Rolf Theunissen, Ramon Schiffelers,
Bert van Beek, Koos Rooda

Systems Engineering Group
Dept. of Mechanical Engineering

TU/e Technische Universiteit Eindhoven University of Technology

December 4, 2008