Model-Based Engineering of Supervisory Controllers using the CIF

Ramon Schiffelers

joint work with

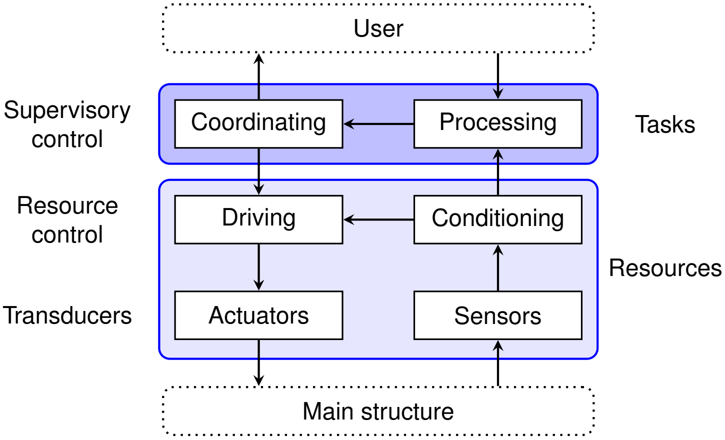
Rolf Theunissen, Bert van Beek, and Koos Rooda
Systems Engineering Group
Dept. of Mechanical Engineering
Eindhoven, University of Technology

October 6, 2009

Outline

- Supervisory control
- Supervisory control design
  - conventional
  - using MBE
  - using MBE and SCS
- Small example of MBE+SCS
- Languages + Translations
- Industrial case study: Patient support system of a MRI scanner
- Concluding remarks

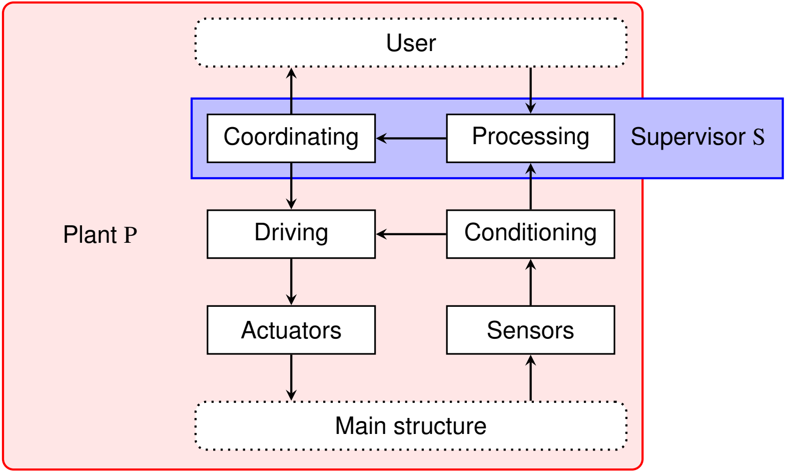
Supervisory control in high-tech systems



Systems view

A system can be divided in

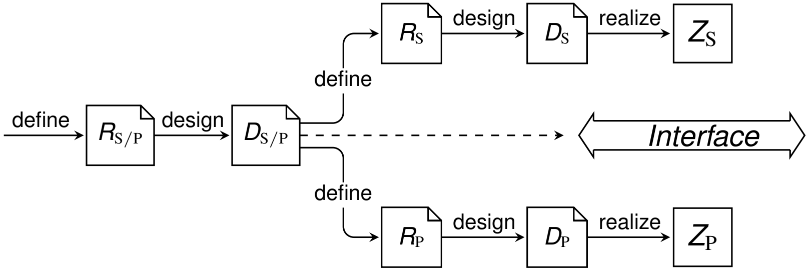
- (uncontrolled) Plant P
- Supervisor (controller) S



Supervisor S ensures that plant P satisfies control requirements R_{S}.

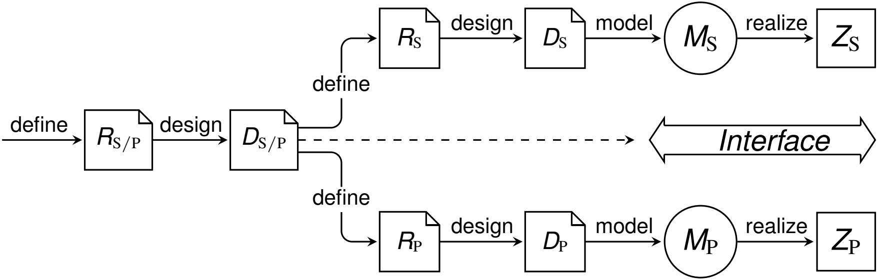
Supervisory control design

Conventional design



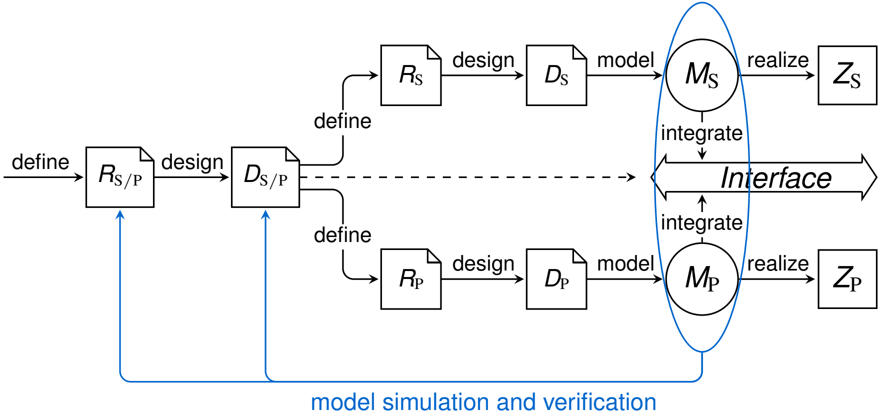
Supervisory control design

Model-based Engineering



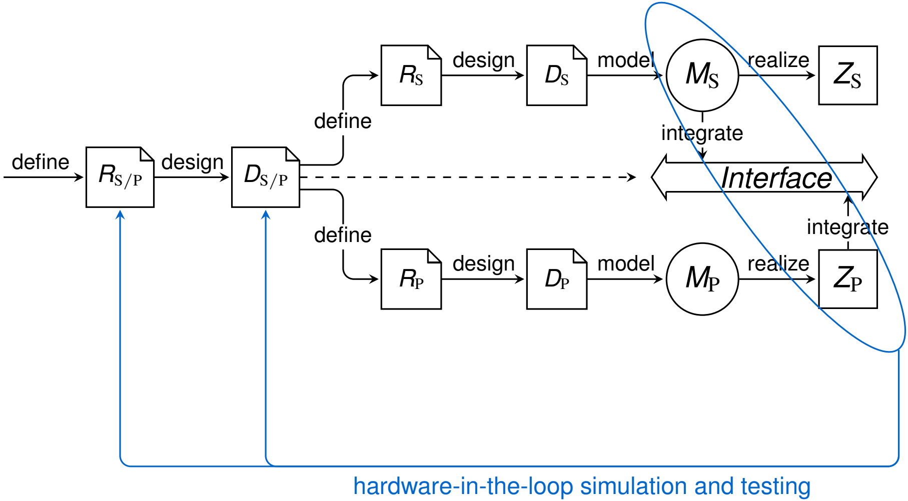
Supervisory control design

Model-based Engineering



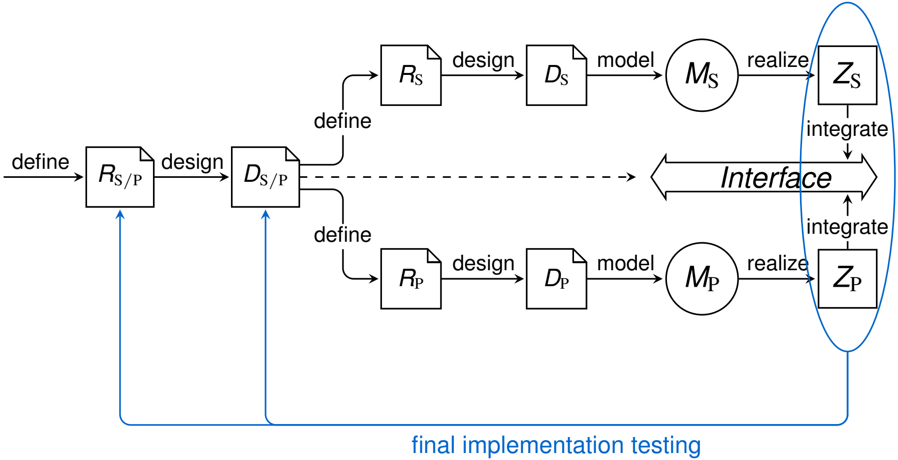
Supervisory control design

Model-based Engineering



Supervisory control design

Model-based Engineering



Supervisory Control Synthesis (SCS)

Approach:

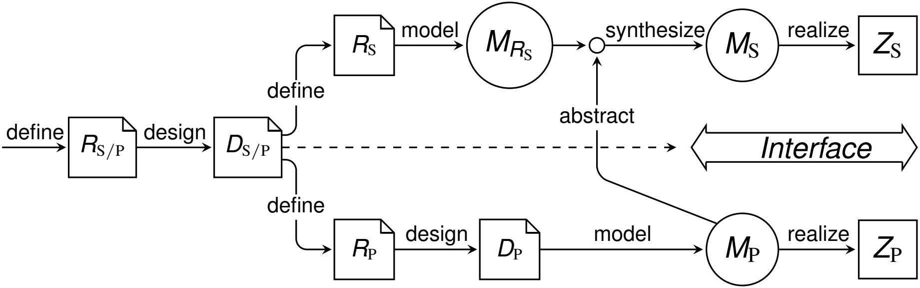
- Model (uncontrolled) plant $\Longrightarrow$ M_{P} (hybrid model)
- Abstract from M_{P} (hybrid model) $\Longrightarrow$ M_{P} (discrete-event model)
- Model control requirements R_{S} that determine when events may happen $\Longrightarrow$ $M_{R_{\mathrm{S}}}$ (formal requirements)
- Synthesize the supervisor $\Longrightarrow$ M_{S} (discrete-event model)

The resulting supervisor is

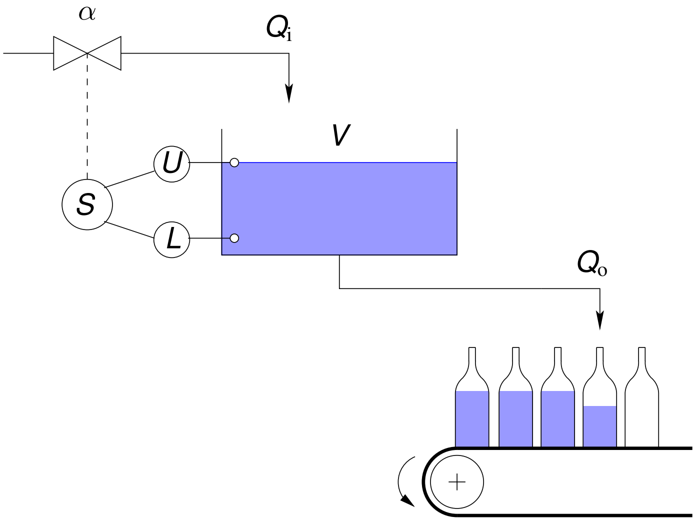
- by construction mathematically correct w.r.t. $M_{R_{\mathrm{S}}}$
- non-blocking (deadlock and livelock free)
- maximally permissive allowing selection of 'optimal' sequence of events

Supervisory control design

Model-based Engineering and Supervisory Control Synthesis



Example: tank controller



Languages and translations

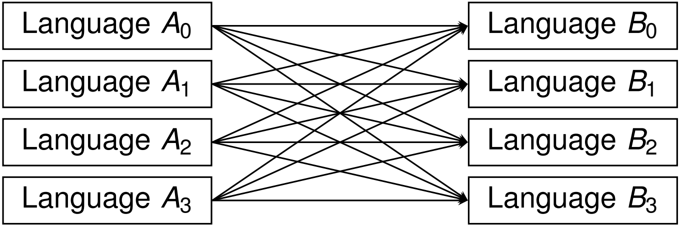
- Input languages for supervisor synthesis:
  - in the paper: *ADS* containing plants + requirements modeled by automata)
  - in this presentation *SCIM* containing plants + requirements modeled by automata + flow chart specifying the synthesis calculation.
- Interchange language: *CIF*.

Purpose of the CIF

- Establish inter-operability of a wide range of tools by means of model transformations to and from the CIF.
  Avoid the implementation of many bi-lateral translators between specific formalisms.
- Provide a generic modeling formalism and tools (such as a simulator) for a wide range of hybrid systems.

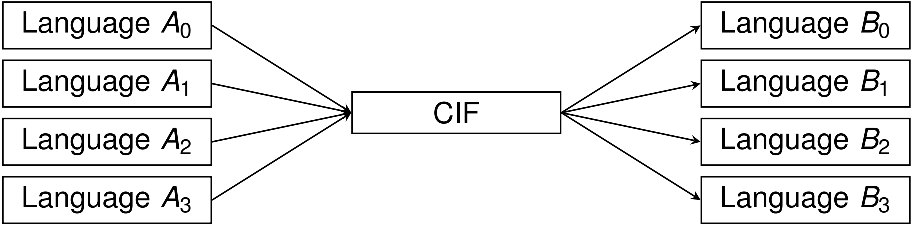
Examples of translations

Without the CIF:

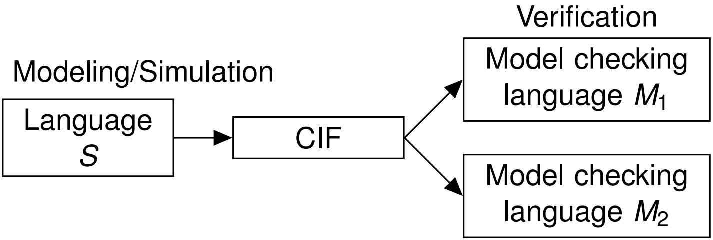


Examples of translations

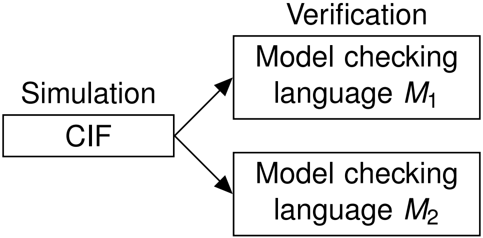
With the CIF:



Example of applying the CIF



Examples of applying the CIF



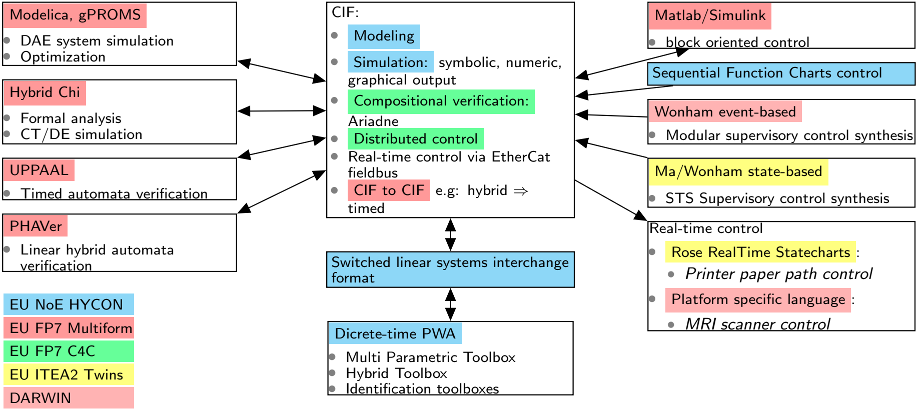
Current status of the CIF (1)

- Language
  - Domain model specified using *Ecore* class diagrams
  - Concrete syntax:
    - Mathematical notation that facilitates the definition of the semantics
    - ASCII notation for modeling
    - Graphical notation for modeling
  - Formal (behavioral) semantics specified using Structural Operational Semantics (SOS) [Plotkin]

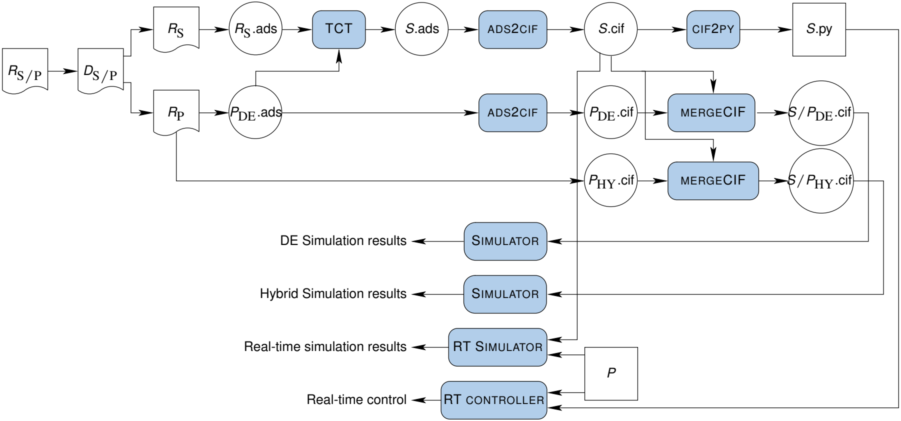
Current status of the CIF (2)

- Tools
  - Graphical Editor (based on GMF)
  - (Hybrid) Simulator
  - Real-Time simulator in order to control hardware via EtherCAT
  - Translators
    - ADS2CIF
    - SCIM2CIF
    - CIF2PY (subset!)
- Applications
  - Small examples
  - Industrial case study

Overview of projects on CIF



Tool framework based on CIF

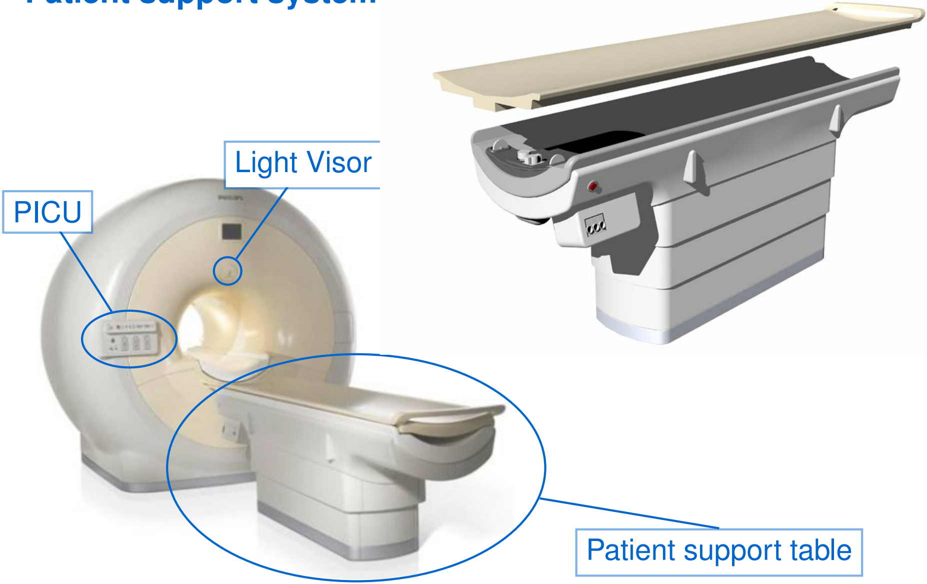


Translations

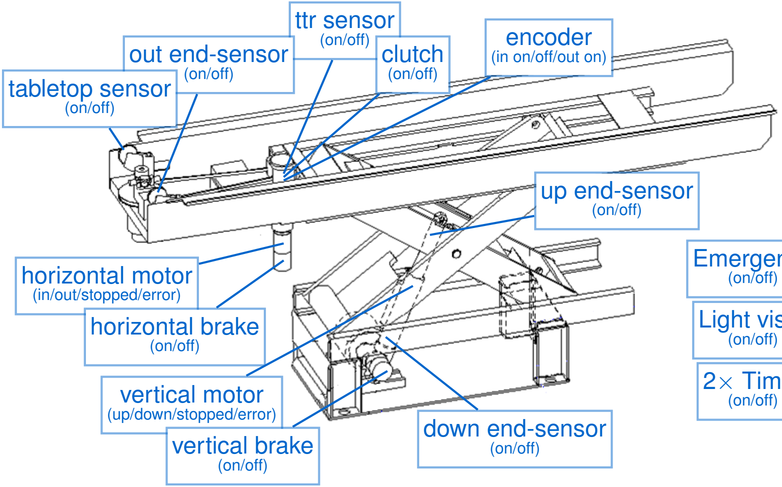
- ADS2CIF, model2model transformation implemented using a GPL (Python)
- CIF2PY: codegeneration, implemented using a GPL (Python)
- SCIM2CIF: model2model transformation
  - implemented using GPL (Python)
  - modeled using *ATL*

Industrial case study

Patient support system



Table



Emergency
(on/off)

Light visor
(on/off)

2× Timer
(on/off)

Patient support system

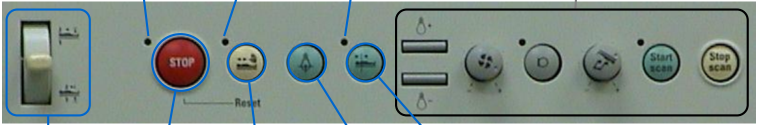
PICU (user interface)

Stop led

TTS led

Manual led

other buttons
light/ventilation/sound/start scan/stop scan



Tumble switch
up/neutral/down

Manual button

TTS button

Stop button

Light visor button

Patient support system

Uncontrolled plant M_P

Uncontrolled plant 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 plant.

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

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

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

Concluding remarks

- Developed a (tool) framework, based on MBE and SCS to develop supervisory controllers
- Applied the framework on an industrial case study
- Prototype SCIDE (Integrated Development Environment for Supervisory control synthesis)

Future work:

- Lots of theory available for supervisory control synthesis
  - monolitic / modular / decentralized / hierarchical / interface-based supervisors
  - event-based / state-based supervision
  - different formalisms for plant modeling and requirement specifications
- Translations
  - Which transformation language(s) (or GPL)?
  - Semantics / property preservation?
- How to deal/manage many translations?

Model-Based Engineering of Supervisory Controllers using the CIF

Ramon Schiffelers

joint work with

Rolf Theunissen, Bert van Beek, and Koos Rooda
Systems Engineering Group
Dept. of Mechanical Engineering
Eindhoven, University of Technology

October 6, 2009