Engineering based on mathematical models

Ramon Schiffelers

joint work with

Rolf Theunissen, Bert van Beek, Asia van de Mortel-Fronczak, Koos Rooda

Systems Engineering Group
Dept. of Mechanical Engineering
Eindhoven, University of Technology

Oktober 9, 2008
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
Supervisory control

Model-based Engineering (MBE)

Supervisory Control Synthesis (SCS)

Supervisory control design
  • conventional
  • using MBE
  • using MBE and SCS

Industrial case study: Patient support system of a MRI scanner

Concluding remarks
Supervisory control in high-tech systems

Main structure

- Actuators
- Sensors
- Driving
- Conditioning
- Resource control
- Coordinating
- Supervisory control

User

Tasks

Resources

Main structure
Model-based engineering

Framework

Figure inspired by the TANGRAM project
Model-based engineering

Simulation and verification

R → D → M → Z

R₁ → D₁ → M₁ → Z₁

Rₙ → Dₙ → Mₙ → Zₙ

Define → Design → Model → Realize

Interface

Model simulation and verification
Model-based engineering

Early integration

hardware-in-the-loop simulation and testing
Model-based engineering

Final implementation testing

$R_i \xrightarrow{\text{design}} D_i \xrightarrow{\text{model}} M_i \xrightarrow{\text{realize}} Z_i$

define

$R_n \xrightarrow{\text{design}} D_n \xrightarrow{\text{model}} M_n \xrightarrow{\text{realize}} Z_n$

interface

final implementation testing
Systems view

A system can be divided in

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

Supervisor S ensures that plant P satisfies its control requirements $R_S$. 
Supervisory control design

Conventional design

\[ \text{define } R_{S/P} \quad \text{design } D_{S/P} \quad \text{define } R_P \quad \text{design } D_P \quad \text{realize } Z_S \quad \text{realize } Z_P \]

\[ \text{Interface} \]
Supervisory control design

Model-based Engineering

\[ R_S / P \xrightarrow{\text{design}} D_S / P \xrightarrow{\text{define}} R_S \xrightarrow{\text{design}} D_S \xrightarrow{\text{model}} M_S \xrightarrow{\text{realize}} Z_S \]

\[ R_P \xrightarrow{\text{design}} D_P \xrightarrow{\text{model}} M_P \xrightarrow{\text{realize}} Z_P \]

Interface
Supervisory control design

Model-based Engineering

\[ R_S \xrightarrow{\text{define}} DS/P \xrightarrow{\text{design}} DS \xrightarrow{\text{define}} R_S \xrightarrow{\text{design}} D_S \xrightarrow{\text{model}} MS \xrightarrow{\text{integrate}} \]

\[ Z_S \xrightarrow{\text{realize}} MP \xrightarrow{\text{model}} DP \xrightarrow{\text{design}} R_P \xrightarrow{\text{define}} DS/P \xrightarrow{\text{design}} \text{Interface} \]

model simulation and verification
Supervisory control design

Model-based Engineering

interface define design define design define design model realize model realize model realize

hardware-in-the-loop simulation and testing
Supervisory control design

Model-based Engineering

interface defines
design defines
model simulation and verification
integrate hardware-in-the-loop simulation and testing
realize final implementation testing
Supervisory Control Synthesis (SCS)

The resulting supervisor is
- by construction mathematically correct w.r.t. $M_{Rs}$
- non-blocking (deadlock and livelock free)
- maximally permissive allowing selection of ’optimal’ sequence of events

Approach:
- Model (uncontrolled) plant $\Rightarrow M_P$ (hybrid model)
- Abstract from $M_P$ (hybrid model) $\Rightarrow M_P$ (discrete-event model)
- Model control requirements $R_S$ that determine when events may happen $\Rightarrow M_{Rs}$ (formal requirements)
- Synthesize the supervisor $\Rightarrow M_S$ (discrete-event model)
Supervisory control design

Model-based Engineering and Supervisory Control Synthesis
Industrial case study

Patient support system

Light Visor

PICU

Patient support table
Patient support system

Table

- Tabletop sensor (on/off)
- Position encoder (on/off)
- Horizontal brake (on/off)
- Horizontal motor (in/out/stopped)
- Clutch (on/off)
- Max out sensor (on/off)
- TTR button (on/off)
- Max up sensor (on/off)
- Max down sensor (on/off)
- Vertical motor (up/down/stopped)
- Vertical brake (on/off)
Patient support system

PICU (user interface)

- Stop led
- TTS led
- Manual led
- Tumble switch: up/neutral/down
- Stop button
- Manual button
- TTS button
- Light visor button
- Other buttons: light/ventilation/sound/start scan/stop scan
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.
Control requirements $M_{Rs}$

- The model of the control requirements $M_{Rs}$ 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
  - ...
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

- Eliminated manual design of the supervisor
- Combination of MBE and SCS works very well, also on a complex industrial case
- Lots of theory available for supervisory control synthesis
  - monolithic / modular / decentralized / hierarchical / interface-based supervisors
  - supervision under partial observation
  - event-based / state-based supervision
  - different formalisms for plant modeling and requirement specifications
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
Engineering based on mathematical models

Ramon Schiffelers

joint work with

Rolf Theunissen, Bert van Beek, Asia van de Mortel-Fronczak, Koos Rooda

Systems Engineering Group
Dept. of Mechanical Engineering
Eindhoven, University of Technology

Oktober 9, 2008