Supervisory Control Synthesis — the Focus in Model-Based Systems Engineering

Jos Baeten and
Asia van de Mortel-Fronczak

Systems Engineering Group
Department of Mechanical Engineering

November 23, 2011
What is a model?

A model is an abstraction.

Structure.
Behavior.
Other characteristics such as energy consumption.
Behavioral models use mathematics:

- Probability, stochastics (queueing, Markov). Performance, optimization. Ortec, CQM.
- Combinations: hybrid. $\chi$.  

Use of models in the system life cycle
Architecture. Sysarch of ESI.

Components are subsystems or aspect-systems.

Levels of abstraction: function (what), process (how), resource (with).
V model

- Requirements analysis
- System design
- Architecture design
- Module design
- Implementation
- User acceptance testing
- System testing
- Integration testing
- Unit testing
Behavioral models

Manufacturing networks
- continuous-state time-driven for performance analysis

Manufacturing machines
- discrete-state event-driven for supervisory control synthesis
- continuous-state time-driven for control synthesis
Embedded systems

- User
  - Supervisory controller(s)
  - Resource controller(s)
  - Actuators
  - Sensors
  - Structure

- Physical components
- Control components
Semiconductor

- Supply chain with nodes (fab, assembly, test)
- Node (fab) with areas (implant, photo, metal)
- Area (photo) with cells (litho, metrology)
- Cell (litho) with tools (track, scanner)
- Tool (scanner) with process units (lens, laser), and handlers (stage, wafer, reticle)
- Handler (stage) with frame, transducers, and controllers
- Transducers with mechanics, electronics, optics, and pneumatics
Key performance indicators $F$, $Q$, $T$, $C$:
- $F$ – functionality, complexity increase
- $Q$ – quality should be maintained
- $T$ – time-to-market increases
- $C$ – cost increases
- Control software greater in size and complexity
- Control software time-consuming testing
Model-based systems engineering

\[ R \xrightarrow{\text{define}} D \xrightarrow{\text{design}} R_S \xrightarrow{\text{design}} D_S \xrightarrow{\text{model}} S \xrightarrow{\text{realize}} S \]

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

Interface \( I \)

Simulation and verification

Early integration

Validation and testing
Model-based systems engineering

Simulation and verification
Early integration
Validation and testing

R → D → Rs → Ds → S

R → D → Rs → Ds → S

Interface I
Integrate

P → S

S → P

S → P
Synthesis-based systems engineering

define design

R → D → Rs → S → Rs → S

define design model synthesize generate

Rs → S

Integrate

Interface I

integrate design model realize

Rp → Dp → P → P

integrate integrate integrate
Synthesis-based systems engineering

R\textsuperscript{S} \xrightarrow{\text{model}} \overset{\text{synthesize}}{\xrightarrow{\text{generate}}} S \xrightarrow{\text{integrate}} \overset{\text{integrate}}{\xrightarrow{\text{integrate}}} P \xrightarrow{\text{realize}} P

\overset{\text{define}}{\xrightarrow{\text{define}}} R \xrightarrow{\text{design}} D

D \xrightarrow{\text{model}} \overset{\text{synthesize}}{\xrightarrow{\text{generate}}} S

\overset{\text{define}}{\xrightarrow{\text{define}}} R\text{\textsubscript{P}} \xrightarrow{\text{design}} D\text{\textsubscript{P}}

simulation and verification

early integration
validation and testing

TU/e
Technische Universiteit
Eindhoven
University of Technology
Supervisory control problem

Plant $P$ and supervisor $S$ form a discrete-event system:

- $P$ under control of $S$ ($S/P$) satisfies requirement $R$
- $S$ does not disable uncontrollable events
- Output of $S$ only depends on observable outputs of $P$
- $S/P$ is nonblocking
- $S$ is optimal (maximally permissive)
A workcell consists of two machines $M_1$ and $M_2$, and an automated guided vehicle $AGV$.

Components functionality:
- $AGV$ can load a workpiece at $M_1/M_2$ and unload it at $M_2/B$. 

\[ a_1 \rightarrow M_1 \rightarrow AGV \uparrow \begin{array}{c} a_2 \\ \downarrow b_2 \end{array} M_2 \rightarrow B \]
$M_1$, $M_2$, and AGV are modeled by automata:

**$M_1$:**

- Idle
- Busy

**$M_2$:**

- Idle
- Busy

**AGV:**

- At$_M_1$
- Empty
- At$_M_2$

Transitions:

- $a_1$: Idle to Busy
- $b_1$: Busy to Idle
- $a_2$: Empty to At$_M_1$
- $b_2$: At$_M_1$ to Empty
- $c$: Empty to At$_M_2$
Absence of control results in a blocking situation (deadlock in state 7).

In this case, we have no additional restrictions on admissible behavior.
The system under control of the following "supervisor" avoids the blocking situation.

- This "supervisor" disables uncontrollable event $b_1$ in state 5.
- A supervisor may only disable controllable events.
The following "supervisor" avoids state 5 by disabling controllable $a_2$ in state 3 and controllable $a_1$ in state 4.

This "supervisor" introduces a new blocking situation, state 3.
Finally, the following supervisor delivers a proper optimal control to the plant.
Supervisory control theory

- Provides means to synthesize $S$
- Conceptually simple framework (based on automata)
- Computational complexity is high for systems of industrial size

Several advanced techniques to reduce synthesis complexity:
- Modular
- Hierarchical
- Interface-based hierarchical
- Coordinated distributed
- Aggregated distributed
Distributed control architecture

Global command

\( P_1 \)

Local command

\( S_1 \)

Command fusion

\( S_2 \)

Local command

Local observation

Composition of \( P_1 \) and \( P_2 \)

Local observation

Global command

\( P_2 \)
Coordinated distributed synthesis

\[ P = W_1 \times W_2, R \]

\[ W_1 = (P_1 \times S_1) / \approx \Sigma_1 \cap \Sigma' \]

\[ W_2 = (P_2 \times S_2) / \approx \Sigma_2 \cap \Sigma' \]
Aggregated distributed synthesis

\[ W_1 = \frac{(P_1 \times S_1)}{\approx \Sigma_1 \cap \Sigma'} \]

\[ P_1, R_1 \rightarrow S_1 \]

\[ P_2 \times W_1, R_2 \rightarrow S_2 \]
Industrial cases

Supervisory control synthesis for:

- Patient support system of an MRI scanner
- Communication system of an MRI scanner
Patient support system of an MRI scanner

Safe tabletop handling
Control requirements

- Ensure that the tabletop does not move beyond its vertical and horizontal end positions.
- Prevent collisions of the tabletop with the magnet.
- Define the conditions for manual and automatic movements of the tabletop.
- Enable the operator to control the system by means of the manual button and the tumble switch.
A centralized supervisor was synthesized using the TCT tool [Wonham].

The system under control of the supervisor was validated using simulation.

The supervisor was tested on the real system.

After a functional change, approximately four hours work was needed to repeat the above steps.
Results

- Plant model: 672 states.
- Requirement model: 4.128 states.
- The supervisor: 2.976 states.
Industrial cases

- Exception handling in printers
- Coordination of maintenance procedures in printers
Coordination of maintenance procedures in printers
Control requirements

- Maintenance operations may only be performed if the power mode of the printing process is Standby.

- Maintenance operations should be scheduled if their soft deadline is reached and no print jobs are in progress or if their hard deadline is reached.

- Only scheduled maintenance operations can be started.

- The power mode of the printing process should conform to the mode determined by the print job managers unless it is overridden by a pending maintenance operation.
Results

▶ A centralized supervisor was synthesized using the synthesis tool based on state-tree structures [Ma].

▶ The system under control of the supervisor was validated using simulation.

▶ The supervisor is converted to C++ for execution on the existing control platform.
Results

- Plant model: 25 automata with 2 to 24 states.
- Requirements: 23 generalized state-based expressions (more than 500 standard state-based expressions).
- The supervisor: $6 \cdot 10^6$ states.
Industrial cases

- Passenger safety in theme park vehicles
Theme park vehicle

Handling of proximity, emergency, and hardware errors in theme park vehicles

- Scene Program Handler (on/off)
- Steer Motor (on/off)
- Ride Control (start/stop)
- Battery (empty/OK)
- 4 Proximity Sensors (on/off)
- User Interface (3 LEDs/3 buttons) (on/off)
- Drive Motor (on/off/stopped)
- Bumper Switch (on/off)
Control requirements

- To avoid collisions with other vehicles or obstacles, the multimover should drive at a safe speed and stop if the obstacle is too close to it.

- The vehicle should stop immediately and should be powered off when:
  - a collision or a system failure occurs,
  - the battery level is too low.

After the problem is resolved, the multimover should be manually deployed back into the ride by an operator.
Results

- A centralized supervisor was synthesized using the synthesis tool based on state-tree structures [Ma].
- A distributed supervisor was synthesized using the synthesis tool based on automaton abstraction [SE group].
- The system under control of both supervisors was validated using simulation.
- Both supervisors were tested on the real system.
- After a functional change, approximately four hours work was needed to repeat the above steps.
Results

- Plant model: 17 automata with 2 to 4 states.
- Requirements: 30 automata with 2 to 7 states.
- Distributed supervisor:

<table>
<thead>
<tr>
<th>Module</th>
<th># states</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED actuation</td>
<td>25</td>
</tr>
<tr>
<td>Motor actuation</td>
<td>41</td>
</tr>
<tr>
<td>Button handling</td>
<td>465</td>
</tr>
<tr>
<td>Emergency handling</td>
<td>89</td>
</tr>
<tr>
<td>Proximity handling</td>
<td>225</td>
</tr>
</tbody>
</table>
Industrial cases

- Cruise control of a truck
Tool chain for SCS

- Algorithms for synthesis
- Model transformations
- Common Interchange Format
Conclusions

- Model-based systems engineering gives faster product development
- Supervisor synthesis eliminates manual design of control software and reduces testing effort
- Successful proofs of concept delivered for implementation of advanced synthesis techniques
- Event-based distributed framework supports reconfigurability
- Synthesis-based systems engineering is applicable in industry for developing supervisory controllers
- Formal models and methods are essential for high-tech systems design
Supervisory Control Synthesis —
the Focus in Model-Based Systems Engineering

Jos Baeten and
Asia van de Mortel-Fronczak

Systems Engineering Group
Department of Mechanical Engineering

November 23, 2011