Integrated Model-based Support for the Design of Complex Controlled Systems

Martin Hufner
Christian Sonntag
Sebastian Engell

TU Dortmund
Process Dynamics and Operations Group
with inputs from the whole MULTIFORM team
Outline

• Illustrative example
  – Integrated design of a pipeless plant
• The MULTIFORM Design Framework
• Design flow example
  – Use of multiple tools and propagation of the results
  – Algorithmic model exchange
  – Logic controller specification and synthesis
  – Verification
• Conclusions & Outlook
Example: Design of a Pipeless Plant
Example: Design of a Pipeless Plant

Control PC

AGVs

Storage station

Mixing station

Color stations

Charging stations

Product

Camera

multiform

Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems
Example: Design of a Pipeless Plant
Challenges for Model-based Design (1)

- Design and validation on different levels of abstraction
  - Specification
    - Specification of the tasks and of the performance of the system
  - High-level design
    - Choice of the equipment, feasibility and bottleneck analysis, throughput maximization, plant layout optimization
  - Low-level design
    - Optimization and control of processing steps and motion dynamics, logic control
    - Choice of sensors and actuators, communication system
  - Implementation
    - PLCs, embedded controllers, communication system

![Diagram of a manufacturing system with labeled components: Control PC, Camera, AGVs, Storage station, Charging stations, Product, Color stations, Mixing station.]

---

**multiform**

Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems
Challenges for Model-based Design (2)

- The control system spans the complete control hierarchy
  - Coordination control
    - Scheduling and performance optimization
  - Advanced control
    - Control of batch processes
    - AGV path planning
  - Regulatory control
    - AGV motion control
    - Docking control
    - Sequence control in the processing stations
    - Low-level continuous control
  - Low-level safety-related control
Challenges for Model-based Design (2)

- The control system spans the complete control hierarchy
  - Coordination control
    - Scheduling and performance optimization
  - Advanced control
    - Control of batch processes
    - AGV path planning
  - Regulatory control
    - AGV motion control
    - Docking control
    - Sequence control in the processing stations
    - Low-level continuous control
  - Low-level safety-related control

Diagram: Coordination Control, Advanced Control, Sequence Control, Regulatory Control, Low-level Safety-related Control.
Challenges for Model-based Design (2)

- The control system spans the complete control hierarchy
  - Coordination control
    - Scheduling and performance optimization
  - Advanced control
    - Control of batch processes
    - AGV path planning
  - Regulatory control
    - AGV motion control
    - Docking control
    - Sequence control in the processing stations
    - Low-level continuous control
  - Low-level safety-related control

Timed or hybrid models
Continuous models
Challenges for Model-based Design (2)

• The control system spans the complete control hierarchy
  – Coordination control
    • Scheduling and performance optimization
  – Advanced control
    • Control of batch processes
    • AGV path planning
  – Regulatory control
    • AGV motion control
    • Docking control
    • Sequence control in the processing stations
    • Low-level continuous control
  – Low-level safety-related control
Challenges for Model-based Design (2)

- The control system spans the complete control hierarchy
  - Coordination control
    - Scheduling and performance optimization
  - Advanced control
    - Control of batch processes
    - AGV path planning
  - Regulatory control
    - AGV motion control
    - Docking control
    - Sequence control in the processing stations
    - Low-level continuous control
  - Low-level safety-related control

Discrete-event, timed, and hybrid models
Continuous models
Timed or hybrid models
Starting point of the project MULTIFORM
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models.
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, several tools and modeling formalisms will be used in the design process.
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, several tools and modeling formalisms will be used in the design process.

• Re-modelling should be avoided as much as possible
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models.

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, several tools and modeling formalisms will be used in the design process.

• Re-modelling should be avoided as much as possible
  – Direct coupling between tools.
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, several tools and modeling formalisms will be used in the design process.

• Re-modelling should be avoided as much as possible
  – Direct coupling between tools
  – Model exchange and tool chains using a neutral interchange format
Starting point of the project MULTIFORM

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models.

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, several tools and modeling formalisms will be used in the design process.

• Re-modelling should be avoided as much as possible
  – Direct coupling between tools
  – Model exchange and tool chains using a neutral interchange format

• Tools should be embedded into a common environment that provides data and model management and backtracking capabilities as well as parameter consistency.
The MULTIFORM Design Framework

- Consistent integration of design models into a common software framework

- Support of a generic design flow model
  - Design decisions
  - System design

- Consistency management
  - Communication of design parameters
  - Conflict detection
  - Models and results management
A design step represents a stage of the development process

Exploration of some design parameters using model-based tools or heuristics
  - Example: Numbers of AGVs and stations

Fixing the values of the system parameters leads to a new design step

A design step may encompass several views
  - E.g. structural or electrical

Design Framework: Design Steps
“Experiment”: General Concept of the Use of a Tool

- Generic interface for tool connection to the MULTIFORM integration framework

Experiment setups are stored for re-use
Design Flow: Design Tasks for the Pipeless Plant

Design tasks

 Requirements
  Feasibility analysis
  Plant layout design
    Docking
      AGV speed analysis
        Validation
          Controller design
            Code generation
Design Flow: Design Tasks for the Pipeless Plant

Design tasks

Requirements

Feasibility analysis

Plant layout design

Docking

AGV speed analysis

Validation

Controller design

Code generation
Design Flow: Design Tasks for the Pipeless Plant

Design tasks

Requirements
  ↓
  Feasibility analysis
  ↓
  Plant layout design
  ↓
  Docking
  ↓
  AGV speed analysis
  ↓
  Validation
  ↓
  Controller design
  ↓
  Code generation

Requirements for the Pipeless Plant include the following design tasks:

1. Feasibility analysis
2. Plant layout design
3. Docking
4. AGV speed analysis
5. Validation
6. Controller design
7. Code generation
Design Flow: Refinement of Requirements

Design tasks

Requirements

Feasibility analysis

Plant layout design

Docking

AGV speed analysis

Validation

Controller design

Code generation

Models

Boderc key drivers
Design Flow: Refinement of Requirements

Design tasks: Requirements

- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

Models: Boderc keydrivers

Results: First design parameters and assumptions
Design Flow: Feasibility Analysis

Design tasks
- Requirements
  - Feasibility analysis
    - Plant layout design
      - Docking
      - AGV speed analysis
        - Design parameter & assumptions
  - Validation
    - Controller design
      - Code generation

Models
- Boderc keydrivers
  - Timed Chi
    - Design parameter & assumptions
    - Complete coarse plant model
    - Purpose: Simulation

Results
- First design parameters and assumptions

Forwarded information via Design Framework
Design Flow: Feasibility Analysis

Boderc key drivers & Design Decisions

System configuration:
• # and type of stations
• Recipes
• # and type of AGVs
Design Flow: Feasibility Analysis

Boderc key drivers & Design Decisions

Timed Chi model – Complete coarse plant model
– Timed model
– Simulation

System configuration:
• # and type of stations
• Recipes
• # and type of AGVs
Design Flow: Feasibility Analysis

Boderc key drivers & Design Decisions

Timed Chi model – Complete coarse plant model
  – Timed model
  – Simulation

System configuration:
• # and type of stations
• Recipes
• # and type of AGVs

Requirements ➔ Feasibility Analysis

Design step ➔ Design step

Design flow
Design Flow: Feasibility Analysis

Boderc key drivers & Design Decisions

Timed Chi model – Complete coarse plant model
  – Timed model
  – Simulation

System configuration:
  • # and type of stations
  • Recipes
  • # and type of AGVs

Design Flow: Feasibility Analysis

Another View
Control View
Structure View
**Design Flow: Feasibility Analysis**

**Boderc key drivers & Design Decisions**

**Timed Chi model** – Complete coarse plant model
- Timed model
- Simulation

**System configuration:**
- # and type of stations
- Recipes
- # and type of AGVs

**Design Flow:** Feasibility Analysis

- Requirements
- Feasibility Analysis
- Design step

**Another View**

**Control View**

**Structure View**

**Plant**

*multiform*

Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems
Design Flow: Feasibility Analysis

Boderc key drivers & Design Decisions

Timed Chi model – Complete coarse plant model
  – Timed model
  – Simulation

System configuration:
  • # and type of stations
  • Recipes
  • # and type of AGVs

Design flow

Requirements
  Design step

Feasibility Analysis
  Design step

Structure View

Another View

Control View

System block

Plant

Chi model

Model of the complete plant using approximation
Design Flow: Feasibility Analysis

Boderc key drivers

Timed Chi
  – Complete plant
  – Timed model
  – Simulation

Chi model

System configuration:
- # Stations = 5
- StationTypes = [5x1]
- StationPositions = [5x2]
- # AGVs = 3
- RecipeList = [6x1]
- MovementTimes = [5x5]
**Design Flow: Feasibility Analysis Experiment**

**Boderc key drivers**
- Timed Chi
  - Complete plant
  - Timed model
  - Simulation

**System configuration:**
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]

**Experiment:**
- Set input:
  - Model parameters
    - #Stations:=5
    - StationTypes
    - #AGVs:=3
    - MovementTimes:=[5x5]
Design Flow: Feasibility Analysis Experiment

Boderc key drivers

- Timed Chi
  - Complete plant
  - Timed model
  - Simulation

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]

Experiment:

Set input:

Model parameters
- #Stations:=5

Tool parameters

Chi model
Design Flow: Feasibility Analysis Experiment

Boderc key drivers

Timed Chi
  - Complete plant
  - Timed model
  - Simulation

System configuration:
  - # Stations=5
  - Station Types=[5x1]
  - Station Positions=[5x2]
  - # AGVs=3
  - Recipe List=[6x1]
  - Movement Times=[5x5]

Experiment:

Set input:

Model parameters
#Stations:=5

Tool parameters

Chi model

Select tool:

Chi Simulator
**Bodarc key drivers**

- Timed Chi
  - Complete plant
  - Timed model
  - Simulation

**System configuration:**
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]

**Experiment:**

Set input:
- **Model parameters**
  - #Stations:=5

Tool parameters

Chi model

Select tool:
Chi Simulator

Save results:
Chi Simulation results
Design Flow: Feasibility Analysis Experiment

Boderc key drivers
- Complete plant
- Timed model
- Simulation

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=12

Chi lot-time diagram

Number in block represents vessel
**Design Flow: Feasibility Analysis Experiment**

**Boderc key drivers**
- Timed Chi
  - Complete plant
  - Timed model
  - Simulation

**System configuration:**
- # Stations = 5
- Station Types = [5x1]
- Station Positions = [5x2]
- # AGVs = 3
- Recipe List = [6x1]
- Movement Times = [5x5]
- Recipes/Day = 12

**Iteration until result satisfactory**

**Experiment:**

**Set input:**
- **Model parameters**
  - #Stations = 5

**Tool parameters**

**Chi model**

**Select tool:**
- Chi Simulator

**Save results:**
- Chi Simulation results

Add new parameter and value from results
Design Flow: Feasibility Analysis

Design tasks

- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

Models

- Boderc keydrivers
  - Design parameter & assumptions
- Timed Chi
  - Complete coarse plant model
  - Here: Timed model
  - Purpose: Simulation

Results

- First design parameters and assumptions
- Feasible plant layouts
  - (1 mixing, 2 filling, 2 charging)

Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems
Design Flow: Plant Layout Design / Scheduling

- **Design tasks**
  - Requirements
  - Feasibility analysis
  - Plant layout design
  - Docking
  - AGV speed analysis
  - Validation
  - Controller design
  - Code generation

- **Models**
  - Boderc keydrivers
  - Timed Chi
  - Uppaal
    - Complete plant model
    - Timed Automata model
    - Scheduling

- **Results**
  - First design parameters and assumptions
  - Feasible plant layouts (1 mixing, 2 filling, 2 charging)
  - Algorithmic model transformation

Forwarded information
Model transformation
The Compositional Interchange Format (CIF)  
[Bert van Beek et al., TUE]

• Purposes
  – Establish inter-operability of a wide range of tools
  – Provide a generic formalism for general hybrid systems

• Major features
  – Formal and compositional semantics
    • Independent of implementation aspects
    • Mathematical correctness proofs of translations
      → Property-preserving model transformations possible
  – Fully implicit DAE dynamics (possibly discontinuous)
  – Hierarchy and re-usability
    • Parallel composition with different communication concepts
  – Model component interaction
    • Point to point communication, multi-component synchronization, broadcast communication, shared variables
  – Different urgency concepts

http://devel.se.wtb.tue.nl/trac/cif/
The Compositional Interchange Format (CIF) [Bert van Beek et al., TUE]

*State*

*Transition*

*Invariants* (equations that are active when state is active)

\[ v' = -g \]

*Initial* (Conditions if state is initially active)
The Compositional Interchange Format (CIF) [Bert van Beek et al., TUE]

State

Transition

**Invariants**
(equations that are active when state is active)
e.g.:
$v' = -g$

**Guards**
(transition can only be taken if guard is true)
e.g.: $a > b$

**Updates**
(new discrete values or reinitialization)
e.g.: $z := 5$, \{v\}: $\text{new}(v) = 2$

**Synchronization**
(between different automata via labels or channels)

**Initial**
(Conditions if state is initially active)

**Urgency**
(nondeterminism, determinism, stochastic)
The Compositional Interchange Format (CIF)  
[Bert van Beek et al., TUE]

Invariants  
(equations that are active when state is active)  
e.g.: \( v' = -g \)

Guards  
(transition can only be taken if guard is true)  
e.g.: \( a > b \)

Updates  
(new discrete values or reinitialization)  
e.g.: \( z := 5, \{v\}: new(v) = 2 \)

Synchronization  
(between different automata via labels or channels)

Urgency  
(nondeterminism, determinism, stochastic)

Initial  
(Conditions if state is initially active)

State  
Transition

Formal definition by *Structural Operational Semantics* (SOS) rules, e.g.:

\[
\begin{align*}
(a_0, \sigma) & \xrightarrow{a, true, X} (a'_0, \sigma'), \ (a_1, \sigma) \xrightarrow{a, true, X} (a'_1, \sigma') \\
(a_0 \parallel a_1, \sigma) & \xrightarrow{a, true, X} (a'_0 \parallel a'_1, \sigma')
\end{align*}
\]
Transformations – gPROMS $\Rightarrow$ CIF (Excerpt)
Transformations – gPROMS ➔ CIF (Excerpt)
Transformations – gPROMS \(\rightarrow\) CIF (Excerpt)

**Model equations**

- **automaton**  
  
  \[
  z_{01} \\
  v = -g \\
  \]

- **automaton**  
  
  \[
  z_{00} \\
  \]
  
  \[
  z_{01} \\
  v = -g \\
  \]
  
  \[
  z_{02} \\
  v = 0 \\
  \]

- \(\neg (h \leq 0 \land v \leq 0)\) // if-true
- \(h \leq 0 \land v \leq 0\) // if-true
- \(\neg (h \leq 0 \land v \leq 0)\) // if-true
Transformations – gPROMS $\rightarrow$ CIF (Excerpt)

- **Process**
  - **Model**
    - **Task**

- **Model equations**
  - Algorithmic statements
  - Automaton

- **transformations**
  - For unconditional equations
  - For conditional equations
  - For sequential statements
  - For loops

**Model** equations

- $z_{01}: v = -g$

**Automaton**

- For unconditional equations
- For conditional equations
- For sequential statements
- For loops

**Algorithmic statements**

- $s_1 := \text{true}, s_2 := \text{true}$
- $z_{10}$
- $z_{11}$
- $z_{12}$
- $z_{13}$

- $h \leq 0 \land v < 0$
- $h > 0 \land v < 0$

- $z_{02}: v = 0$

- $z_{01}: \dot{v} = -g$

- $z_{00}$

- $\neg(h \leq 0 \land v < 0)$
- $\neg(h > 0 \land v < 0)$

- $s_1 := \text{false}$
- $s_2 := \text{false}$

- $b_v < 0$
- $b_h \leq 0$

- $\text{new}(b_v) = -b_e * b_v$
- $b_h = 0$

**Process Task model equations**

- $v = -g$

**Automaton**

- For loops
- For unconditional equations
- For conditional equations
- For sequential statements

- $s_1 := \text{true}$
- $s_2 := \text{true}$
- $s_2 := \text{false}$
- $s_1 := \text{false}$

- $b_v < 0$
- $b_h \leq 0$

- $\text{new}(b_v) = -b_e * b_v$
- $b_h = 0$
Transformations – gPROMS → CIF (Excerpt)

Process → Model → Task

Parallelism provided by inherent parallel automata

Model equations

algorithmic statements

// for unconditional equations

\( z_{01} \)

\( \dot{v} = -g \)

// for conditional equations

\( z_{00} \)

\( h = 0 \land \nu = 0 \)

\( \neg (h = 0 \land \nu = 0) \)

\( h = 0 \land \nu = 0 \)

\( \neg (h = 0 \land \nu = 0) \)

// if-true

\( z_{01} \)

\( \dot{v} = -g \)

// if-false

\( z_{02} \)

\( \dot{v} = 0 \)

// for sequential equations

\( z_{10} \)

\( s_1 = \text{true, } s_2 = \text{true} \)

\( z_{11} \)

\( s_1 = \text{false, } s_2 = \text{false} \)

\( z_{13} \)

\( \text{top true} \)

// for unconditional equations

// for loops

\( z_{20} \)

\( s_1 = \text{true} \)

\( z_{21} \)

\( \text{true} \)

\( z_{22} \)

\( b_h = 0 \)

\( z_{24} \)

\( b_h < 0 \)

\( z_{25} \)

\( \{b_v, b_h\} : \)

\( s_1 = \text{false} \)

\( z_{23} \)

\( \neg \text{true} \)

\( z_{26} \)

\( \neg \text{true} \)

\( \text{new}(b_v) = -b_e \cdot b_v \)

\( b_h = 0 \)
Compositional Interchange Format (CIF)

Modelica, gPROMS
- IPDAE simulation
- Dynamic optimization

Muscod
- CT/DE simulation

Chi
- CT/DE simulation

UPPAAL
- Timed automata verification

SpaceEx, Ariadne
- Hybrid automata verification

CIF:
- Concepts
- Simulation
- External interfaces:
  - Co-simulation
  - Function calls
  - EtherCAT fieldbus
- Refinement:
  - And-Or superstates
  - Stochastics
- CIF to CIF:
  - CIFhybrid → CIFtimed
  - CIFandor → CIFflat
- Eclipse graphical editor

Switched linear systems interchange format

Discrete-time PWA
- Multi-Parametric toolbox
- Hybrid toolbox
- Identification toolboxes

Master-slave cosimulation
- Matlab/Simulink and others

EtherCAT real-time fieldbus

Logic controller design
- SFC, DC/FT

Supervisory Control Synthesis
- Event based
- State based

Programmable Logic Controller
- IEC 61131-3 PLCopen

Industrial applications
- Twins
  - Printer paper path control: Rose RT Statecharts
- Darwin
  - MRI scanner patient support control
- C4C
  - Control of distributed printing processes

http://se.wtb.tue.nl/sewiki/cif/start
Compositional Interchange Format (CIF)

References:


Design Flow: Plant Layout Design / Scheduling

Boderc key drivers

Timed Chi
- Complete plant
- Timed model
- Simulation

Uppaal
- Complete plant
- Timed Automata
- Scheduling

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=12

Design step
Requirements → Feasibility Analysis → Plant Layout

Design Flow: Plant Layout Design / Scheduling

Requirements
Feasibility Analysis
Plant Layout

Another View
Control View
Structure View

Plant
Chi

Uppaal
Design Flow: Transformation Experiment

Boderc key drivers

Timed Chi
  - Complete plant
  - Timed model
  - Simulation

Uppaal
  - Complete plant
  - Timed Automata
  - Scheduling

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=12

Experiment:

Set input:
- Model parameters
- Tool parameters
- Chi model

Select tool:
- Tool chain: Chi-to-Uppaal transformation

Save results:
- Uppaal model file
Design Flow: Transformation Experiment

Experiment:
Set input:

Model parameters
Tool parameters
Chi model

Boderc key drivers

Timed Chi
- Complete plant
- Timed model
- Simulation

Uppaal
- Complete plant
- Timed Automata
- Scheduling

System configuration:
- # Stations = 5
- Station Types = [5x1]
- Station Positions = [5x2]
- # AGVs = 3
- Recipe List = [6x1]
- Movement Times = [5x5]
- Recipes/Day = 12
Design Flow: Plant Layout Design / Scheduling

Boderc key drivers

Timed Chi
- Complete plant
- Timed model
- Simulation

Uppaal
- Complete plant
- Timed Automata
- Scheduling

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=12

Experiment:

Set input:

Select tool:

Save results:

Model parameters

Tool parameters

Uppaal model

Uppaal

Uppaal scheduling

Uppaal trace file
### Design Flow: Plant Layout Design / Scheduling

**Boderc key drivers**
- Complete plant
- Timed model
- Simulation

**System configuration:**
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]

**Experiment:**
- Set input:
  - Model parameters
  - Tool parameters

**Select tool:**
- Uppaal
- Boderc

**Uppaal trace file**

**Uppaal scheduling**

**Experiment:**
- Design Flow: Plant Layout Design / Scheduling

**Model configuration:**
- Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=12

**System configuration**

<table>
<thead>
<tr>
<th>No</th>
<th>Device</th>
<th>Duration</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AGV 1</td>
<td>3.00</td>
<td>Drive x y velocity</td>
</tr>
<tr>
<td>2</td>
<td>AGV 1</td>
<td>9.00</td>
<td>Dock</td>
</tr>
<tr>
<td>3</td>
<td>AGV 1</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>4</td>
<td>HAR</td>
<td>1.00</td>
<td>Arm UP</td>
</tr>
<tr>
<td>5</td>
<td>HAR</td>
<td>10.00</td>
<td>Dock</td>
</tr>
<tr>
<td>6</td>
<td>HAR</td>
<td>1.00</td>
<td>Arm DOWN</td>
</tr>
<tr>
<td>7</td>
<td>HAR</td>
<td>1.00</td>
<td>Release Vessel</td>
</tr>
<tr>
<td>8</td>
<td>HAR</td>
<td>1.00</td>
<td>Arm UP</td>
</tr>
<tr>
<td>9</td>
<td>HAR</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>10</td>
<td>HAR</td>
<td>2.00</td>
<td>UNDock</td>
</tr>
<tr>
<td>11</td>
<td>AGV 1</td>
<td>2.00</td>
<td>UNDock</td>
</tr>
<tr>
<td>12</td>
<td>AGV 1</td>
<td>5.00</td>
<td>Drive x y velocity</td>
</tr>
<tr>
<td>13</td>
<td>AGV 1</td>
<td>9.00</td>
<td>Dock</td>
</tr>
<tr>
<td>14</td>
<td>AGV 1</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>15</td>
<td>COL 1</td>
<td>10.00</td>
<td>Dock</td>
</tr>
<tr>
<td>16</td>
<td>COL 1</td>
<td>4.00</td>
<td>Fill</td>
</tr>
<tr>
<td>17</td>
<td>COL 1</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>18</td>
<td>COL 1</td>
<td>2.00</td>
<td>UNDock</td>
</tr>
<tr>
<td>19</td>
<td>AGV 1</td>
<td>2.00</td>
<td>UNDock</td>
</tr>
<tr>
<td>20</td>
<td>AGV 1</td>
<td>13.00</td>
<td>Drive x y velocity</td>
</tr>
<tr>
<td>21</td>
<td>AGV 1</td>
<td>9.00</td>
<td>Dock</td>
</tr>
<tr>
<td>22</td>
<td>AGV 1</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>23</td>
<td>MIX 1</td>
<td>10.00</td>
<td>Dock</td>
</tr>
<tr>
<td>24</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Arm DOWN</td>
</tr>
<tr>
<td>25</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Lock Vessel</td>
</tr>
<tr>
<td>26</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Arm UP</td>
</tr>
<tr>
<td>27</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Fill</td>
</tr>
<tr>
<td>28</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Mix</td>
</tr>
<tr>
<td>29</td>
<td>MIX 2</td>
<td>45.00</td>
<td>Fill Mix</td>
</tr>
<tr>
<td>30</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>31</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Arm DOWN</td>
</tr>
<tr>
<td>32</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Release Vessel</td>
</tr>
<tr>
<td>33</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Arm UP</td>
</tr>
<tr>
<td>34</td>
<td>MIX 1</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>35</td>
<td>MIX 1</td>
<td>2.00</td>
<td>UNDock</td>
</tr>
<tr>
<td>36</td>
<td>AGV 2</td>
<td>7.00</td>
<td>Drive x y velocity</td>
</tr>
<tr>
<td>37</td>
<td>AGV 2</td>
<td>9.00</td>
<td>Dock</td>
</tr>
<tr>
<td>38</td>
<td>AGV 2</td>
<td>1.00</td>
<td>Stop</td>
</tr>
<tr>
<td>39</td>
<td>HAR</td>
<td>10.00</td>
<td>Dock</td>
</tr>
<tr>
<td>40</td>
<td>HAR</td>
<td>1.00</td>
<td>Traverse</td>
</tr>
<tr>
<td>41</td>
<td>HAR</td>
<td>1.00</td>
<td>Arm DOWN</td>
</tr>
<tr>
<td>42</td>
<td>HAR</td>
<td>1.00</td>
<td>Lock Vessel</td>
</tr>
<tr>
<td>43</td>
<td>HAR</td>
<td>1.00</td>
<td>Arm UP</td>
</tr>
<tr>
<td>44</td>
<td>HAR</td>
<td>1.00</td>
<td>Traverse</td>
</tr>
<tr>
<td>45</td>
<td>HAR</td>
<td>1.00</td>
<td>Arm DOWN</td>
</tr>
<tr>
<td>46</td>
<td>HAR</td>
<td>1.00</td>
<td>Release Vessel</td>
</tr>
</tbody>
</table>
**Design Flow: Plant Layout Design / Scheduling**

**Boderc key drivers**
- Timed Chi
  - Complete plant
  - Timed model
  - Simulation
- Uppaal
  - Complete plant
  - Timed Automata
  - Scheduling

**System configuration:**
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- **Recipes/Day=14**

**Experiment:**

**Set input:**
- Model parameters
- Tool parameters
- **Uppaal model**

**Select tool:**
- Uppaal

**Save results:**
- Uppaal scheduling
- **Uppaal trace file**
Design Flow: Plant Layout Design / Scheduling

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

**Models**
- Boderc keydrivers
  - Design parameter & assumptions
- Timed Chi
  - Transformation
- Uppaal
  - Complete plant model
  - Timed Automata model
  - Scheduling

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace

**Multiform**
Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems
## Design Flow: AGV Docking

<table>
<thead>
<tr>
<th>Design tasks</th>
<th>Models</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Boderc keydrivers</td>
<td>First design parameters and assumptions</td>
</tr>
<tr>
<td>Feasibility analysis</td>
<td>Timed Chi</td>
<td>Feasible plant layouts (1 mixing, 2 filling, 2 charging)</td>
</tr>
<tr>
<td>Plant layout design</td>
<td>Uppaal</td>
<td>Cost optimal plant layout and scheduling trace</td>
</tr>
<tr>
<td>Docking</td>
<td>Modelica</td>
<td></td>
</tr>
<tr>
<td>AGV speed analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controller design</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code generation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Models
- **Boderc keydrivers**
  - Design parameter & assumptions

- **Timed Chi**
  - Transformation

- **Uppaal**
  - Movement times

- **Modelica**
  - Detailed AGV & station model
  - Hybrid Dynamics

### Results
- Forwarded information
- Model transformation
Design Flow: AGV Docking

Boderc key drivers

- Timed-Chi
  - Complete plant
  - Timed model
  - Simulation
- Uppaal
  - Complete plant
  - Timed Automata
  - Scheduling
- Modelica
  - Detailed AGV & station model
  - Hybrid Dynamics

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=14

- Detailed Model of 1 AGV + 1 station
- Goal:
  - Development of a docking algorithm
  - Speed and movement times for docking
- Modeling Language: Modelica
- Tool: Dymola

63
System configuration:
• # Stations=5
• Station Types=[5x1]
• Station Positions=[5x2]
• # AGVs=3
• Recipe List=[6x1]
• Movement Times=[5x5]
• Recipes/Day=14
• avgAGVSpeed=500

Design Flow: AGV Docking

Boderc key drivers

Timed-Chi
– Complete plant
– Timed model
– Simulation

Uppaal
– Complete plant
– Timed Automata
– Scheduling

Modelica
– Detailed AGV & station model
– Hybrid Dynamics

Extract parameters:
• Average AGV speed
Design Flow: AGV Docking Experiment

Boderc key drivers

Timed-Chi
- Complete plant
  - Timed model
  - Simulation

Uppaal
- Complete plant
  - Timed Automata
  - Scheduling

Modelica
- Detailed AGV & station model
  - Hybrid Dynamics

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=14
- avgAGVS=500

Experiment:

Set input:

Select tool:

Dymola

Save results:

Simulation results

Model parameters

Tool parameters

Modelica model

Modelica
Design Flow: AGV Docking Experiment

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- **Movement Times**=[5x5]
- Recipes/Day=14
- avgAGVSpeed=500

Boderc key drivers
- Timed-Chi
  - Complete plant
  - Timed model
  - Simulation
- Uppaal
  - Complete plant
  - Timed Automata
  - Scheduling
- Modelica
  - Detailed AGV & station model
  - Hybrid Dynamics

Experiment:
- Set input:
  - Model parameters
  - Tool parameters
  - Modelica model
- Select tool: Dymola
- Save results: Simulation results

(*inserted image of integrated multi-formalisim tool support*)
Design Flow: AGV Docking

System configuration:
- # Stations = 5
- Station Types = [5x1]
- Station Positions = [5x2]
- # AGVs = 3
- Recipe List = [6x1]
- Movement Times = [5x5]
- Recipes/Day = 14
- avgAGVSpeed = 500

Boderc key drivers
- Timed-Chi
  - Complete plant
  - Timed model
  - Simulation
- Uppaal
  - Complete plant
  - Timed Automata
  - Scheduling
- Modelica
  - Detailed AGV & station model
  - Hybrid Dynamics

Design step
- Requirements
- Feasibility
- Analysis
- Docking

Another View
- Control View
- Structure View

Modelica
- Detailed AGV & station model
- Hybrid Dynamics

Plant
- Chi
- Uppaal
- Modelica

Stations + AGVs
1. Rerun experiments

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=14
- avgAGVSspeed=500

Design Flow: Parameter Propagation

Boderc key drivers

Timed-Chi
- Complete plant
- Timed model
- Simulation

Upaall
- Complete plant
- Timed Automata
- Scheduling

Modelica
- Detailed AGV & station model
- Hybrid Dynamics

Plant

Stations + AGVs

Chi

Uppaal

Modelica
1. Rerun experiments
2. Update results

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=13
- avgAGVSpeed=500
1. Rerun experiments

2. Update results

Design Flow: Parameter Propagation

System configuration:
• # Stations=5
• Station Types=[5x1]
• Station Positions=[5x2]
• # AGVs=3
• Recipe List=[6x1]
• Movement Times=[5x5]
• Recipes/Day=13
• avgAGVSpeed=500
Design Flow: Parameter Propagation

**Design tasks**
- Requirements
  - Feasibility analysis
  - Plant layout design
    - Docking
  - AGV speed analysis
  - Validation
  - Controller design
  - Code generation

**Models**
- Boderc keydrivers
  - Design parameter & assumptions
- Timed Chi
  - Transformation
  - Schedules
- Uppaal
  - Movement times
  - Docking time
- Modelica
  - Detailed AGV & station model
  - Hybrid Dynamics

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)

**multiform**

Forwarded information
Feedback information
Model transformation
Design Flow: AGV Speed Analysis

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking

**Models**
- Boderc keydrivers
  - Design parameter & assumptions
  - Timed Chi
    - Transformation
    - Uppaal
      - Movement times
      - Docking time
  - Modelica
    - Speed & acceleration
    - gPROMS
      - Hybrid dynamics, PDE models
      - Optimization

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)

**Multiform**
Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems
Design Flow: AGV Speed Analysis

Boderc key drivers

Timed-Chi
- Complete plant
- Timed model
- Simulation

Uppaal
- Complete plant
- Timed Automata
- Scheduling

Modelica
- Detailed AGV & station model
- Hybrid Dynamics
- Dynamics
- Optimization

gPROMS

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=13
- avgAGVSpeed=500

Requirements

Feasibility Analysis

Design step

Another View

Control View

Structure View

Plant

Stations + AGVs

Modelica

AGVs

gPROMS

Uppaal

Chi

Speed analysis

Model of AGV movement and fluid dynamics

multiform
Integrated Multi-formation Tool Support for the Design of Networked Embedded Control Systems
System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=13
- avgAGVSPEED=500

Optimization of acceleration profile with constraints:
Avoidance of spilling during transport
Design Flow: $gPROMS$ Experiment

**Set input:**

- **Model parameters**
  - $gPROMS$ model

- **Tool parameters:**
  - Maximize $\text{avgAGVSpeed}$
  - Without spilling liquid

**Select tool:**

- $gPROMS$

**Save results:**

- $gPROMS$ optimization results

**System configuration:**

- # Stations = 5
- Station Types = [5x1]
- Station Positions = [5x2]
- # AGVs = 3
- Recipe List = [6x1]
- Movement Times = [5x5]
- Recipes/Day = 13
- $\text{avgAGVSpeed} = 432$

**Boderc key drivers**

- **Timed-Chi**
  - Complete plant
  - Timed model
  - Simulation

- **Uppaal**
  - Complete plant
  - Timed Automata
  - Scheduling

- **Modelica**
  - Detailed AGV & station model
  - Hybrid Dynamics
  - Dynamics
  - Optimization
1. Rerun experiments

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=13
- avgAGVSpeed=432
System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=13
- avgAGVSspeed=432

1. Rerun experiments
2. Update results
1. Rerun experiments

System configuration:
- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- **Movement Times=[5x5]**
- Recipes/Day=13
- avgAGVSspeed=432

Boderc key drivers

- Timed-Chi
  - Complete plant
  - Timed model
  - Simulation

- Uppaal
  - Complete plant
  - Timed Automata
  - Scheduling

- Modelica
  - Detailed AGV & station model
  - Hybrid Dynamics
  - Dynamics
  - Optimization

- gPROMS

Design Flow: Parameter Propagation
System configuration:

- # Stations=5
- Station Types=[5x1]
- Station Positions=[5x2]
- # AGVs=3
- Recipe List=[6x1]
- Movement Times=[5x5]
- Recipes/Day=11
- avgAGVS=432

1. Rerun experiments
2. Update results
Design Flow: Parameter Propagation

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

**Models**
- **Boderc keydrivers**
  - Design parameter & assumptions
- **Timed Chi**
  - Transformation
- **Uppaal**
  - Movement times
- **Modelica**
  - Speed & acceleration
  - Maximum speed & acceleration
- **gPROMS**
  - Hybrid dynamics, PDE models
  - Optimization

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)

**Forwarded information**
- Feedback information
- Model transformation
Design Flow: Validation by Simulation

Design tasks

- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

Models

- Boderc keydrivers
  - Design parameter & assumptions
  - Timed Chi
  - Transformation
  - Uppaal
    - Plant layout & schedule trace
    - Movement times
    - dock time
    - Cost optimal plant layout and scheduling trace
  - Modelica
    - Model composition
    - Hybrid dynamics
    - Complete plant
    - Speed & acceleration
    - Maximum speed & acceleration
- Modelica
  - Model composition
  - Hybrid dynamics
  - Complete plant

Results

- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)

Forwarded information
Feedback information
Model transformation
Model fragmentation / composition
Design Flow: Validation by Simulation

- **Modelica** Model of the complete plant
- Composed from previous specialized models
- Goal:
  - Simulation & visualization of the complete plant
  - Integration of previous results
- Tools: *Dymola* & self developed Process Control Framework
Design Flow: Validation by Simulation

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

**Models**
- Boderc keydrivers
  - Design parameter & assumptions
  - Timed Chi
    - Transformation
    - Uppaal
      - Movement times
      - Docking time
    - Modelica
      - Maximum speed & acceleration
      - Validation
    - gPROMS
      - Speed & acceleration
      - Maximum speed & acceleration
    - Modelica
      - Model-composition
        - Hybrid dynamics
        - Complete plant

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)
- Visualization and validation

**Forwarded information**
- Feedback information
- Model transformation
- Model fragmentation / composition
Design Flow: Validation by Verification

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

**Models**
- Boderc keydrivers
- Timed Chi Transformation
- Uppaal
- Modelica
- gPROMS
- SpaceEx

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)
- Visualization and validation

**Integrated Multi-formalism Tool Support for the Design of Networked Embedded Control Systems**
Verification

• Verification framework *SpaceEx* [Goran Frehse, Rajarshi Ray]
  – Extensible plug-in framework for reachability analysis
  – Java-based GUI editor, web-based architecture
    • Analysis on a web server controlled via a web browser
  – Currently supports reachability algorithms based on *PHAVer* and on support functions
  – Scalability: Experimental results for >100 variables (before: <5)
Verification (3)

- 28th order Helicopter with Disturbance Rejection Controller

Design Flow: Controller Design

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

**Models**
- Boderc keydrivers
- Timed Chi
- Uppaal
- Modelica
- gPROMS
- SpaceEx

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)
- Visualization and validation
- Docking controller validation

**Forwarded information**
- Feedback information
- Model transformation
- Model fragmentation / composition
Logic Controller Design and Synthesis

- Industrial practice: Logic controller design is an often manual and error-prone procedure.

- Benefits of systematic specification refinement and analysis:
  - Improved and systematic communication between the plant designer and the control engineer → less “misunderstandings” and errors.
  - Algorithmic requirements analysis → error avoidance early in the design phase.

- Benefits of the synthesis tool chain:
  - Largely automated design procedure.
  - Comfortable choice between different synthesis approaches.

Informal and vague specifications → Systematic analysis → Refinement → Formal and precise specifications → Algorithmic synthesis → Control system → Plant model.
Logic Controller Design by Refinement [TU DO]

- Systematic and hierarchical approach
  - **Dependency Chart (DC)**
    - Conceptional design
    - Similar to Gantt chart
    - Hierarchy – DCs may contain DCs
    - On the lowest level, a DC is specified by a Function Table
  
- **Function Table (FT)**
  - Detailed description
  - List of sequential actions
  - Documentation of design decisions

<table>
<thead>
<tr>
<th>Precondition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Informal</td>
<td>Formal</td>
</tr>
<tr>
<td>Informal</td>
<td>Formal</td>
</tr>
<tr>
<td><strong>Tank level is to high</strong></td>
<td>H &gt; H_max</td>
</tr>
</tbody>
</table>
Logic Controller Design by Refinement (2)

- Systematic refinement of informal specifications into logic controllers

**DC/FT view**

**SFC view**
Logic Controller Design by Refinement (3)

• Tool chains

SFC → CIF

SFC → Uppaal
Design Flow: Controller Design

Design tasks
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

Models
- Boderc keydrivers
- Uppaal
- Modelica
- gPROMS
- SpaceEx
- DC/FT

Results
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)
- Visualization and validation
- Docking controller validation
- Controller specification

Forwarded information
- Feedback information
- Model transformation
- Model fragmentation / composition
Design Flow: Code Generation

Design tasks
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

Models
- Boderc keydrivers
- Timed Chi
- Uppaal
- Modelica
- gPROMS
- SpaceEx
- DC/FT
- PLC Code

Results
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)
- Visualization and validation
- Docking controller validation
- Controller specification
- Controller Code

Forwarded information
Feedback information
Model transformation
Model fragmentation / composition
Design Flow: Code Verification

**Design tasks**
- Requirements
- Feasibility analysis
- Plant layout design
- Docking
- AGV speed analysis
- Validation
- Controller design
- Code generation

**Models**
- Boderc keydrivers
- Timed Chi
- Uppaal
- Modelica
- gPROMS
- Modelica
- SpaceEx
- DC/FT
- PLC Code

**Results**
- First design parameters and assumptions
- Feasible plant layouts (1 mixing, 2 filling, 2 charging)
- Cost optimal plant layout and scheduling trace
- Docking time (10 s) and optimal station angle (90°)
- Maximum speed (500 mm/s) and acceleration (500 mm/s²)
- Visualization and validation
- Docking controller validation
- Controller specification
- Controller Code
- Verified Code

**Additional notes**
- Model transformation
- Feasible plant layouts
- Controller design
- AGV speed analysis
- Validation
- Code generation
- Integrated Multi-formation Tool Support for the Design of Networked Embedded Control Systems

**Key terms**
- Design parameter & assumptions
- Movement times
- Docking time
- Speed & acceleration
- Maximum speed & acceleration
- Validation
- Errors, Warnings, Etc.
- Code generation
Summary

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models.
Summary

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models.

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, so several tools and modeling formalisms must be used in the design process.
Summary

• There is a need for efficient model-based support of the design of complex automated systems with trans-level propagation and iteration, and re-use of models

• An all-encompassing mega-tool for the design of complex automated systems is not realistic, so several tools and modeling formalisms must be used in the design process.

• Three different routes to tool and model integration and design support were pursued in MULTIFORM:
  – Model exchange and tool chains via the CIF
  – Direct coupling of tools for testing of specifications
  – Propagation of parameters via the Design Framework
Lessons and Challenges from MULTIFORM

- The *CIF* and its tool set are stable and relatively mature
- Available under open source licence
- The effort for developing model transformations is high
- Transformations from the CIF in most cases can only be performed for subsets of the models which can be represented in the formalism.
  - A **formal** specification of the the supported *CIF* subset of a tool is needed
- It should be possible to trace elements of a model after the transformation
- Model blow-up is not as bad as could be expected
Lessons and Challenges from MULTIFORM (2)

- The CIF is very expressive and well suited for model exchange between automata-based formalisms, but conceptually different from equation-oriented languages (e.g. Modelica, gPROMS, EcosimPro)
  - **Possible solution:** Use a Modelica subset as an exchange formalism for equation-oriented languages, bridge equation- and automata-oriented formalisms via the CIF $\leftrightarrow$ Modelica transformation

- Often only some elements of a system are modeled precisely, and these models are formulated in different formalisms (*fragmented modeling*)
  - How can the interdependencies between model fragments be **formally** described and exploited?

- **Model ontology needed**
  - Specification of model formalism expressivity using a common formal vocabulary
  - Equipping model artifacts with meta data on their origin(s) $\rightarrow$ traceability
  - Description of relations of partial models to an overall model
THANK YOU VERY MUCH FOR YOUR ATTENTION!

QUESTIONS, PLEASE!
Equation-based vs. Automaton-based Formalisms

- Simulation/Solver/Tool options encoded in model code (e.g. EcosimPro, gPROMS)
  - Tool specific options cannot be transformed  
    ➞ Other tools might not find a solution for a difficult initialization problem

- Formal semantics not available  ➞ Transformation not provably correct
Equation-based vs. Automaton-based Formalisms

- Simulation/Solver/Tool options encoded in model code (e.g. EcosimPro, gPROMS)
  - Tool specific options cannot be transformed ➔ Other tools might not find a solution for a difficult initialization problem

- Formal semantics not available ➔ Transformation not provably correct

- Equation-based models can be more restrictive than automata models
  - E.g. Modelica enforces globally and locally balanced models ➔ Automata models need to be preprocessed
Equation-based vs. Automaton-based Formalisms

- Simulation/Solver/Tool options encoded in model code (e.g. EcosimPro, gPROMS)
  - Tool specific options cannot be transformed
    ➔ Other tools might not find a solution for a difficult initialization problem

- Formal semantics not available ➔ Transformation not provably correct

- Equation-based models can be more restrictive than automata models
  - E.g. Modelica enforces globally and locally balanced models
    ➔ Automata models need to be preprocessed
    • Either by flattening of the model
    • Or by rebuilding the automata structure in an equation-based formalism
Design Framework

- Design Flow Support with the *Design Framework*
  - The consortium has shown that the DF concept is well suited for different design flows and supports both, low-level data and tool integration and high-level design flow support as well as data management
Design Framework

• Design Flow Support with the *Design Framework*
  – The consortium has shown that the DF concept is well suited for different design flows and supports both, low-level data and tool integration and high-level design flow support as well as data management
  – **Challenge:** For well-defined, bottom-up design flows, it is necessary to seamlessly integrate the design flow with the existing legacy engineering tool infrastructure
    • Design flow support directly provided by those engineering tools that are applied in the engineering process
    • Design flow graphs accessible from all parts of the engineering system *on request* (hidden during normal operation)
Design Framework

- **Design Flow Support with the *Design Framework***
  - The consortium has shown that the DF concept is well suited for different design flows and supports both, low-level data and tool integration and high-level design flow support as well as data management
  - **Challenge:** For well-defined, bottom-up design flows, it is necessary to seamlessly integrate the design flow with the existing legacy engineering tool infrastructure
    - Design flow support directly provided by those engineering tools that are applied in the engineering process
    - Design flow graphs accessible from all parts of the engineering system **on request** (hidden during normal operation)
  - Future goals of TUDO and *euTeXoo*
    - Seamless integration of design flow support and tool integration with the existing infrastructures