Semantics for Rollback-Based Continuous/Discrete Simulation

Luiza Gheorghe,
Gabriela Nicolescu, Hanifa Boucheneb
Ecole Polytechnique de Montréal
Tel : (514) 340 4711 ext 5434
Fax: (514) 340 3240
Email : gabriela.nicolescu@polymtl.ca
Continuous/Discrete Systems

- Applications to various domains
  - Defense, medical, communications, automotive, ...

- Examples of Continuous/Discrete Systems
  - MEMS, real-time controllers, mixed-signal systems …

- Main characteristics
  - Complexity, heterogeneity

- Main design challenges
  - Global specification and validation
Continuous/Discrete Systems Design

- Collaboration between different teams
- Incremental refinements through different abstraction levels with specific execution models
- Validation requires joint execution of heterogeneous execution models
  - Co-Simulation Technique
Challenges for Continuous/Discrete Co-Simulation

- Defining new tools facilitating cooperation between different teams
  - Enabling easy specification, automatic generation for simulation interfaces
  - Taking into account implementation choices
  - Exploiting powerful existing tools (Simulink, SystemC, …)
  - Based on a single well defined formalism for domain interaction
Contributions

- Definition of the semantics for the continuous and the discrete co-simulation interfaces
  - The interfaces representation using DEVS models [University of Arizona] and timed automata

- Formal verification of the simulation interfaces
  - Study for rollback-based continuous/discrete simulation models
Outline

- Global Simulation for Continuous/Discrete Systems
- **Design Methodology for Continuous/Discrete Systems Co-Simulation Tools**
- Semantics for Rollback-based Continuous/Discrete Co-Simulation
- Formal verification for Co-Simulation Interfaces
- Conclusion
Design Methodology for Continuous/Discrete Simulation Tools

Generic Stage

- Definition of the synchronization operational semantic
- Distribution of the synchronization functionality to sim. interfaces
- Interfaces behavior formalization and verification
- Definition of the internal architecture and simulation library

Implementation stage

- Sim. tools analysis
- Library elements implementation
- Implementation validation

Discrete

Continuous
Design Methodology for Continuous/Discrete Simulation Tools

Generic Stage
- Definition of the synchronization operational semantic
- Distribution of the synchronization functionality to sim. interfaces
- Interfaces behavior formalization and verification
- Definition of the internal architecture and simulation library

Implementation stage
- Sim. tools analysis
- Library elements implementation
- Implementation validation

Discrete
Simulation Interface

Continuous
Simulation Interface

Simulation Bus

BMAS 2008
Design Methodology for Continuous/Discrete Simulation Tools

Generic Stage

1. Definition of the synchronization operational semantic
2. Distribution of the synchronization functionality to sim. interfaces
3. Interfaces behavior formalization and verification
4. Definition of the internal architecture and simulation library

Implementation stage

- Sim. tools analysis
- Library elements implementation
- Implementation validation

Discrete

- Layer 1
- Layer 2
- Layer 3
- ... (repeated)
- Layer N

Continuous

- Layer 1
- Layer 2
- Layer 3
- ... (repeated)
- Layer N

Simulation Bus

BMAS 2008
Design Methodology for Continuous/Discrete Simulation Tools

Generic Stage

1. Definition of the synchronization operational semantic
2. Distribution of the synchronization functionality to sim. interfaces
3. Interfaces behavior formalization and verification
4. Definition of the internal architecture and simulation library

Implementation stage

Sim. tools analysis → Library elements implementation → Implementation validation

Discrete
Continuous

Sim. Library

Discrete Interface
Continuous Interface
Simulation Bus

BMAS 2008
Design Methodology for Continuous/Discrete Simulation Tools

Generic Stage
- Definition of the synchronization operational semantic
- Distribution of the synchronization functionality to sim. interfaces
- Interfaces behavior formalization and verification
- Definition of the internal architecture and co-simulation library

Implementation stage
- Sim. tools analysis
- Library elements implementation
- Implementation validation

BMAS 2008
Outline

- Global Simulation for Continuous/Discrete Systems
- Design Methodology for Continuous/Discrete Systems Co-Simulation Tools
- **Semantics for Rollback-based Continuous/Discrete Co-Simulation**
- Formal verification for Co-Simulation Interfaces
- Conclusion

BMAS 2008
Continuous/Discrete Systems
- Global Simulation -

<table>
<thead>
<tr>
<th>Concept Model</th>
<th>Time</th>
<th>Communication means</th>
<th>Processes activation rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>Advances discretely (constant intervals)</td>
<td>Set of events</td>
<td>Processes are sensitive to events</td>
</tr>
<tr>
<td>Continuous</td>
<td>It advances by integration steps (IS)</td>
<td>Piecewise-Continuous signals</td>
<td>Processes are executed at each IS</td>
</tr>
</tbody>
</table>

- Events exchanged between the continuous and the discrete models
  - State events
  - Sampling events
  - Update signal events
Continuous/Discrete Systems - Global Simulation -

Simulation Step | Synchronization

State Event | Signal Update/Sampling Event
Rollback-Based Continuous/Discrete Synchronization Model

Discrete Model

Continuous Model

Signal Update/Sampling Event
Synchronization Operational Semantics

• DEVS (Discrete Event Systems specification) and DESS (Differential Equation Systems Specification) formalisms
  • Set of rules respecting the actions of
    - Discrete model
    - Continuous model
    - Discrete model interface
    - Continuous model interface
  • Timed Automata
    - Classical finite state automata with clock variables and logical formulas on the clocks (temporal constraints)

DEVS/DSS set of rules ➔ Equivalent timed-automata ➔ Simulation & Formal Verification

BMAS 2008
Synchronization Operational Semantics using DEVS

\[ \text{synch} = 1 \land \text{flag} = 0 \land q = \delta_{\text{ext}}(q) \]

\[ q \xrightarrow{(DataFromBus,t_a(s_{\text{dk}}))}\text{;synch:=0} q \]

BMAS 2008
Synchronization Operational Semantics using DEVS

Set of rules for the overall synchronization model
Verification of Simulation Interfaces

- Based on timed-automata – use of UPPAAL tool
Verification of Simulation Interfaces

- Based on timed-automata – use of UPPAAL tool

- Global Verification requires Timed Automata for
  - Discrete Simulator
  - Continuous Simulator
  - The Continuous Simulator Interface
  - The Discrete Simulator Interface
Verification of Simulation Interfaces

- Properties verified
  - Absence of deadlock
  - Timing synchronization
  - Detection of all state events
  - No false state events
  - Respect of causality principle
Verification of Simulation Interfaces

Established direct connection to local server.
UPPAAL version 4.0.2 (rev. 2491), August 2006 -- server.
Disconnected.
Established direct connection to local server.
UPPAAL version 4.0.2 (rev. 2491), August 2006 -- server.
(IDiscrete,StateRestoration and IContinu,StEvDetect) --> (IContinu.tc - IDiscrete.td == 0)
Property is satisfied.
IContinu,StEvDetect --> IDiscrete,StEvDetect
Property is satisfied.
A[]((IDiscrete,Start and IContinu,Start) imply (IContinu.tc - IDiscrete.td <= period))
Property is satisfied.
A[]((IDiscrete,StEvDetect imply StateEvent)
Property is satisfied.
A[] not deadlock.
Property is satisfied.
Conclusions

- Challenges for global validation of continuous/discrete systems
- Definition of global execution models
- Automatic generation of simulation interfaces
- Operational semantics definition and formal representation and verification
- Key step for generalization and library definition