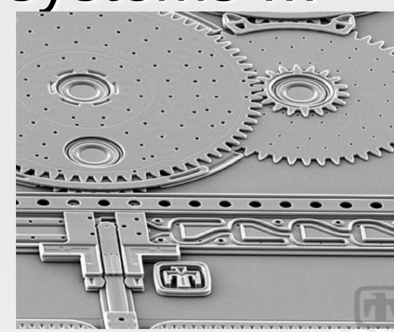
Semantics for Rollback-Based Continuous/Discrete Simulation

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

ÉCOLE POLYTECHNIQUE MONTRÉAL

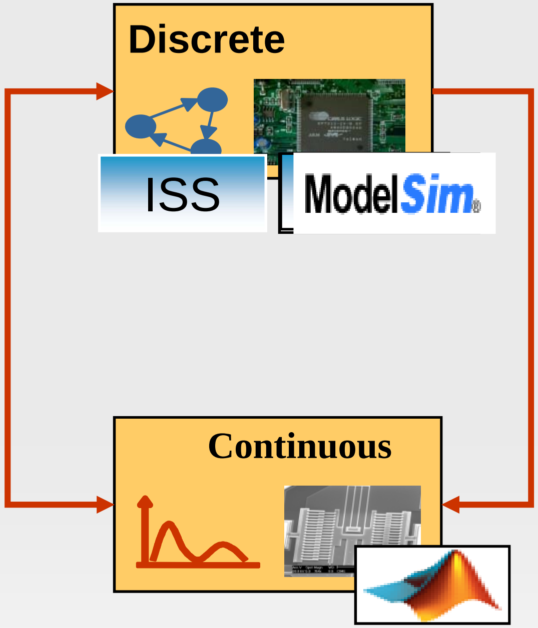
Continuous/Discrete Systems

- 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

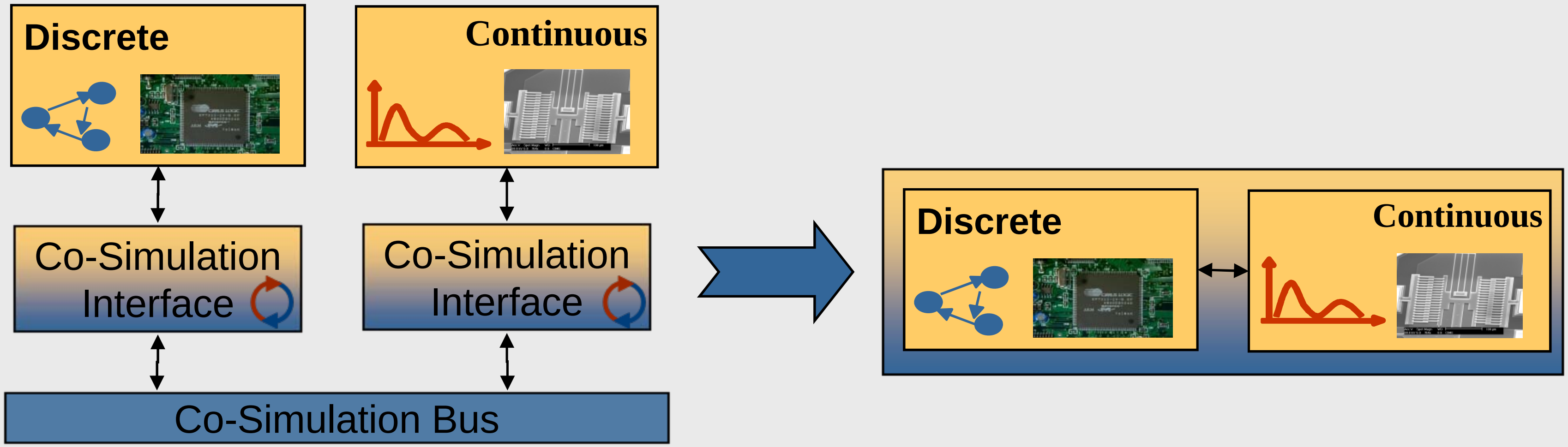


Continuous/Discrete Systems Design

- 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

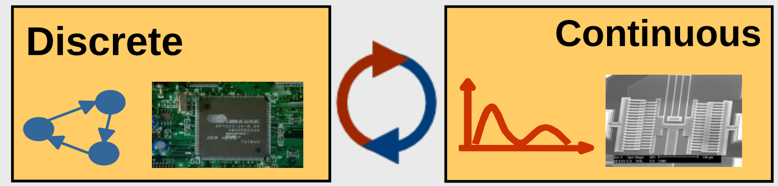
Design Methodology for Continuous/Discrete Simulation Tools

Generic Stage

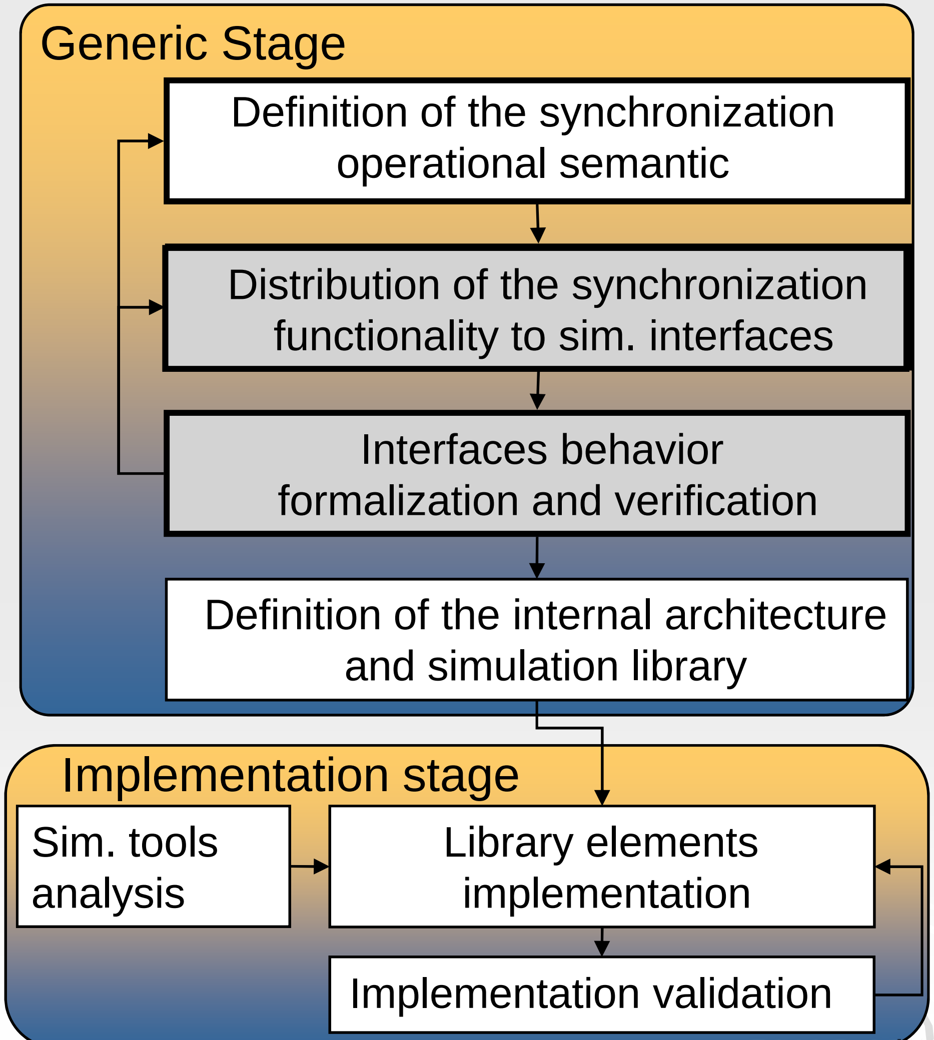
- Definition of the synchronization operational semantic
- Distribution of the synchronization functionality to sim. interfaces
- Interfaces behavior formalization and verification
- Definition of the internal architecture and simulation library

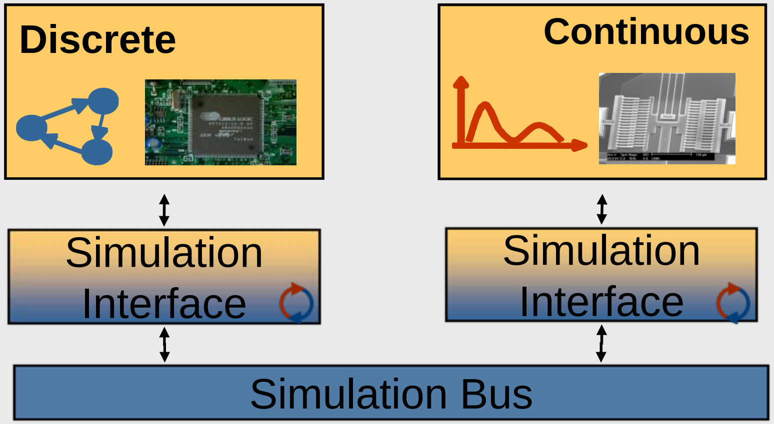
Implementation stage

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

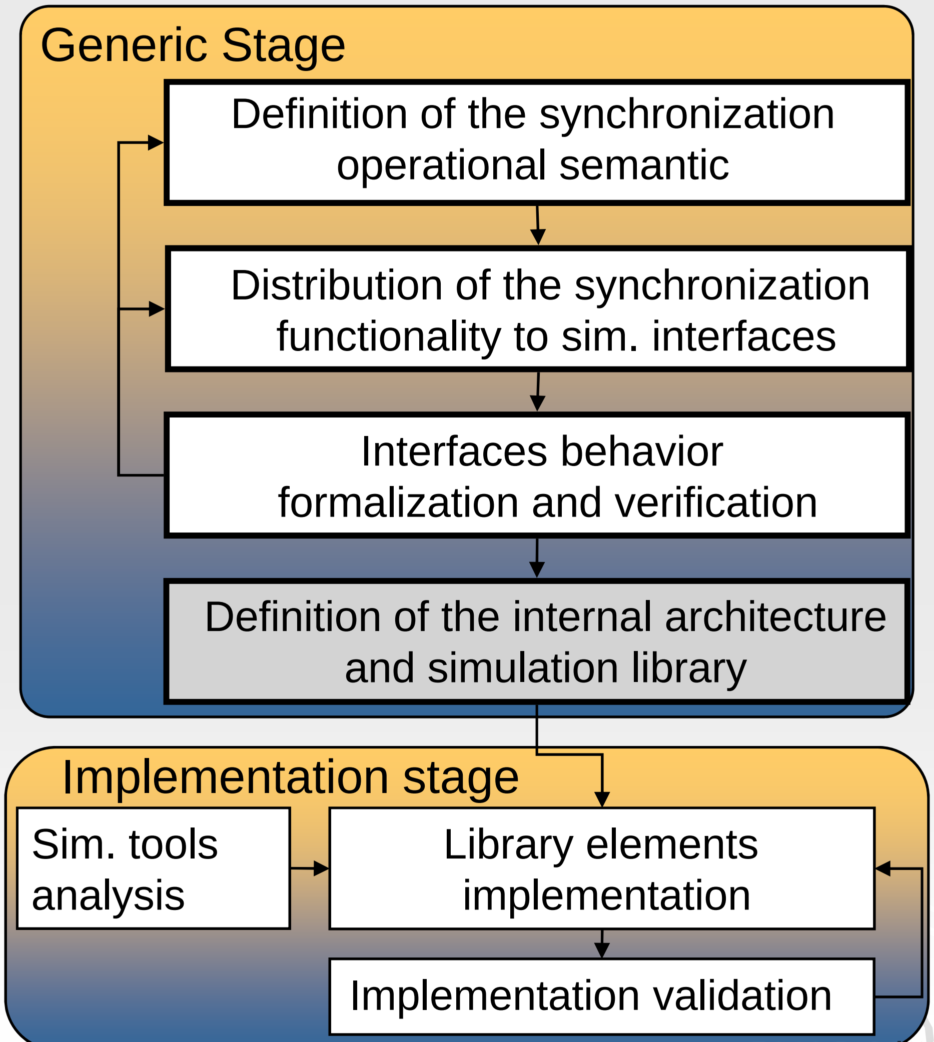


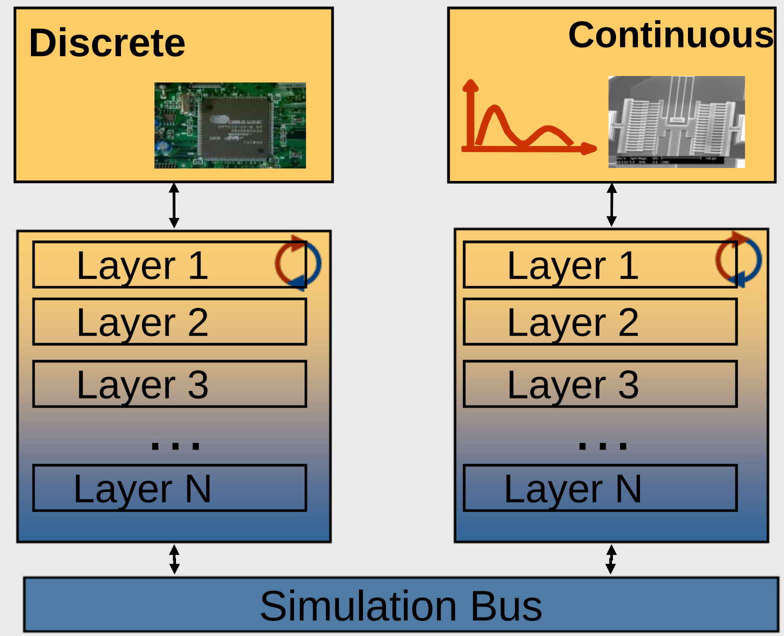
Design Methodology for Continuous/Discrete Simulation Tools



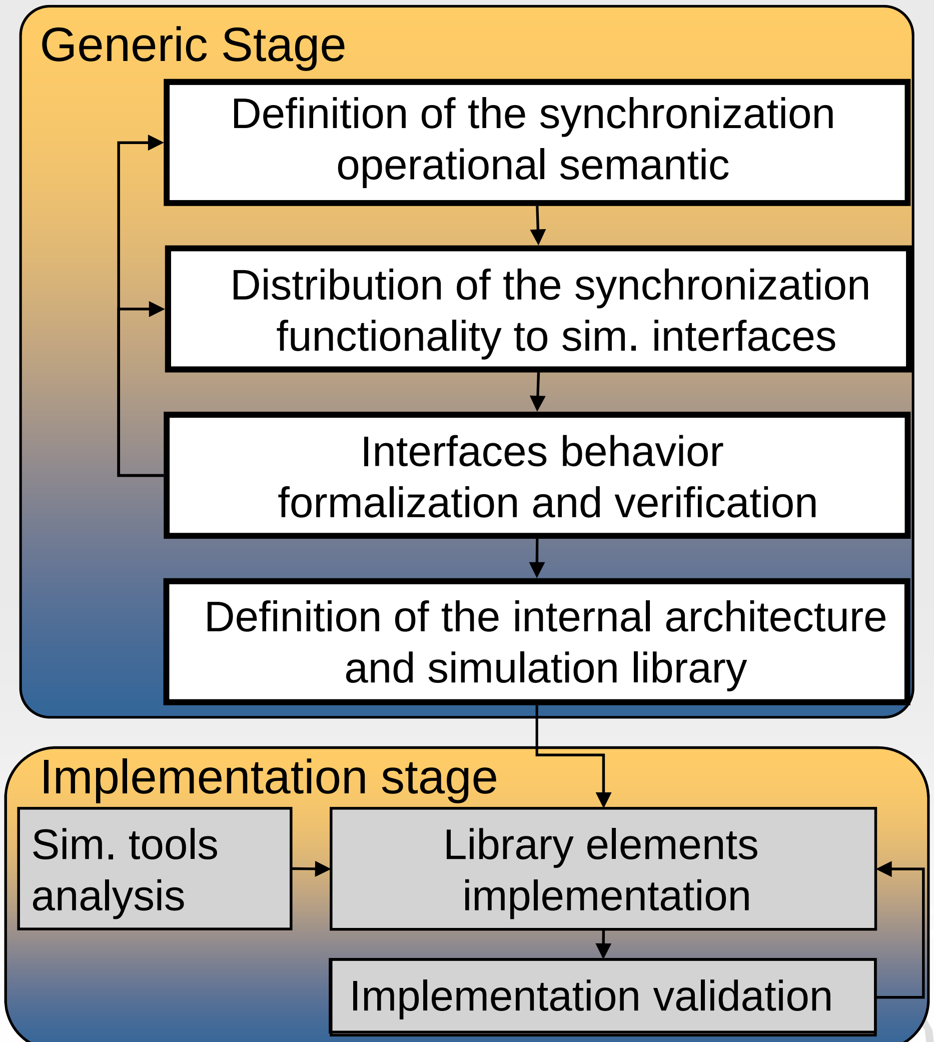


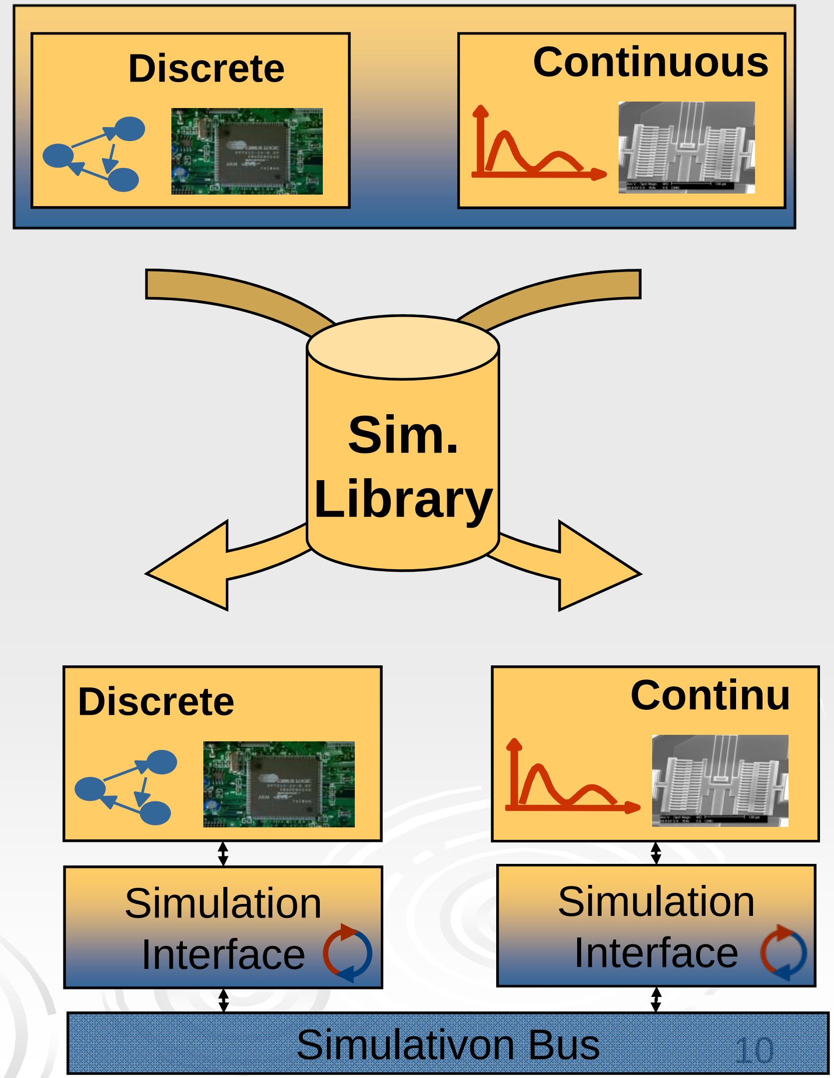
Design Methodology for Continuous/Discrete Simulation Tools



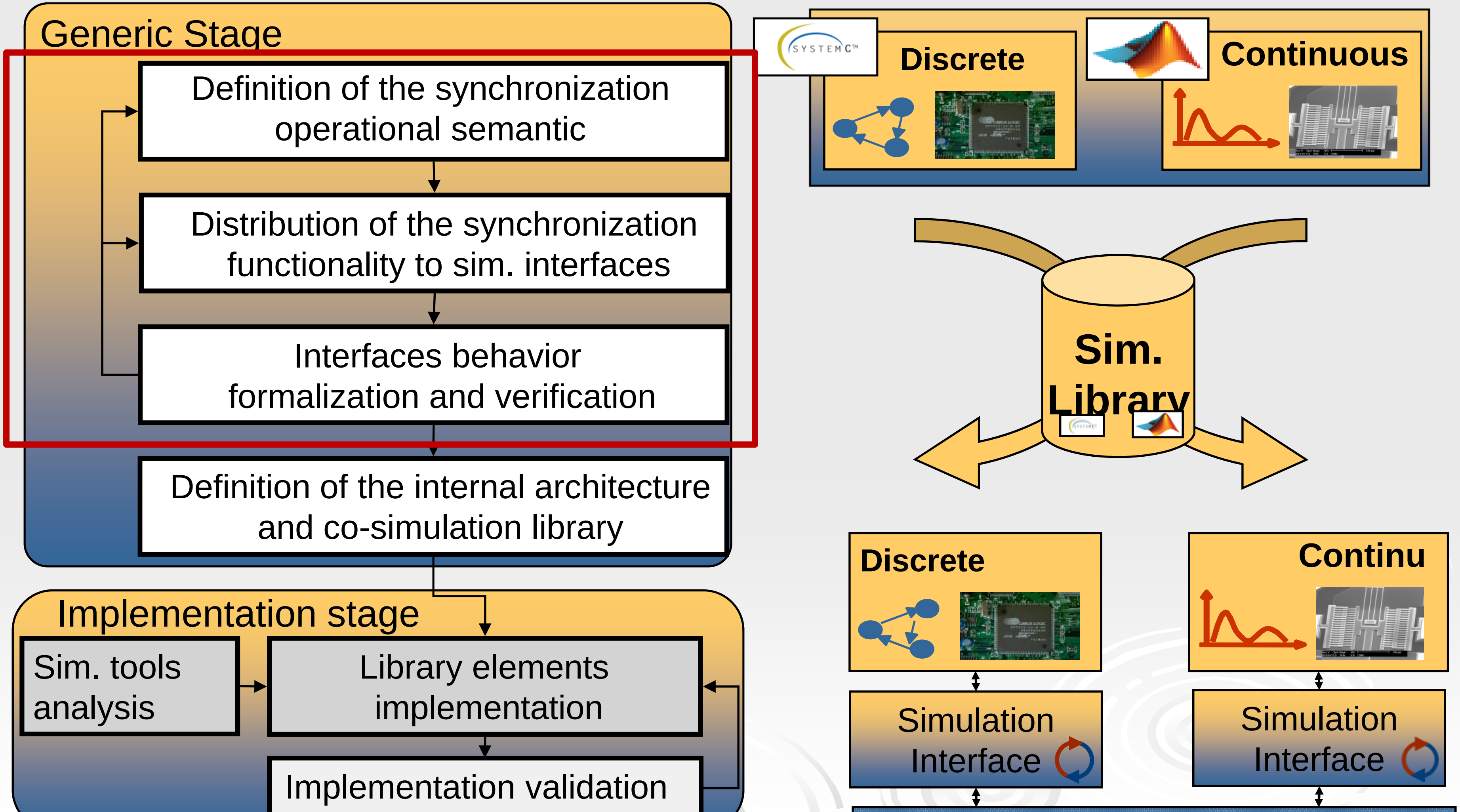


Design Methodology for Continuous/Discrete Simulation Tools





Design Methodology for Continuous/Discrete Simulation Tools



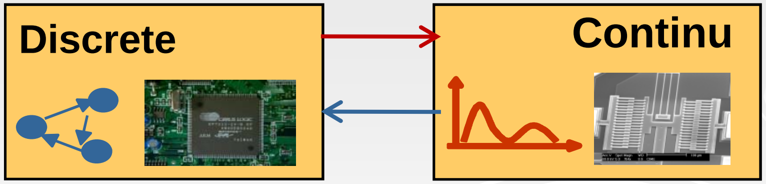
Outline

- 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

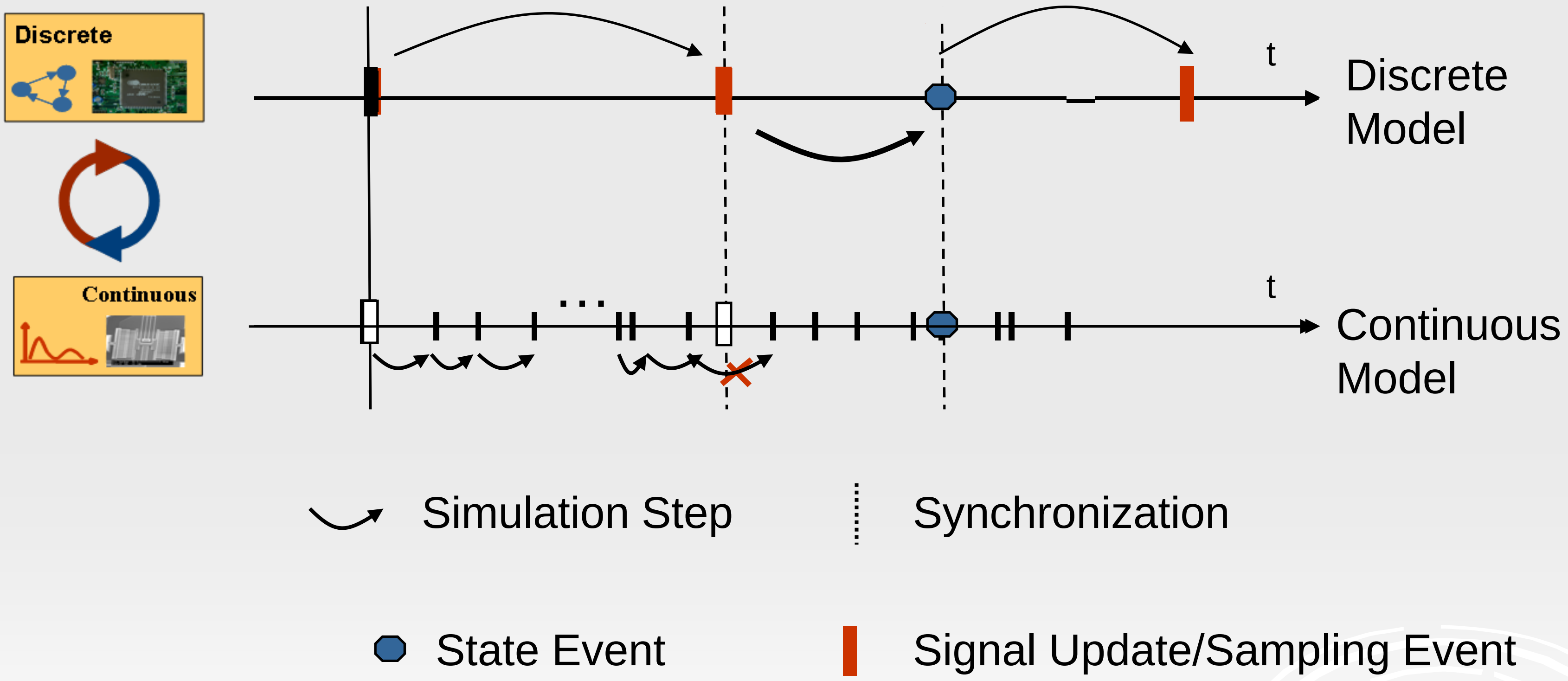
Continuous/Discrete Systems - Global Simulation -

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

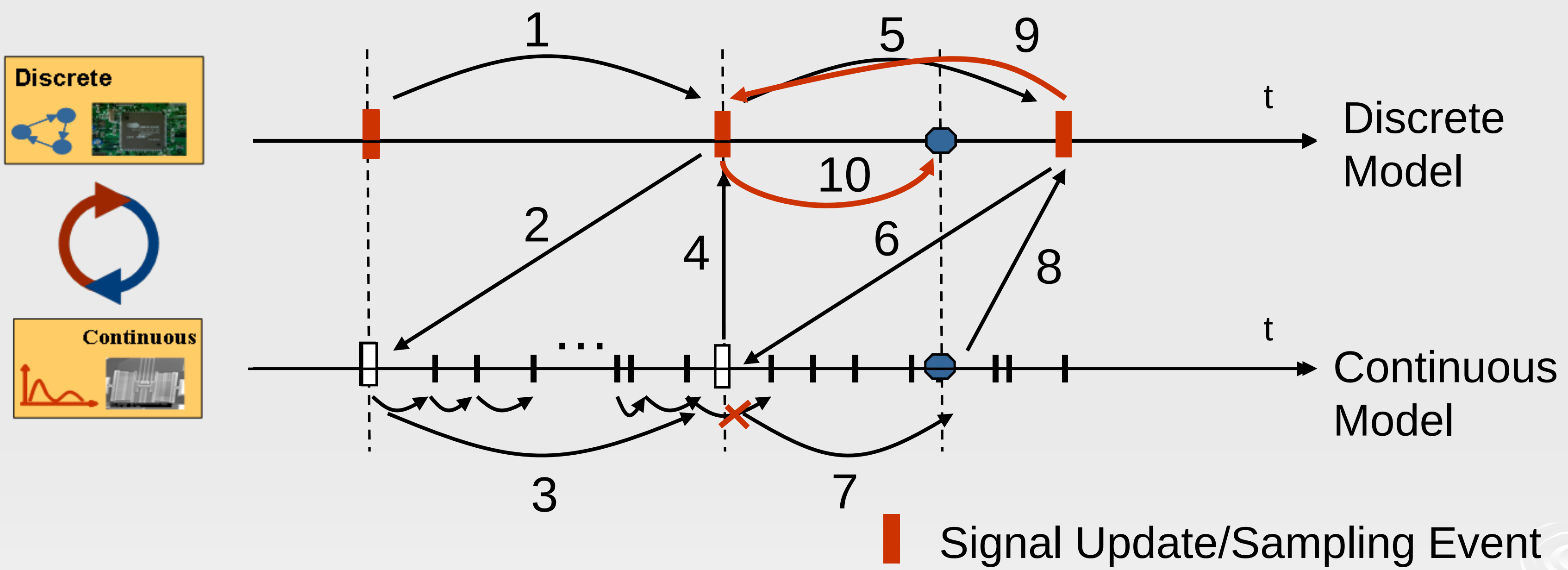
- Events exchanged between the continuous and the discrete models
  - State events
  - Sampling events
  - Update signal events



Continuous/Discrete Systems
- Global Simulation -

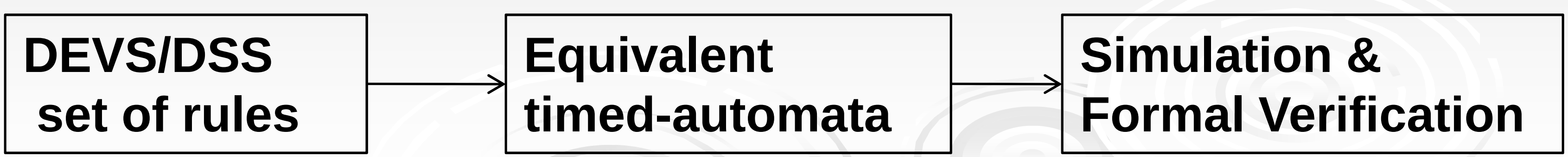


Rollback-Based Continuous/Discrete Synchronization Model

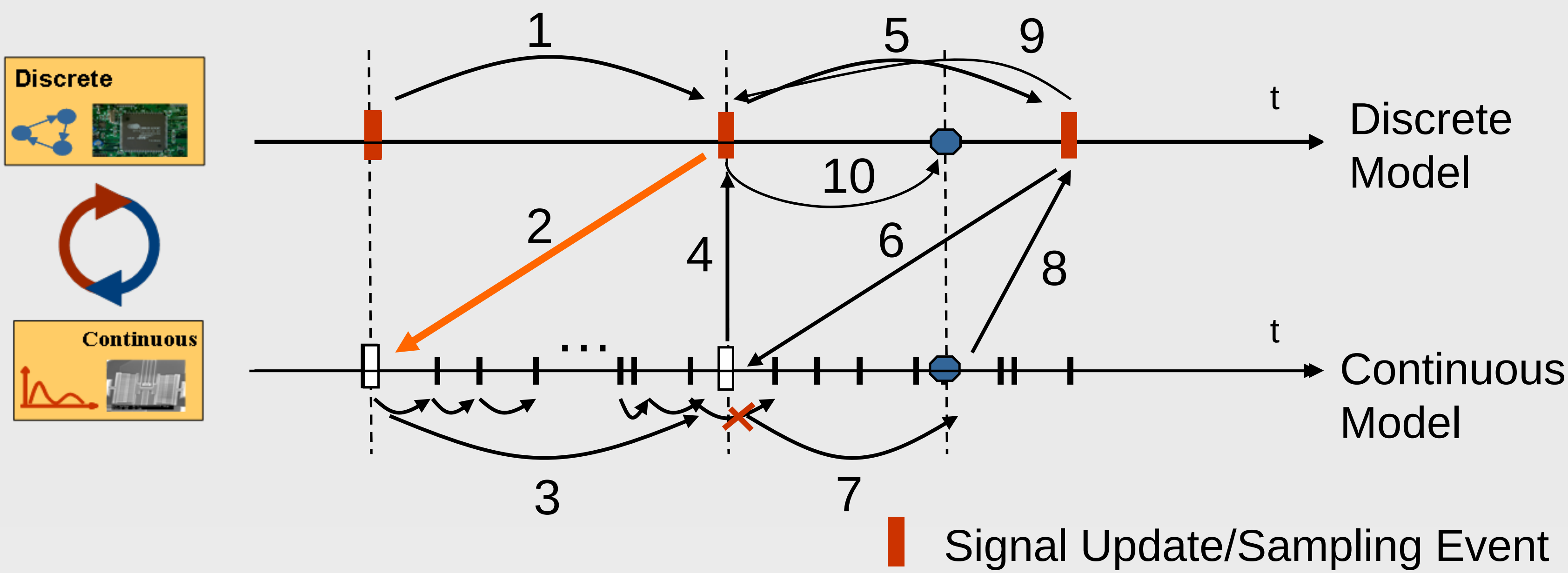


Synchronization Operational Semantics

- DEVS (Discrete Event Systems specification) and DESS (Differential Equation Systems Specification) formalisms
  - Set of rules $\dfrac{Premise}{Conclusion}$ 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)

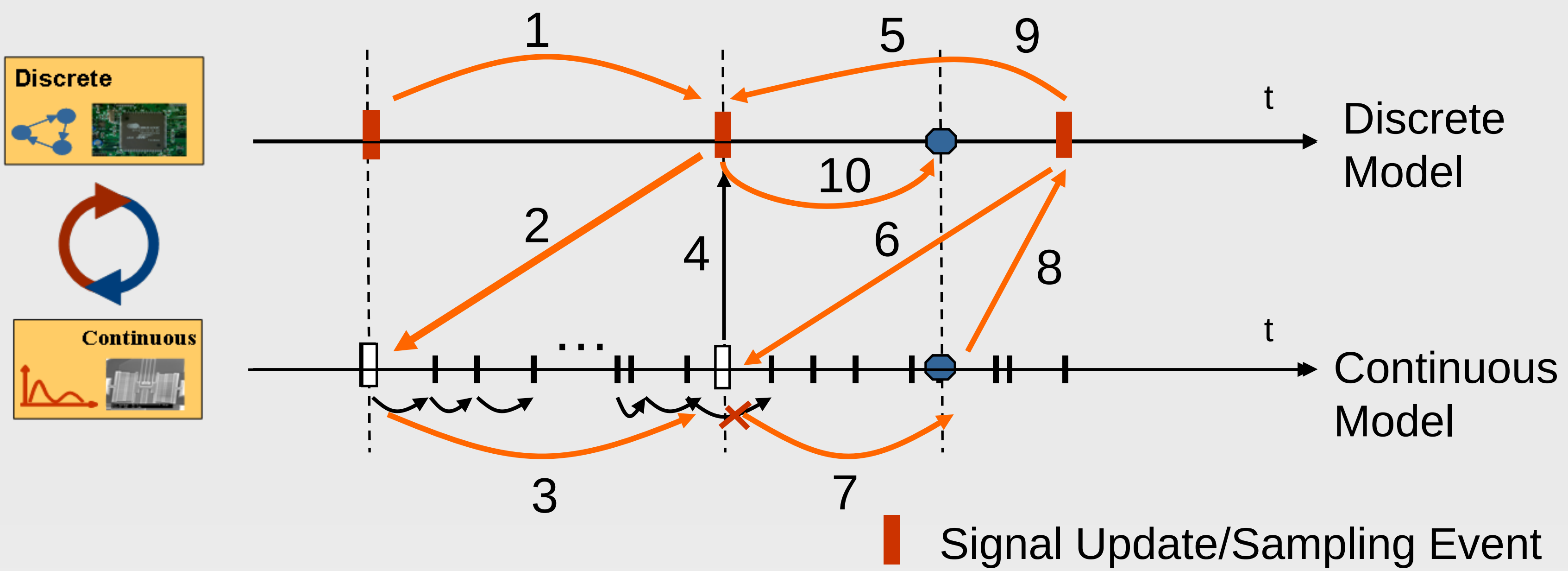


Synchronization Operational Semantics using DEVS



$$\frac{synch = 1 \wedge flag = 0 \wedge q = \delta_{ext}(q)}{q \xrightarrow{(DataFromBus,\, t_a(s_{dk}))?;\, synch:=0} q}$$

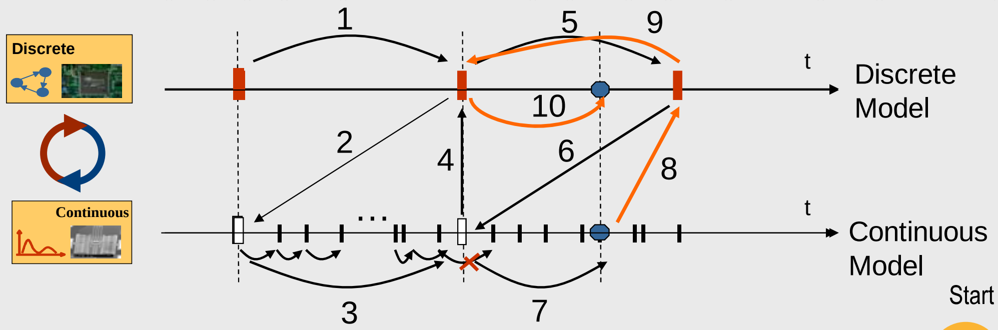
Synchronization Operational Semantics using DEVS

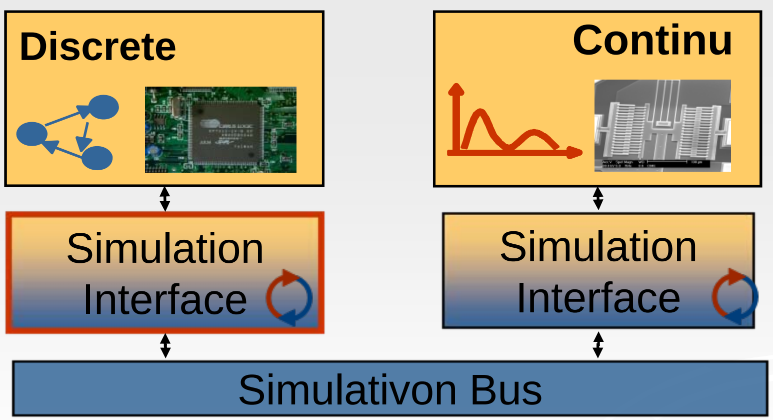


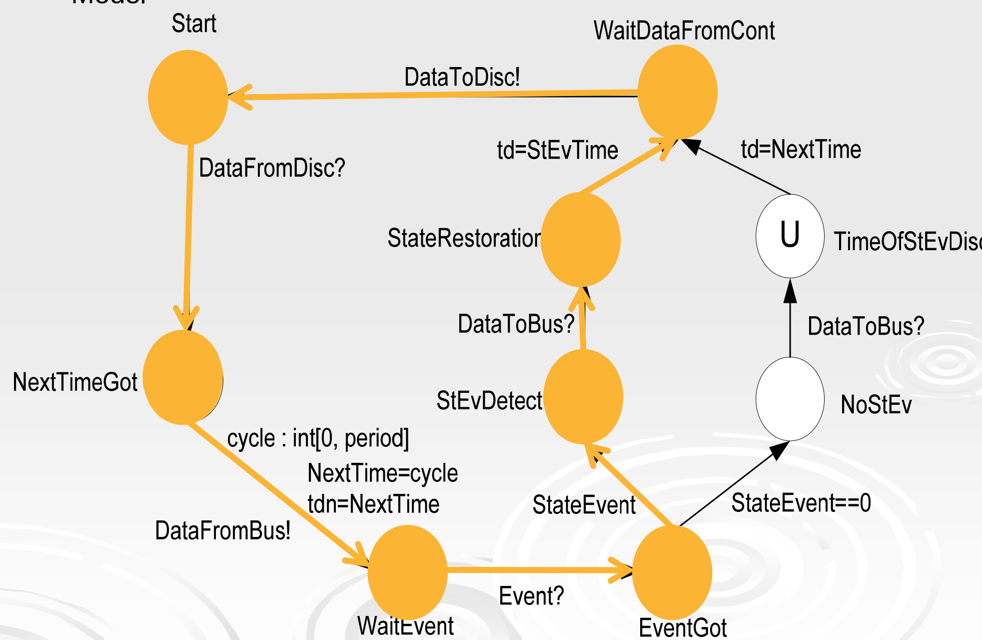
- Set of rules for the overall synchronization model

Verification of Simulation Interfaces

- Based on timed-automata – use of UPPAAL tool

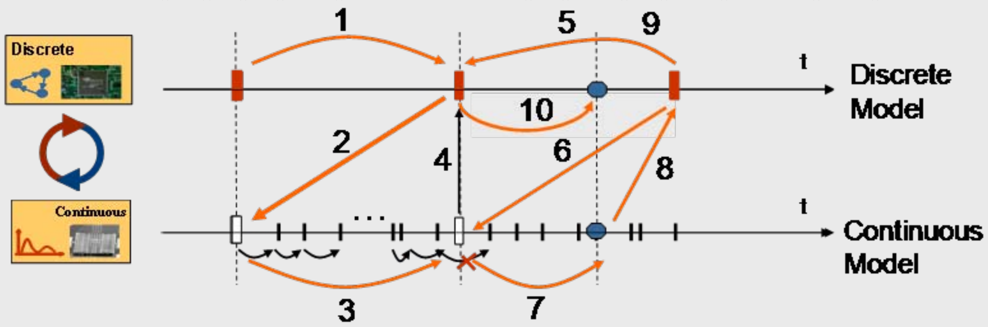


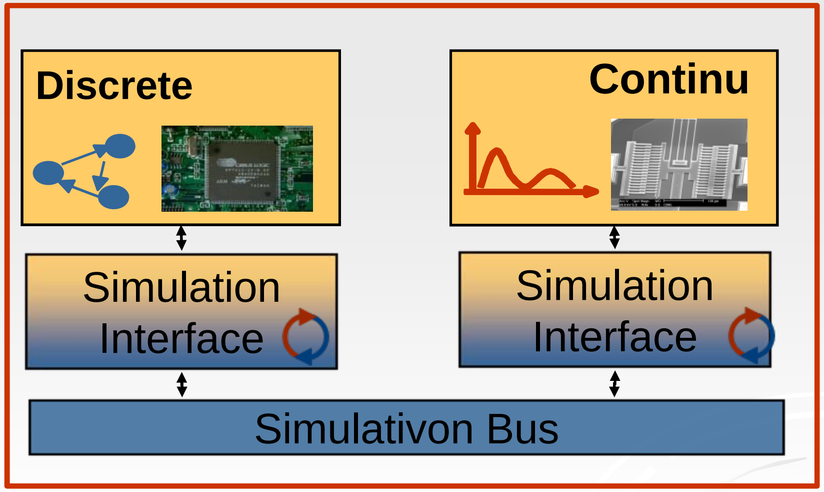




Verification of Simulation Interfaces

- Based on timed-automata – use of UPPAAL tool



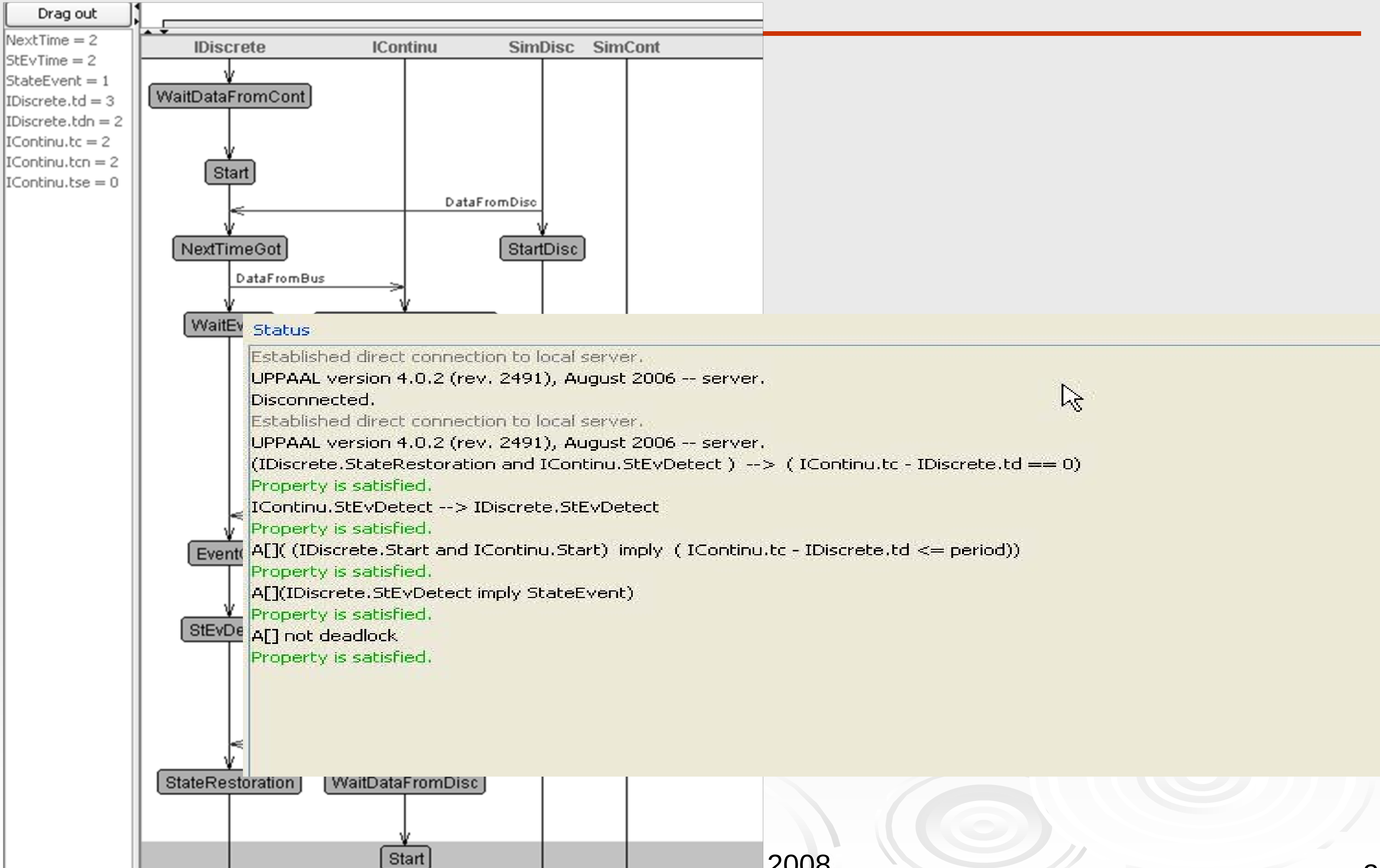


- Global Verification requires Timed Automata for
  - Discrete Simulator
  - Continuous Simulator
  - The Continuous Simulator Interface
  - The Discrete Simulator Interface

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



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