Colored Petri Nets for Modeling and Analyzing Biological Systems

Fei Liu

Department of Computer Science
Brandenburg University of Technology Cottbus

Dagstuhl Seminar
April 10 2011
Outline

- Motivation
- Snoopy
Motivation
Producer-Consumer
Producer-Consumer
Producer-Consumer

Colored Petri Nets for Modeling and Analyzing Biological Systems

April 10, 2011
Why use colored Petri nets
Why use colored Petri nets

- Compact and readable representation,
Why use colored Petri nets

- Compact and readable representation,
- Scalable models,
- Increasing net size = increasing color sets,
Why use colored Petri nets

- Compact and readable representation,
- Scalable models,
- Increasing net size = increasing color sets,
- Analysis techniques of low-level Petri nets by automatic unfolding,
  - Animation/Simulation,
  - Structural analysis,
  - State space analysis.
Why use colored Petri nets

- Compact and readable representation,
- Scalable models,
- Increasing net size = increasing color sets,
- Analysis techniques of low-level Petri nets by automatic unfolding,
  - Animation/Simulation,
  - Structural analysis,
  - State space analysis.
- Analysis techniques of high-level Petri nets.
  - Symbolic simulation,
  - State space analysis.
Why use colored Petri nets

- Compact and readable representation,
- Scalable models,
- Increasing net size = increasing color sets,
- Analysis techniques of low-level Petri nets by automatic unfolding,
  - Animation/Simulation,
  - Structural analysis,
  - State space analysis.
- Analysis techniques of high-level Petri nets.
  - Symbolic simulation,
  - State space analysis.
Scenarios
Scenarios

- biological systems with similar objects
  - cells
  - genes
  - receptors
  - transducers
Scenarios

- biological systems with similar objects
  - cells
  - genes
  - receptors
  - transducers

- biological systems with spatial aspects
  - grid-based agent systems
  - membrane systems
Example: C. elegans

- Colset CS = integer with 3-8;
Example: agent-based models with a grid

Example: membrane systems

1  {a}

\[ r_{11} : b \rightarrow a \]
\[ r_{12} : a \rightarrow (a, in_2), b \]

2

\[ r_{21} : a \rightarrow (a, in_3), b \]
\[ r_{22} : b \rightarrow \lambda \]

3  {b}

\[ r_{31} : a, b \rightarrow b, (b, out) \]
Snoopy
Colored Petri nets-based framework

SPN\textsuperscript{C}  
\begin{align*}
\text{Discrete State Space}
\end{align*}

QPN\textsuperscript{C}  
\begin{align*}
\text{Continuous State Space}
\end{align*}

CPN\textsuperscript{C}

\begin{align*}
\text{Approximation}
\end{align*}

Timed, Quantitative

Time-free

Abstraction

Abstraction
Features for modeling

- Drawing of the Petri net graph as usual.
- Rich data types for color set definition: dot, int, string, bool, enum, index, product, union.
- User-defined functions.
Features for modeling

- Drawing of the Petri net graph as usual.
- Rich data types for color set definition: dot, int, string, bool, enum, index, product, union.
- User-defined functions.
- Several extended arc types, such as inhibitor arc, read arc, equal arc, reset arc, and modifier arc.
- Several special transitions: stochastic transitions with freestyle rate functions, immediate transitions, deterministic transitions, and scheduled transitions.
Features for modeling

- Drawing of the Petri net graph as usual.
- Rich data types for color set definition: dot, int, string, bool, enum, index, product, union.
- User-defined functions.
- Several extended arc types, such as inhibitor arc, read arc, equal arc, reset arc, and modifier arc.
- Several special transitions: stochastic transitions with freestyle rate functions, immediate transitions, deterministic transitions, and scheduled transitions.
- Concise specification of initial marking for larger color sets.
- Rate function definition for each transition instance.
- Highlighting the markings, color sets, guards, and expressions.
Features for animation (for $QPN^C/SPN^C$)

- Automatic animation,
- Single-step animation by manually choosing a binding.
Features for simulation (for $SPN^C/CPN^C$)

- Simulation is done on an automatically unfolded Petri net.
- Show or export simulation results for colored or uncolored places/transitions separately or together.
Features for simulation (for $SPN^C/CPN^C$)

- Simulation is done on an automatically unfolded Petri net.
- Show or export simulation results for colored or uncolored places/transitions separately or together.
- Several simulation algorithms to simulate $SPN^C$, including the Gielespie stochastic simulation algorithm (SSA) [Gil77].
- Several simulation algorithms to simulate $CPN^C$, including the Euler algorithm, Runge-Kutta algorithm etc.

Features for export

- $QPN^C$, $SPN^C$ and $CPN^C$ are exported to different net formalisms within Snoopy,

- Export/import beyond Snoopy, e.g., export to CPN tools [CPN].

Ref: [CPN] http://cpntools.org
Thank You for Your Attention!