Petri nets
A brief presentation

Fabrice.Kordon@lip6.fr

mardi 9 juin 2009
Elements in a Petri net

Petri nets = bipartite graph

A state transition model

- Resources ➡ Places
- Evolution ➡ Transitions
- Evolution ➡ Arcs + tokens (firing rule)
Elements in a Petri net

Petri nets = bipartite graph

A state transition model

- Resources ➔ Places
- Evolution ➔ Transitions
- Evolution ➔ Arcs + tokens (firing rule)
Elements in a Petri net

Petri nets = bipartite graph

A state transition model

Resources  ➔  Places
Evolution  ➔  Transitions
Evolution  ➔  Arcs + tokens (firing rule)
Elements in a Petri net

Petri nets = bipartite graph

A state transition model

- Resources ➔ Places
- Evolution ➔ Transitions
- Evolution ➔ Arcs + tokens (firing rule)
The firing rule

Defines the behavior of the system
The firing rule

Defines the behavior of the system
The firing rule

Defines the behavior of the system
How to define the basics of distributed execution

Sequence

P1 → T1 → P2 → T2 → P3

Parallelism

P1 → T1 → P2 → T2 → P3

P1' → T1' → P2' → T2' → P3'

Synchronous communication

P1a → T1a → Sync → P2a → P3a

P1b → T1b → P2b

Asynchronous communication

P1 → T1 → P2 → T2 → P3

P1' → T1' → P2' → T2' → P3'

Buff
First example: two people waking up (1)
First example: two people waking up (1)
First example:

Two people waking up (1)
First example:
Two people waking up (2)
First example: two people waking up (3)
First example: two people waking up (4)
First example: two people waking up (5)
First example:
Two people waking up (6)

- Sleep
- Awake
- Go to eat
- In the bathroom
- Out of the bathroom
- Ready
- Gone

- Clock ringing

- Noise

- Eating
- Washed

- Bathroom
- Leave

- Mardi 9 juin 2009
First example:
Two people waking up (6)
First example: two people waking up (7)
First example:
Two people waking up (8)
First example: two people waking up (9)
First example: two people waking up (10)

- sleep1 → awake1
- aclock → ringing
- noise1
- noise2
- sleep2 → awake2
- goeat1 → eating1 → inbath1 → outbath1 → ready1
- gone1
- gone2
- washing2 → eating2 → inbath2 → outbath2 → ready2
- washer
The state space for this model

Expresses all possible behavior in the system
- 26 states
- 38 arcs

One state
- Integer vector representing marking of places

Expresses indeterminism of a parallel execution
- Interleaving of actions
The state space for this model

Expresses all possible behavior in the system
- 26 states
- 38 arcs

One state
Integer vector representing marking of places

Expresses indeterminism of a parallel execution
Interleaving of actions
It is important to relate the network with its reachability graph. Representation of a state as a vector of place marking.
It is important to relate the network with its reachability graph.

Representation of a state as a vector of place marking:

\[ <2, 1, 0, 0, 1> \]
Building the state space (also called reachability graph)

It is important to relate the network with its reachability graph.

Representation of a state as a vector of place marking

\[ \langle 2, 1, 0, 0, 1 \rangle \]
It is important to relate the network with its reachability graph.

Representation of a state as a vector of place marking:

\[
\begin{bmatrix}
  p1 \\ P2 \\ P3 \\ P4 \\ P5
\end{bmatrix}
\]
It is important to relate the network with its reachability graph.

Representation of a state as a vector of place marking.
Modeling the management of parking lots
Identifying actors

Class
Car is 1..10;

Var
\( c, p \) in Cars;

Coloured Petri Nets

Parking lots

Ready
\(<\text{Car.all}>\)

Get in \([c=p]\) in parking

Get out

Out
\(<\text{Car.all}>\)

\(<c, p>\)

\(<c>\)

\(<p>\)

\(<1>, <2>\)
Time management for the Parking model

Stochastic nets

Adding probability to transitions
Showing the principle

Fabrice.Kordon@lip6.fr
A small example

Guard_Out =>

BadData when IP1[BadData] or IP2[BadData],
NoData when (IP1[NoData] and IP2[BadData]) or (IP1[BadData] and IP2[NoData])
applies to OP;
BadData, NoData

Guard_Out =>

\[
\text{BadData when } \text{IP1}[\text{BadData}] \text{ or } \text{IP2}[\text{BadData}],
\]

\[
\text{NoData when } (\text{IP1}[\text{NoData}] \text{ and } \text{IP2}[\text{BadData}]) \text{ or }
\]

\[
(\text{IP1}[\text{BadData}] \text{ and } \text{IP2}[\text{NoData}])
\]

applies to OP;

Class Error is [NoData, BadData,...];
Var x, y, z in Error;
A small example

Guard_Out =>

BadData when IP1[BadData] or IP2[BadData],
NoData when (IP1[NoData] and IP2[BadData]) or
    (IP1[BadData] and IP2[NoData])

applies to OP;

Class Error is [NoData, BadData,...];
Var x, y, z in Error;
A SMALL EXAMPLE

BadData, NoData

Guard_Out =>

BadData when IP1[BadData] or IP2[BadData],
NoData when (IP1[NoData] and IP2[BadData]) or
(IP1[BadData] and IP2[NoData])

applies to OP;

Class Error is [NoData, BadData,...];
Var x, y, z in Error;

mardi 9 juin 2009
A SMALL EXAMPLE

Guard_Out =>

BadData when IP1[BadData] or IP2[BadData],
NoData when (IP1[NoData] and IP2[BadData]) or
(IP1[BadData] and IP2[NoData]) applies to OP;

Class Error is [NoData, BadData,...];
Var x, y, z in Error;

mardi 9 juin 2009