PeabraiN: a PIPE extension for Performance Estimation and Resource Optimisation

Ricardo J. Rodríguez, Jorge Júlvez, José Merseguer

{rjrodriguez, julvez, jmerse}@unizar.es

Universidad Zaragoza 1542

June 28th, 2012

12th International Conference on Application of Concurrency to System Design

Hamburg, Deutschland

Outline

Problems we are dealing with (I)

Informal way – for human beings

Main goals

- Performance Estimation
- Resource Optimisation

Universidad Zaragoza 1542

Problems we are dealing with (I)

Informal way – for human beings

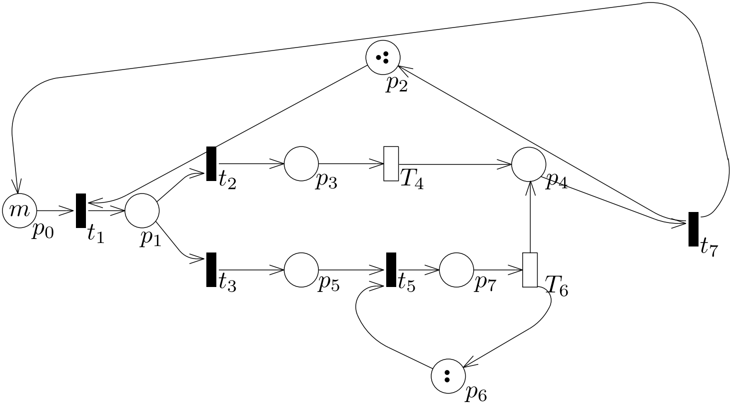
Main goals

- Performance Estimation
- Resource Optimisation

- PN class: Process Petri nets
- Shared resources systems
- Any process which involves resource usage to complete
- Different jobs with dissimilar handling
- Examples:
  - Assembly lines
  - Service-Oriented-Architecture services
  - Supermarket shopping
  - ...

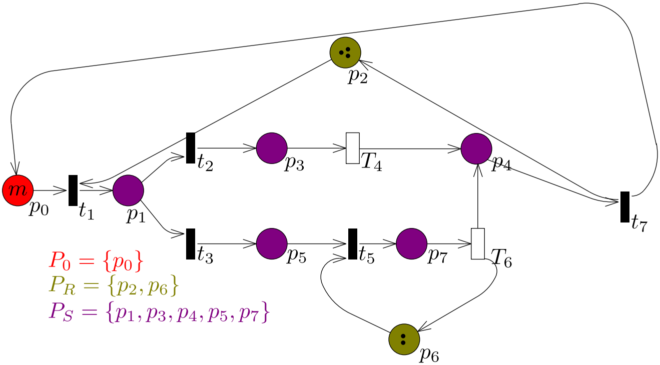
Problems we are dealing with (II)

Informal way – some examples



Problems we are dealing with (II)

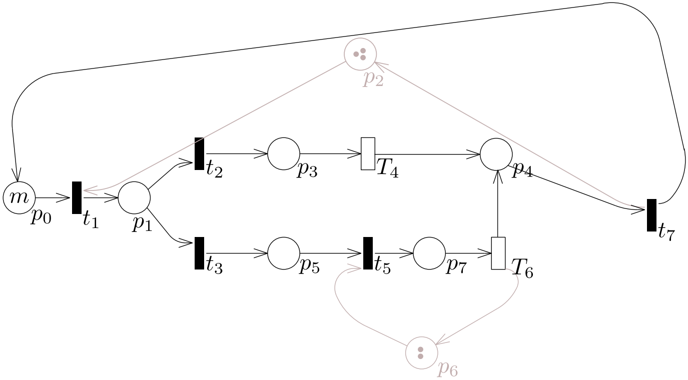
Informal way – some examples



- Places can be divided in three subsets: $P = P_0 \cup P_S \cup P_R$
  - *Process-idle place*, $P_0 = \{p_0\}$
  - *Process-activity places*, $P_S \neq \emptyset, P_S \cap P_0 = \emptyset, P_S \cap P_R = \emptyset$
  - *Resources places*, $P_R = \{r_1, \ldots, r_n\},\ n > 0, P_R \cap P_0 = \emptyset$

Problems we are dealing with (II)

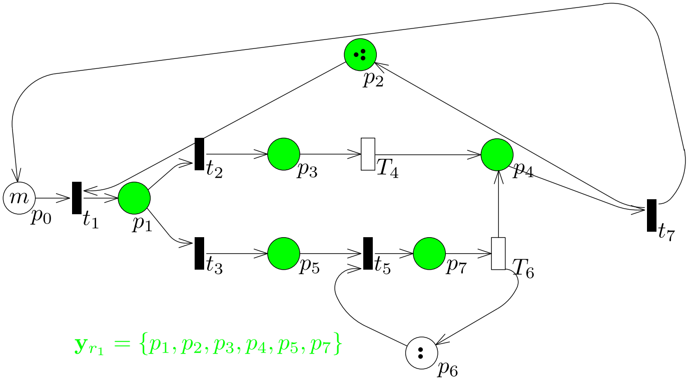
Informal way – some examples



- When removing P_R places, we get a strongly connected state machine, s.t. every cycle contains p_0

Problems we are dealing with (II)

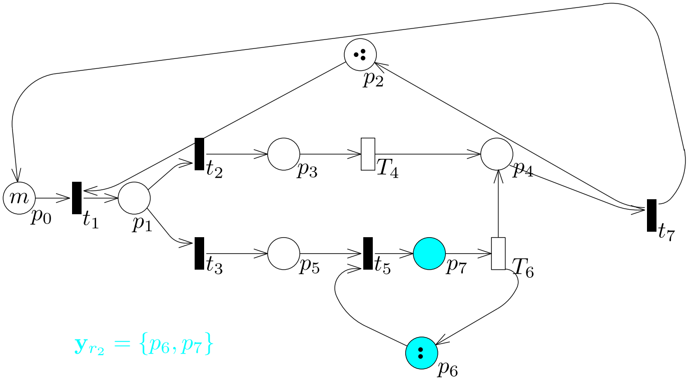
Informal way – some examples



- For each $r \in P_R$, there exists a unique minimal p-semiflow associated to $r, \mathbf{y}_r \in \mathbb{N}^{|P|}$ s.t. it contains on its support just the resource r and does not contains p_0

Problems we are dealing with (II)

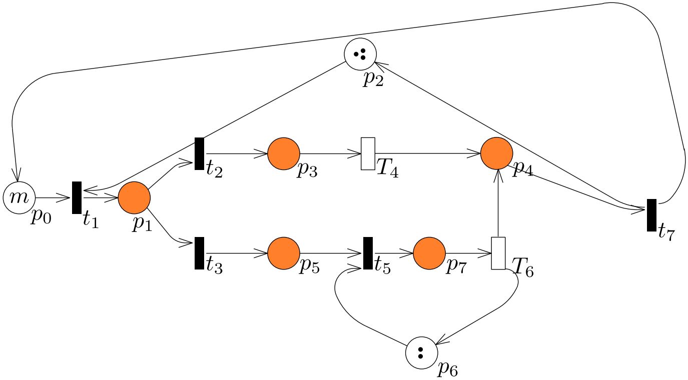
Informal way – some examples



- For each $r \in P_R$, there exists a unique minimal p-semiflow associated to $r, \mathbf{y}_r \in \mathbb{N}^{|P|}$ s.t. it contains on its support just the resource r and does not contains p_0

Problems we are dealing with (II)

Informal way – some examples



- Activity places set P_S does not contain resource places, neither process-idle place

Outline

Universidad Zaragoza 1542

The PeabraiN tool (I): what is it? (1)

Our (primary) interest

- Performance estimation based on Linear Programming (LP) techniques
  - Upper throughput bound computation
- Resource optimisation
  - Might the assembly line get more items completed per unit of time?
  - If so, how many staff it is needed to achieve it?

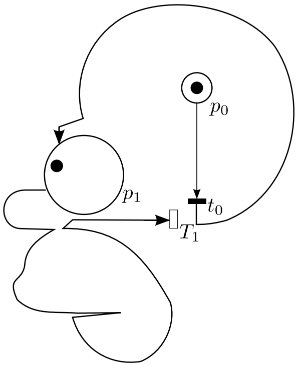
Alternatives studied

- GreatSPN (platform-dependent)
- MATLAB (proprietary ¬¬)
- HISim (PIPE extension for Hybrid PNs)

Zaragoza
1542

The PeabraiN tool (I): what is? (2)

Performance Estimation bAsed (on) Bounds (and) Resource optimisAtIon (for Petri) Nets

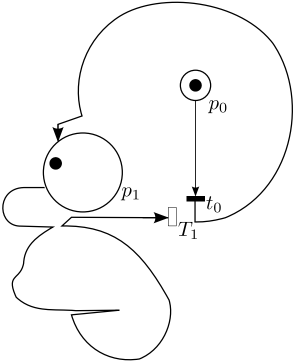


- Based on PIPE tool
  - User-friendly Java interface

1542

The PeabraiN tool (I): what is? (2)

Performance Estimation bAsed (on) Bounds (and) Resource optimisAtIon (for Petri) Nets



- Based on PIPE tool
  - User-friendly Java interface
  - Non-CPU-friendly Java interface :)
- PNML-compliant tool (+1)
  - PNML is ISO/IEC 15909 standard
- Easiness of extension: modules (+1)
- GPL license
  - Some flaws discovered → -1
- Multiplatform (Java-based)

1542

The PeabraiN tool (II): behind the scenes (1a)

Performance Estimation

- LP problem (LPP) for computing upper throughput bounds

$$\begin{aligned} \mathbf{\Gamma}(t_i) = \;\; & \textit{maximum}\; \mathbf{y} \cdot \mathbf{Pre} \cdot \mathbf{D} \\ & \textit{subject to}\; \mathbf{y} \cdot \mathbf{C} = \mathbf{0} \\ & \mathbf{y} \cdot \mathbf{m_0} = 1 \\ & \mathbf{y} \geq \mathbf{0} \end{aligned} \tag{1}$$

  Lower bound for the *average inter-firing time* of transition t_i, $\mathbf{\Gamma}(t_i)$

  Upper bound $\Theta(t_i)$ for the steady-state throughput $\Theta(t_i) = \dfrac{1}{\mathbf{\Gamma}(t_i)}$

- Based on previous work
  - Regrowing technique
- More accurate than previous methods
- Further reading: google "RJ-EPEW-10"

The PeabraiN tool (II): behind the scenes (1b)

Performance Estimation

Input

- Degree of improvement to achieve

Output

- New upper throughput bound

Steps

1. Compute initial upper throughput bound and initial bottleneck cycle (LPP in previous slide!)
2. Iterate until no significant improvement is achieved
   1. Look for next slowest p-semiflow (constraining the current one)
   2. Add the next slowest p-semiflow to the current one
   3. Compute new throughput bound

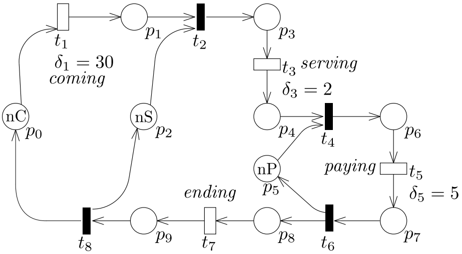
Zaragoza
1542

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



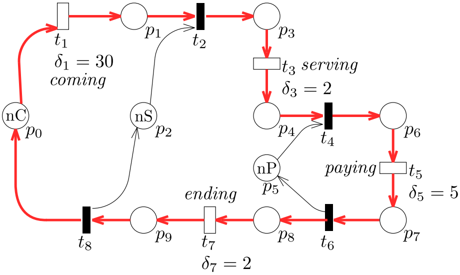
Zaragoza
1542

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$

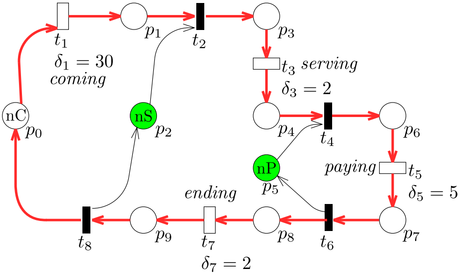
Zaragoza

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$
- Next candidates: $\{p_2, p_5\}$

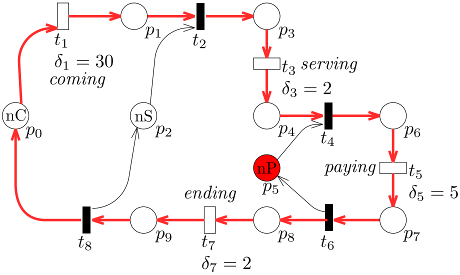
Zaragoza
1542

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



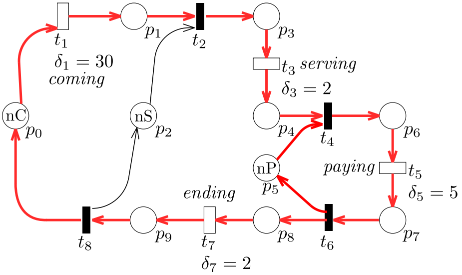
- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$
- Next candidates: $\{p_2, p_5\} \rightarrow p_5$

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



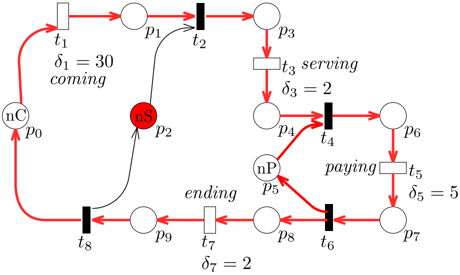
- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$
- Next candidates: $\{p_2, p_5\} \rightarrow p_5$ $\Theta' = 0.339410$

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$
- Next candidates: $\{p_2, p_5\} \rightarrow p_5$ $\Theta' = 0.339410$
- Next candidate: $\{p_2\}$

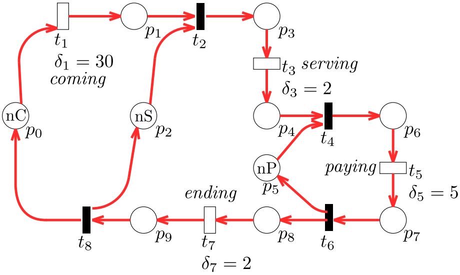
Zaragoza
1542

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$
- Next candidates: $\{p_2, p_5\} \rightarrow p_5$ $\Theta' = 0.339410$
- Next candidate: $\{p_2\}$ $\Theta'' = 0.328190$

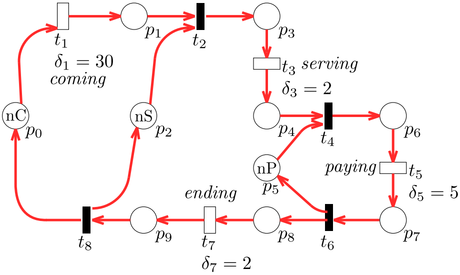
Zaragoza
1542

The PeabraiN tool (II): behind the scenes (1c)

Performance Estimation (example)

Parameters

- $nC = 15$; $nS = 4$; $nP = 2$



- Slowest p-semiflow: $\{p_0, p_1, p_3, p_4, p_6, p_7, p_8, p_9\}$, $\Theta = 0.384612$
- Next candidates: $\{p_2, p_5\} \rightarrow p_5$ $\Theta' = 0.339410$
- Next candidate: $\{p_2\}$ $\Theta'' = 0.328190$
- Improvement of 14.67%

Zaragoza
1542

The PeabraiN tool (II): behind the scenes (1d)

Performance Estimation (example)

Performance Estimation Module in PeabraiN

Performance Estimation Computation

Epsilon: 0

Compute

Regrowing step	Components	Throughput	Error accuracy	Sim. time
0	{P0, P1, P3, P4, P6, P7, P8, P9}	0.384612	±0.0000	0.0010
1	{P0, P1, P3, P4, P5, P6, P7, P8, P9}	0.339410	±0.0008	13.2000
2	{P0, P1, P2, P3, P4, P5, P6, P7, P8, P9}	0.328190	±0.0006	13.4140

Total elapsed time: 26.6140 seconds.

Universidad Zaragoza 1542

The PeabraiN tool (II): behind the scenes (2a)

Resource Optimisation

Inputs

- Resource places
- Process-idle place
- Available budget
- Cost per resource (vector)

Outputs

- Throughput maximised
- Vector (resource, number_of_instances)
- When reached process idle-place $\longrightarrow$ over-dimensioned system (idle resources)

Steps

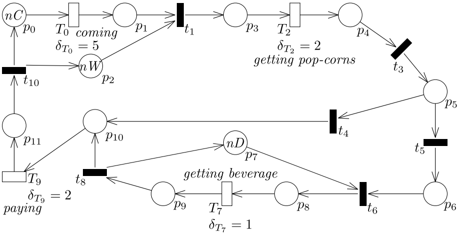
1. LPP for slowest p-semiflow (previous slide!)
2. Iterate until no more money available, or reached process idle-place
   1. Compute next p-semiflow constraining current one
   2. Compute the cost of incrementing such a resource
3. Have all resources been incremented? Still remaining money?
   1. If so, we can keep incrementing $\sim$ Bounded Knapsack Problem (NP-complete)

The PeabraiN tool (II): behind the scenes (2b)

Resource Optimisation (example)

Parameters

- $nC = 10$, 50% take beverage; $nW = 2$; $nD = 2$;
- Initial budget \$80,000
- New hiring \$6,000; new beverage dispenser \$250

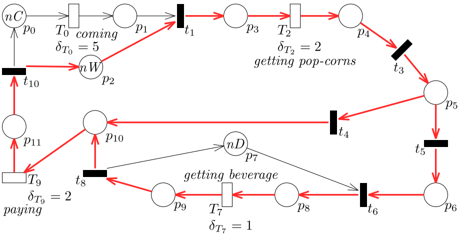


The PeabraiN tool (II): behind the scenes (2b)

Resource Optimisation (example)

Parameters

- $nC = 10$, 50% take beverage; $nW = 2$; $nD = 2$;
- Initial budget \$80,000
- New hiring \$6,000; new beverage dispenser \$250



- Slowest p-semiflow: $\{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$

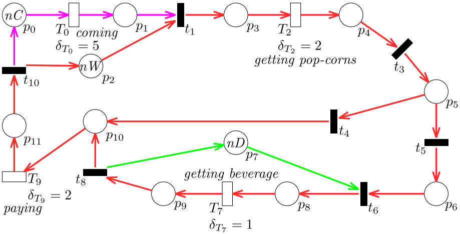
The PeabraiN tool (II): behind the scenes (2b)

Resource Optimisation (example)

Parameters

- $nC = 10$, 50% take beverage; $nW = 2$; $nD = 2$;

- Initial budget \$80,000
- New hiring \$6,000; new beverage dispenser \$250



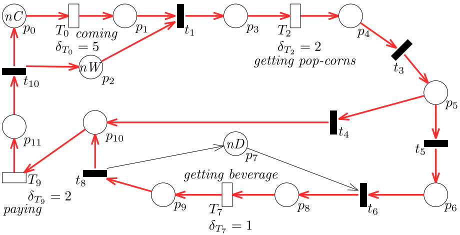
- Slowest p-semiflow: $\{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$
- Candidates: $\{p_0, p_1\}$ or $\{p_7\}$

The PeabraiN tool (II): behind the scenes (2b)

Resource Optimisation (example)

Parameters

- $nC = 10$, 50% take beverage; $nW = 2$; $nD = 2$;
- Initial budget \$80,000
- New hiring \$6,000; new beverage dispenser \$250



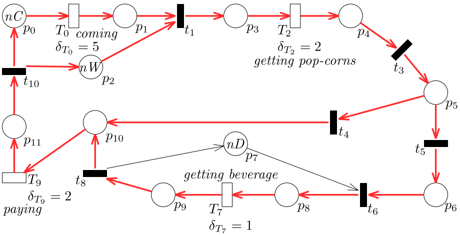
- Slowest p-semiflow: $\{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$
- Candidates: $\{p_0, p_1\}$ or $\{p_7\}$ $\rightarrow \{p_0, p_1\}$

The PeabraiN tool (II): behind the scenes (2b)

Resource Optimisation (example)

Parameters

- $nC = 10$, 50% take beverage; $nW = 2$; $nD = 2$;
- Initial budget \$80,000
- New hiring \$6,000; new beverage dispenser \$250



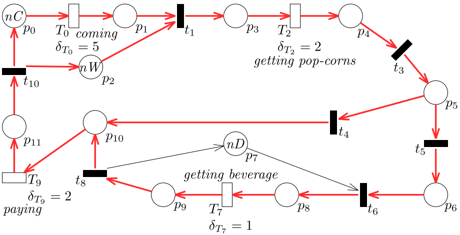
- Slowest p-semiflow: $\{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$
- Candidates: $\{p_0, p_1\}$ or $\{p_7\}$ $\rightarrow \{p_0, p_1\}$
- Contains $p_0 \rightarrow$ no more increments can be carried out!

The PeabraiN tool (II): behind the scenes (2b)

Resource Optimisation (example)

Parameters

- $nC = 10$, 50% take beverage; $nW = 2$; $nD = 2$;
- Initial budget \$80,000
- New hiring \$6,000; new beverage dispenser \$250

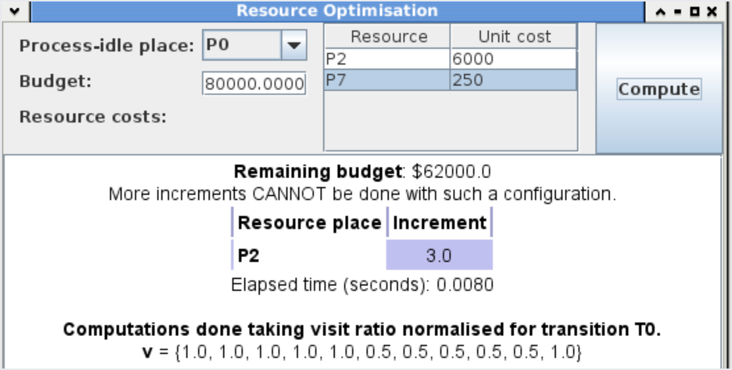


- Slowest p-semiflow: $\{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}\}$
- Candidates: $\{p_0, p_1\}$ or $\{p_7\}$ $\rightarrow \{p_0, p_1\}$
- Contains $p_0 \rightarrow$ no more increments can be carried out!
- p_2 must be incremented in 3 units

The PeabraiN tool (II): behind the scenes (2c)

Resource Optimisation (example)

Resource Optimisation Module in PeabraiN



The PeabraiN tool (III): Other Features

Some Easter eggs

Extra Functionalities added

- Lower and Upper throughput bound
- Slowest P-semiflow
- Structural marking (enabling) bound of a place (transition)

Universidad Zaragoza 1542

The PeabraiN tool (III): Other Features

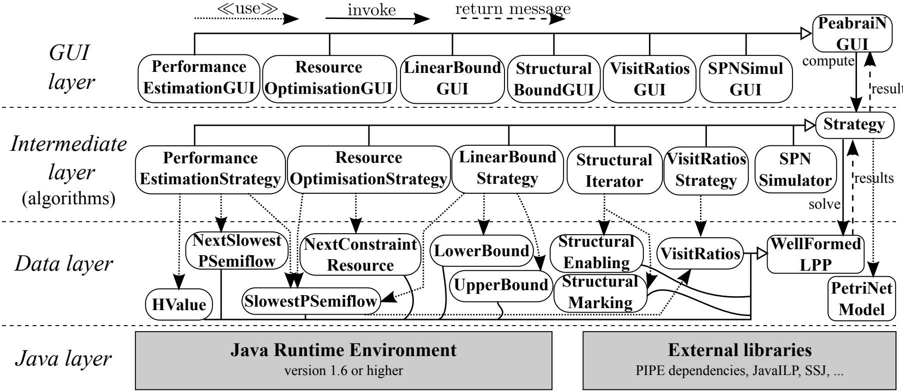
Some Easter eggs

Extra Functionalities added

- Lower and Upper throughput bound
- Slowest P-semiflow
- Structural marking (enabling) bound of a place (transition)
- Visit ratios computation
- SPN simulator
  - PIPE built-in simulator DOES NOT handle infinite-server semantics
    - Simulates just a no. of firings
  - Can be invoked stand-alone
  - Supports command-line invokation
  - Uses an exact stochastic simulation algorithm

Universidad Zaragoza 1542

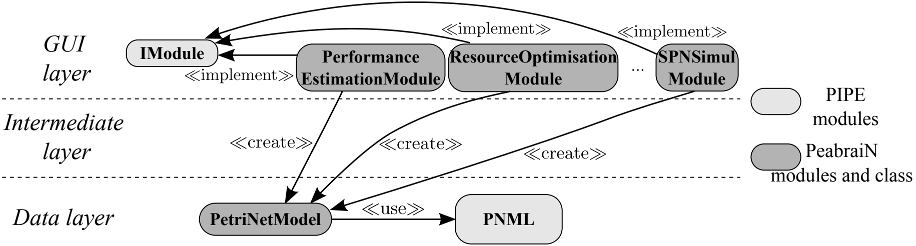
The PeabraiN tool (IV): Architecture (1)



PeabraiN software architect

- Model-View-Controller (MVC) architectural pattern

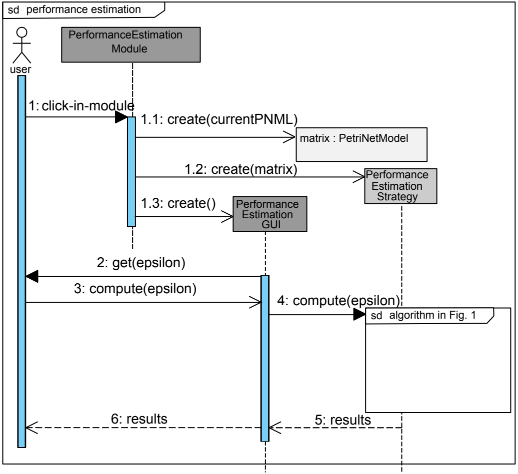
The PeabraiN tool (IV): Architecture (2)

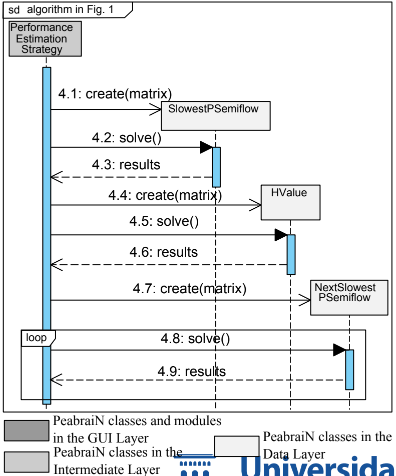


Integration of PeabraiN in PIPE

- Modules
- PN matrix definition (PetriNetModel)

The PeabraiN tool (IV): Architecture (3)





Universidad Zaragoza 1542

Outline

Universidad Zaragoza

Tool Installation and Availability

What do I need for executing it?

- JRE $\geq$ v1.6
- An LPP solver, namely GLPK and GLPK JNI library
- Rest of necessary files included in deployment!
- Successfully tested in
  - GNU/Linux
  - Windows
  - Not yet in MacOS $\rightarrow$ error on GLPK JNI library

Sounds nice, where can I get it?

- Project description
  - `http://webdiis.unizar.es/GISED/?q=tool/peabrain`
- Source code + binaries
  - `http://peabrain.sourceforge.net`

Outline

Future Work

What do we expect in the future?

- Automatic detection of installed LPP solvers
- LPP solver chosen by user
- Parametric evaluation (as GreatSPN does)
- Approximate stochastic simulation algorithm
- Simulation with DSPN

Future Work

What do we expect in the future?

- Automatic detection of installed LPP solvers
- LPP solver chosen by user
- Parametric evaluation (as GreatSPN does)
- Approximate stochastic simulation algorithm
- Simulation with DSPN
- (put here your idea, we'll [try to. . .] do it!)

Universidad Zaragoza 1542

Future Work

What do we expect in the future?

- Automatic detection of installed LPP solvers
- LPP solver chosen by user
- Parametric evaluation (as GreatSPN does)
- Approximate stochastic simulation algorithm
- Simulation with DSPN
- (put here your idea, we'll [try to. . .] do it!)

[Some of these currently under development by a Final Degree Project]

Universidad Zaragoza 1542

Wait! Not yet finished :D

It's demo time! (in a while)

Just reach me on Tool Exhibition slot!

PeabraiN: a PIPE extension for Performance Estimation and Resource Optimisation

Ricardo J. Rodríguez, Jorge Júlvez, José Merseguer

{rjrodriguez, julvez, jmerse}@unizar.es

Universidad Zaragoza

June 28^{th}, 2012

12^{th} International Conference on Application of Concurrency to System Design

Hamburg, Deutschland