Modelling and Analysis of Non-Functional Properties in Critical Systems with Petri Nets

Elena Gómez-Martínez
Ricardo J. Rodríguez
{elena.gomez, rjrodriguez}@fi.upm.es

Babel Group
Universidad Politécnica de Madrid
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
Basic concepts (I)

• A graphical tool for the formal description of the flow activities in complex systems.


• Useful for modelling concurrent, distributed, asynchronous behaviour in a discrete system.
Basic concepts (II)

- A Petri net (PN) is a four-tuple:
  \[ N = (P, T, Pre, Post) \]
  - \( P = \{p_1, p_2, \ldots, p_n\} \) is a finite set of places.
  - \( T = \{t_1, t_2, \ldots, t_m\} \) is a finite set of transitions.
  - \( Pre = |P|x |T| \) represents the arcs from places to transitions.
  - \( Post = |T|x |P| \) represents the arcs from transitions to places.
Basic concepts (III)

• $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$

• Weight = multiplicity of the arc
  \[ \text{Post}[p, t] = w \rightarrow \text{an arc from } t \text{ to } p \text{ with multiplicity } w. \]

• Ordinary nets
  \[ \text{Arcs have weight } 1 \]

• Incidence matrix:
  \[ C = Pre - Post \]
Basic concepts (IV)

\[ N = (P, T, Pre, Post) \]
Basic concepts (V)

• A Petri net model consists of:
  – A net structure
    \[ N = (P, T, Pre, Post) \]
  – A initial marking
    \[ M = \{m_1, m_2, ..., m_p\} \]
Basic concepts (VI)

• Firing rules
  – A transition is *enabled* in a given marking if all its input places carry at least one token.
  – An enabled transition fires by removing one token per arc from each input place and adding one token per arc to each output place.
Basic concepts (VII)

• Firing rules
  – A transition is *enabled* in a given marking if all its input places carry at least one token.
  – An enabled transition fires by removing one token per arc from each input place and adding one token per arc to each output place.
Basic concepts (VIII)

- A p-semiflow implies a token conservation law independent from any firing of transitions.
  - nonnegative-integer vector $\mathbf{y} \geq \mathbf{0}$,
  - $\mathbf{y} \cdot C = 0$

- A t-semiflow
  - nonnegative-integer vector $\mathbf{x} \geq \mathbf{0}$,
  - $C \cdot \mathbf{x} = 0$
Basic concepts (IX)

• Modelling expressivity
  – Sequences
Basic concepts (X)

• Modelling expressivity
  – Sequences
  – Conflicts:
    • Decisions
Basic concepts (XI)

• Modelling expressivity
  – Sequences
  – Conflicts:
    • Decisions
    • Iterations
Basic concepts (XII)

- Modelling expressivity
  - Sequences
  - Conflicts:
    - Decisions
    - Iterations
  - Concurrency and synchronizations
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
Extensions (I)

• Application context:
  – Timed
  – Coloured
  – Well formed
  – Continuous
  – Hybrid
  – …
Extensions (II): adding time

• PN models include no notion of time.
• Duration of the events:
  – Deterministic
  – Random
• Time associated to:
  – Places
  – Transitions
  – Tokens
Extensions (III): adding time

• Stochastic Petri nets (SPN):
  – Temporal interpretation.
  – Firing delay/time is associated with each transition.
    • Amount of time that must elapse before transition can fire.
    • A random variable with negative exponential probability density function.
  – Conflict resolution policy.
Extensions (IV): adding time

- Generalized Stochastic Petri nets (GSPN)
  - SPN extension.
  - Two kinds of transitions
    - Immediate:
      - Priority greater than zero.
      - It fires in zero time.
    - Temporized:
      - Zero priority.
      - Random variable: negative exponential probability distribution
Outline

• Basic concepts
• Extensions
• **Properties**
• Analysis
  – Performance analysis
• Conclusions
• References
Properties (I)

• Behavioural properties
  – Related to the dynamic (marking)

• Structural properties
  – Related to the static net
Properties (II)

• Behavioural properties:
  – **Boundedness**: finiteness of the state space, i.e. the marking of all places is bounded.
  – **Safeness** = 1-boundedness (binary marking)
  – **Mutual Exclusion**: two or more places cannot be marked simultaneously (problem of shared resources).
  – **Deadlock**: situation where there is no transition enabled.
  – **Liveness**: infinite potential activity of all transitions.
  – **Home state**: a marking that can be recovered from every reachable marking.
  – **Reversibility**: recovering of the initial marking.
Properties (II)

• Structural properties:
  – **Structurally bounded**: all marking is bounded.
  – **Structurally live**: there exists a marking for which the net is live.
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
Analysis (I)

- Techniques for the analysis of net systems:
  - **Enumeration** (Reachability/Coverability graphs)
  - **Transformation**: Reduction
  - **Structural**: bridge between behaviour and structure.
Analysis (II)

• Complementary classification
  – **Exact techniques**: construction of the isomorphic Continuous Time Markov Chain.
    • State explosion problem.
  – **Approximation techniques**: solution of smaller components.
  – **Bounds.**
Analysis (III)

• Performance Analysis
  – Response time/throughput
  – Scalability

• Fault Tolerance Analysis

• Dependability

• Safety Analysis
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
Performance Analysis of PNs (I)

• Throughput: jobs completed per unit of time

• Exact computation
  – We need to explore the reachability graph!
  – State explosion problem: computation of performance becomes unachievable

• Approximate computation: upper throughput bounds
  – Using Linear Programming (LP) techniques
  – Good accuracy – computational complexity trade-off
Performance Analysis of PNs (II)

• Petri net subclasses
  – State Machine: \( \forall t \in T, |t\bullet| = |\bullet t| = 1 \)
  – Marked Graph: \( \forall p \in P, |\bullet p| = |p\bullet| = 1 \)

• Process Petri nets (PPNs)
  – Any process which involves resource usage to complete
  – Different jobs with dissimilar handling
  – Examples: Assembly lines, Service-Oriented-Architecture systems, etc.

Performance Analysis of PNs (III)
Performance Analysis of PNs (IV)

- Places divided in 3 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**, $P_R = \{r_1, \ldots, r_n\}$, $n > 0$, $P_R \cap P_0 = \emptyset$
Performance Analysis of PNs (V)

- When removing $P_R$ places, we get a strongly connected state machine, s.t. every cycle contains $p_0$. 
Performance Analysis of PNs (VI)

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

$y_{r_2} = \{p_2, p_3, p_4, p_5, p_7\}$
• Activity places set \( P_s \) does not contain neither resource places, nor process-idle place
Performance Analysis of PNs (VIII)

• **Little’s law**: \( L = \lambda \cdot W \) (queue length, arrival rate, waiting time)

• Applying it to PNs: \( m \geq \text{Pre} \cdot D \cdot \Theta \)

\[
\begin{align*}
\text{Maximize } \Theta : \\
\overline{m} \geq \text{Pre} \cdot D \cdot \Theta \\
\overline{m} = m_0 + C \cdot \sigma \\
\sigma \geq 0
\end{align*}
\]

• \( y \) is the slowest \( p \)-semiflow of the system (bottleneck)

• **Our aim**: Find next constraining \( p \)-semiflow
Performance Analysis of PNs (IX)

Given a $y^*$ slowest p-semiflow, we can compute the next in a PPN as:

\[
\text{maximum } y \cdot \text{Pre} \cdot D
\]

subject to $y \cdot C = 0$

$y \cdot m_0 = 1$

$y(p) > 0, \forall p \in Q$

\[
\sum_{p \in V} y(p) > 0
\]

where $V = \{v|v \in \cdot(\|y^*\|) \setminus \|y^*\|\}$, and $Q = \{q \in P, q \in \|y^*\|\}$

- Can lead us to numerical problems:
  - the lower the sum, the higher the value of optimisation function
  - We need to find a value $h$ strictly positive s.t. $\sum_{p \in V} y(p) \geq h$
Performance Analysis of PNs (X)

\[
\begin{align*}
\text{maximum } h & \\
\text{subject to } y \cdot C &= 0 \\
y \cdot m_0 &= 1 \\
y &\geq h \cdot 1 \\
h &> 0
\end{align*}
\]

where \( V = \{ v | v \in \text{•}(\|y^*\|) \setminus \|y^*\| \} \), and \( Q = \{ q \in P, q \in \|y^*\| \} \).

\[
\begin{align*}
\text{maximum } y \cdot \text{Pre} \cdot D & \\
\text{subject to } y \cdot C &= 0 \\
y \cdot m_0 &= 1 \\
y(p) &\geq h, \forall p \in Q \\
\sum_{p \in V} y(p) &\geq h
\end{align*}
\]
Performance Analysis of PNs (XI)

• **Input data:**
  - A PPN
  - An accuracy degree

• **Algorithm steps**
  1. Calculate initial upper throughput bound and initial bottleneck cycle
  2. Calculate value \( h \)
  3. Iterate until no significant improvement is achieved or all places are considered
     1. Compute the next constraining \( p \)-semiflow
     2. Calculate new thr. bound
A Running Example (I)
A Running Example (II)
A Running Example (III)
A Running Example (IV)

<table>
<thead>
<tr>
<th>Number of requests</th>
<th>Regrowing step</th>
<th>Size</th>
<th>Throughput</th>
<th>Partial improvement</th>
<th>Bound error</th>
<th>Execution time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(full system)</td>
<td>61 (100%)</td>
<td>56 (100%)</td>
<td>0.525685</td>
<td>-</td>
<td>&gt; +1 day</td>
</tr>
<tr>
<td>15</td>
<td>(initial bound)</td>
<td>56 (91.80%)</td>
<td>56 (100%)</td>
<td>0.551637</td>
<td>-</td>
<td>4.7045%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.87s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57 (93.44%)</td>
<td>56 (100%)</td>
<td>0.533037</td>
<td>3.3718%</td>
<td>1.3792%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>122.94s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58 (95.08%)</td>
<td>56 (100%)</td>
<td>0.522379</td>
<td>1.9995%</td>
<td>-0.6330%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>751.20s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59 (96.72%)</td>
<td>56 (100%)</td>
<td>0.522346</td>
<td>0.0063%</td>
<td>-0.6393%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>34256.97s</td>
</tr>
<tr>
<td>20</td>
<td>(full system)</td>
<td>61 (100%)</td>
<td>56 (100%)</td>
<td>0.652313</td>
<td>-</td>
<td>&gt; +1 day</td>
</tr>
<tr>
<td></td>
<td>(initial bound)</td>
<td>56 (91.80%)</td>
<td>56 (100%)</td>
<td>0.735930</td>
<td>-</td>
<td>11.3621%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.80s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57 (93.44%)</td>
<td>56 (100%)</td>
<td>0.675957</td>
<td>8.1493%</td>
<td>3.4979%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>302.60s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58 (95.08%)</td>
<td>56 (100%)</td>
<td>0.637812</td>
<td>5.6431%</td>
<td>-2.2735%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>300.17s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59 (96.72%)</td>
<td>56 (100%)</td>
<td>0.637860</td>
<td>-0.0075%</td>
<td>-2.2658%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3166.09s</td>
</tr>
<tr>
<td>21</td>
<td>(full system)</td>
<td>61 (100%)</td>
<td>56 (100%)</td>
<td>0.671806</td>
<td>-</td>
<td>&gt; +1 day</td>
</tr>
<tr>
<td></td>
<td>(initial bound)</td>
<td>9 (14.75%)</td>
<td>9 (16.07%)</td>
<td>0.740741</td>
<td>-</td>
<td>9.3063%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57 (93.44%)</td>
<td>56 (100%)</td>
<td>0.697133</td>
<td>5.8871%</td>
<td>3.6331%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>826.82s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58 (95.08%)</td>
<td>56 (100%)</td>
<td>0.653556</td>
<td>6.2509%</td>
<td>-2.7924%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>280.46s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59 (96.72%)</td>
<td>56 (100%)</td>
<td>0.653116</td>
<td>0.0673%</td>
<td>-2.8616%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2216.06s</td>
</tr>
<tr>
<td>22</td>
<td>(full system)</td>
<td>61 (100%)</td>
<td>56 (100%)</td>
<td>0.687808</td>
<td>-</td>
<td>&gt; +1 day</td>
</tr>
<tr>
<td></td>
<td>(initial bound)</td>
<td>9 (14.75%)</td>
<td>9 (16.07%)</td>
<td>0.740741</td>
<td>-</td>
<td>7.1459%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57 (93.44%)</td>
<td>56 (100%)</td>
<td>0.713762</td>
<td>3.6422%</td>
<td>3.6362%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2763.5s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>58 (95.08%)</td>
<td>56 (100%)</td>
<td>0.666148</td>
<td>6.6709%</td>
<td>-3.2515%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>502.95s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>59 (96.72%)</td>
<td>56 (100%)</td>
<td>0.667222</td>
<td>-0.1612%</td>
<td>-3.0853%</td>
</tr>
<tr>
<td>23...30</td>
<td>(full system)</td>
<td>61 (100%)</td>
<td>56 (100%)</td>
<td>0.700056</td>
<td>-</td>
<td>&gt; +1 day</td>
</tr>
<tr>
<td></td>
<td>(initial bound)</td>
<td>9 (14.75%)</td>
<td>9 (16.07%)</td>
<td>0.740741</td>
<td>-</td>
<td>5.4925%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14 (22.95%)</td>
<td>13 (23.21%)</td>
<td>0.740733</td>
<td>0.0011%</td>
<td>5.4915%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.262s</td>
</tr>
</tbody>
</table>
A Running Example (V)
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
Conclusions (I)

• Modelling with Petri nets:
  – Well-known formalism
  – Useful for modelling:
    • concurrent;
    • distributed; and
    • asynchronous behaviour in a discrete system
  – Extensions depending on the application domain
Conclusions (II)

• Performance Analysis in PNs
  – Exact analysis: exploration of state space
    • Becomes unachievable for long/complex systems
  – **Approximate analysis: upper/lower thr. bounds**
    • Linear Programming problems: *good accuracy-computational complexity trade-off*
    • Proposed approach based on an iterative algorithm
      – Takes initial thr. bound and refines it in each iteration
      – Accurate upper bound in few iterations
      – Outputs:
        » Accurate estimate for the steady state thr
        » Subnet representing bottleneck of the system
Outline

• Basic concepts
• Extensions
• Properties
• Analysis
  – Performance analysis
• Conclusions
• References
References

Modelling and Analysis of Non-Functional Properties in Critical Systems with Petri Nets

Elena Gómez-Martínez
Ricardo J. Rodríguez
{elena.gomez, rjrodriguez}@fi.upm.es

Babel Group
Universidad Politécnica de Madrid