Reconfiguration in Stochastic Petri Nets

PhD Student: Samir Tigane
Supervisors: Laïd Kahloul and Souheib Baarir

LINFI Laboratory Biskra University, Algeria
Invited member at LIP6.

MoVe
October 11 2019
Plan

1. Motivation Behind Reconfiguration in PNs
2. Contribution: Reconfigurable Generalized Stochastic PNs
3. Comparative Evaluation
4. Conclusion and Perspectives
Why we consider reconfigurability in PNs?
Modern DESs are more likely to be **structurally dynamic and variably interconnected** at run-time.
Modern DESs are more likely to be **structurally dynamic and variably interconnected** at run-time.

PNs are characterized by their **rigid structures**.
Reconfiguration is expressed via transformation rules
• Reconfiguration is expressed via transformation rules
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Transformation rule

Left-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[ \text{Initial Configuration} \]

\[ \text{Context Graph} \]

\[ p_1 \quad t_1 \quad q_1 \]

\[ \text{Obtained Configuration} \]

\[ \text{Intermediate Graph} \]

\[ p_1 \quad t_1 \quad q_1 \]

\[ p_1' \quad t_1' \quad q_1' \]

\[ \text{Right-hand Side} \]

\[ \text{Transformation rule} \]
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Transformation rule

Left-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[ \bigcirc \rightarrow | \rightarrow \bigcirc \]

Right-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[ \bigcirc \rightarrow | \rightarrow \bigcirc \]

\[ \bigcirc \rightarrow | \rightarrow \bigcirc \]

\[ p'_1 \quad t'_1 \quad q'_1 \]

Rule application

Initial Configuration

Initial Configuration

\[ p_0 \]

\[ t_0 \quad p_1 \quad t_1 \quad p_2 \quad t_2 \]
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Transformation rule

Left-hand Side

$p_1 \quad t_1 \quad q_1$

$\bigcirc \rightarrow | \rightarrow \bigcirc$

Rule application

Initial Configuration

$p_0$

$t_0 \quad p_1 \quad t_1 \quad p_2 \quad t_2$

Right-hand Side

$p_1 \quad t_1 \quad q_1$

$\bigcirc \rightarrow | \rightarrow \bigcirc$

$p_1' \quad t_1' \quad q_1'$
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs
Evaluation
Conclusion and Perspectives

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs

Transformation rule

Left-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[ \bullet \rightarrow | \rightarrow \bullet \]

Right-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[ \bullet \rightarrow | \rightarrow \bullet \]

\[ \bullet \rightarrow | \rightarrow \bullet \]

\[ p'_1 \quad t'_1 \quad q'_1 \]

Rule application

Initial Configuration

Context Graph

\[ \bullet \]

\[ p_0 \]

\[ t_0 \quad p_1 \quad t_1 \quad p_2 \quad t_2 \]

\[ t_0 \quad p_0 \]

\[ t_0 \quad t_2 \]
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Transformation rule

Left-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[
\begin{array}{c}
\bigcirc \\
\rightarrow \\
\| \\
\rightarrow \\
\bigcirc
\end{array}
\]

Right-hand Side

\[ p_1 \quad t_1 \quad q_1 \]

\[
\begin{array}{c}
\bigcirc \\
\rightarrow \\
\| \\
\rightarrow \\
\bigcirc
\end{array}
\]

Rule application

Initial Configuration

Context Graph

Intermediate Graph
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

 Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Transformation rule

Left-hand Side

\[
p_1 \quad t_1 \quad q_1
\]

\[
\circ \rightarrow \mid \rightarrow \circ
\]

Right-hand Side

\[
p_1' \quad t_1' \quad q_1'
\]

\[
\circ \rightarrow \mid \rightarrow \circ
\]

Rule application

Initial Configuration

Context Graph

\[
\begin{align*}
p_0 & \quad p_1 \quad t_1 \quad p_2 \quad t_2 \\
t_0 & \quad p_1 \quad t_1 \quad p_2 \quad t_2
\end{align*}
\]

Intermediate Graph

Obtained Configuration

\[
\begin{align*}
p_0 & \quad p_1 \quad t_1 \quad q_1 \\
t_0 & \quad p_1 \quad t_1 \quad q_1
\end{align*}
\]

\[
\begin{align*}
p_0 & \quad p_1 \quad t_1 \quad q_1 \\
t_0 & \quad p_1 \quad t_1 \quad q_1
\end{align*}
\]
Successive applications of a rewriting rule.

(a) 2\textsuperscript{nd} application.

(b) 3\textsuperscript{rd} application.

(c) 4\textsuperscript{th} application........
Research Question:
A trade-off between modeling and verification level must be found.
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Modeling reconfigurable systems by Reconfigurable Petri nets
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

YES

RG is finite?

YES

RG is finite?
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren't PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

YES

RG is finite?

YES

Execute GTS and compute RG
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

YES

RG is finite?

YES

Execute GTS and compute RG

High temporal and spatial Complexity
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

YES

is structurally bounded?

YES

RG is finite?

High temporal and spacial Complexity

Execute GTS and compute RG

Systems having infinite RGs are not considered
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs
Evaluation
Conclusion and Perspectives

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

- YES
- is structurally bounded?

- NO
- RG is finite?

Enode or Unfold into PNs
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs
Evaluation
Conclusion and Perspectives

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

- YES
  - is structurally bounded?

- NO
  - RG is finite?

Enode or Unfold into PNs

Using off-the-shelf tools
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs
Evaluation
Conclusion and Perspectives

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

- YES
  - is structurally bounded?

- NO
  - RG is finite?

- Enode or Unfold into PNs

  - Reconfigurations are limited to topology level
  - Using off-the-shelf tools

Using off-the-shelf tools
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

- YES
  - is structurally bounded?
  - NO
    - RG is finite?
      - Enode or Unfold into PNs
      - Contribution: Allowing unrestricted reconfiguration forms & Transforming into (GS)PNs

Reconfigurations are limited to topology level

Using off-the-shelf tools
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

Restrict reconfiguration in order to preserve some properties
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs
Evaluation
Conclusion and Perspectives

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

NO

Restrict reconfiguration in order to preserve some properties

Preserved properties are decidable with reduced complexity

Aren’t PNs already enough?
Benefits of having reconfigurable PNs
Challenging of having reconfigurable PNs
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Modeling reconfigurable systems by Reconfigurable Petri nets

is structurally bounded?

Restrict reconfiguration in order to preserve some properties

Too restricted reconfiguration forms & consider only live, bounded and reversible ordinary PNs

Preserved properties are decidable with reduced complexity
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Aren’t PNs already enough?

Benefits of having reconfigurable PNs

Challenging of having reconfigurable PNs

Contribution: Enriching reconfiguration forms & Considering more properties & Considering any GSPN

Modeling reconfigurable systems by Reconfigurable Petri nets

Restrict reconfiguration in order to preserve some properties

Too restricted reconfiguration forms & consider only live, bounded and reversible ordinary PNs

Preserved properties are decidable with reduced complexity

is structurally bounded?
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Definition

Example

Properties Preservation

---

Contribution: Reconfigurable Generalized Stochastic PNs
RecGSPNs $^1$ introduce three major advantages:

---

RecGSPNs\(^1\) introduce three major advantages:

- reconfiguration any GSPN at run-time while preserving several properties,

---

RecGSPNs ¹ introduce three major advantages:

- reconfiguration any GSPN at run-time while preserving several properties,

- a wider range of possible structural changes,

RecGSPNs\(^1\) introduce three major advantages:

- reconfiguration any GSPN at run-time while preserving several properties,
- a wider range of possible structural changes,
- decidability with reduced complexity (infinite graph).

A RecGSPN consists of:
A RecGSPN consists of:

- a GSPN modeling an initial configuration,
A RecGSPN consists of:

- a GSPN modeling an initial configuration,
- set of rewriting rules.
A RecGSPN consists of:

- a GSPN modeling an initial configuration,
- set of rewriting rules.

Both sides of any rule must show a specific behavior.
A net $N$ having interface $(I_N, O_N)$ can be used if its container net $G$ is a live, bounded and reversible GSPN:

![Diagram of a net with interfaces and transitions](image-url)
A net $N$ having interface $(I_N, O_N)$ can be used if its container net $G$ is a live, bounded and reversible GSPN:
A net $N$ having interface $(I_N, O_N)$ can be used if its container net $G$ is a live, bounded and reversible GSPN:
A net $N$ having interface $(I_N, O_N)$ can be used if its container net $G$ is a live, bounded and reversible GSPN:
A net $N$ having interface $(I_N, O_N)$ can be used if its container net $G$ is a live, bounded and reversible GSPN:
A net $N$ having interface $(I_N, O_N)$ can be used if its container net $G$ is a live, bounded and reversible GSPN:
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

<table>
<thead>
<tr>
<th>Left-hand Side</th>
<th>Initial Configuration</th>
<th>Context Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1 \quad t_1 \quad q_1$</td>
<td>$p_0$</td>
<td>$p_0$</td>
</tr>
<tr>
<td><img src="left-hand-side.png" alt="Diagram" /></td>
<td><img src="initial-configuration.png" alt="Diagram" /></td>
<td><img src="context-graph.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Right-hand Side</th>
<th>Intermediate Graph</th>
<th>Obtained Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1 \quad t_1 \quad q_1$</td>
<td>$p_0$</td>
<td>$p_0$</td>
</tr>
<tr>
<td>$p'_1 \quad t'_1 \quad q'_1$</td>
<td><img src="intermediate-graph.png" alt="Diagram" /></td>
<td><img src="obtained-configuration.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs

Definition
Example
Evaluation
Conclusion and Perspectives

Properties Preservation
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs
Evaluation
Conclusion and Perspectives

Definition
Example
Properties Preservation

Left-hand Side

Initial Configuration

Context Graph

Right-hand Side

Intermediate Graph

Obtained Configuration
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

**Definition**

**Example**

**Properties Preservation**

---

**Left-hand Side**

<table>
<thead>
<tr>
<th>Initial Configuration</th>
<th>Context Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 ) ( t_1 ) ( q_1 )</td>
<td>( p_0 )</td>
</tr>
<tr>
<td>( l_{IN} )</td>
<td>( p_0 )</td>
</tr>
</tbody>
</table>

**Right-hand Side**

<table>
<thead>
<tr>
<th>Intermediate Graph</th>
<th>Obtained Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 ) ( t_1 ) ( q_1 )</td>
<td>( p_0 )</td>
</tr>
<tr>
<td>( l_{IN} )</td>
<td>( p_0 )</td>
</tr>
</tbody>
</table>

\( l_{IN} \) \( p_1 \rightarrow t_1 \rightarrow q_1 \)

\( l_{IN} \) \( p_1' \rightarrow t_1' \rightarrow q_1' \)
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Definition

Example

Properties Preservation

Left-hand Side

\[\begin{align*}
   l_N & \quad \text{Initial Configuration} & \quad \text{Context Graph} \\
   p_1 & \quad t_1 & \quad q_1 \\
   p_1' & \quad t_1' & \quad q_1'
\end{align*}\]

Right-hand Side

\[\begin{align*}
   l_N & \quad \text{Intermediate Graph} & \quad \text{Obtained Configuration} \\
   p_1 & \quad t_1 & \quad q_1 \\
   p_1' & \quad t_1' & \quad q_1'
\end{align*}\]
Why do we consider reconfigurability in PNs?

**Contribution:** Reconfigurable Generalized Stochastic PNs

**Evaluation**

**Conclusion and Perspectives**
Reconfiguration preserves
Reconfiguration preserves

- liveness, boundedness, reversibility, home-state, and deadlock-freedom.
Reconfiguration preserves

- liveness, boundedness, reversibility, home-state, and deadlock-freedom.

- $\text{LTL}_{\sim\text{next}}$ properties involving only preserved places.
Reconfiguration preserves

- liveness, boundedness, reversibility, home-state, and deadlock-freedom.

- \( \text{LTL}_{\text{next}} \) properties involving only preserved places.

These properties are decidable with reduced complexity, even if the obtained graph is infinite.
Evaluation
What is the impact of increasing modeling power on decidability?
Why we consider reconfigurability in PNs?  
Contribution: Reconfigurable Generalized Stochastic PNs

### Modeling and verification features

<table>
<thead>
<tr>
<th>Formalism</th>
<th>Modeling</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>US</td>
<td>+/- P/T</td>
</tr>
<tr>
<td>NRS</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RPN</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>R-TNCES</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>INRS</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Evolving PN</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>R-SPNs</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>GSPNs-RT</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td><strong>RecGSPNs</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Modeling and verification features: Existing approaches v.s RecGSPNs.

where US, BS, QUAL and QUAN stand for unbounded structure, bounded structure, qualitative, and quantitative.
What is the impact of introducing reconfigurability on spacial and temporal complexity?
Assume a reconfigurable manufacturing system composed of machine $M_1$ permanently active and $n$ machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.
Assume a reconfigurable manufacturing system composed of machine $M_1$ permanently active and $n$ machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.
Assume a reconfigurable manufacturing system composed of machine $M_1$ permanently active and $n$ machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.
Assume a reconfigurable manufacturing system composed of machine $M_1$ permanently active and $n$ machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.
Assume a reconfigurable manufacturing system composed of machine $M_1$ permanently active and $n$ machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.
Assume a reconfigurable manufacturing system composed of machine $M_1$ permanently active and $n$ machines each of which is plugged to the system when the number of raw materials in a buffer exceeds a threshold.
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Modeling and verification features

Performance

Initial and second configurations.
RMS with two machines (based on basic GPSNs).
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Modeling and verification features
Performance

Factor 1: Model size.

Number of nodes

Number of machines

GSPNs
RecGSPNs

Factor 1: Model size.
Why we consider reconfigurability in PNs?

Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Conclusion and Perspectives

Modeling and verification features

Performance

<table>
<thead>
<tr>
<th># of machines</th>
<th>RecGSPNs</th>
<th>GSPNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>100 states</td>
<td>1963 states</td>
</tr>
<tr>
<td>3</td>
<td>220 states</td>
<td>11340 states</td>
</tr>
<tr>
<td>4</td>
<td>421 states</td>
<td>State space explosion!</td>
</tr>
<tr>
<td>5</td>
<td>743 states</td>
<td>State space explosion!</td>
</tr>
</tbody>
</table>

Factor 2: Semi-Markov chains size according to the number of machines.
Why we consider reconfigurability in PNs?
Contribution: Reconfigurable Generalized Stochastic PNs

Evaluation

Modeling and verification features
Performance

Conclusion and Perspectives

Factor 3: Time to compute semi-Markov chains.

Number of machines

Time in seconds

GSPNs
RecGSPNs

Factor 3: Time to compute semi-Markov chains.
Conclusion and Perspectives
Summary

We have proposed a formalism, called RecGSPNs, that allows to preserve several important properties after each reconfiguration.
Summary

We have proposed a formalism, called RecGSPNs, that allows to preserve several important properties after each reconfiguration.

These properties are decidable with reduced complexity even if the structure can be infinite.
Perspectives

We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.
- Reducing the quantitative verification complexity.
- The quantitative properties of structurally unbounded systems.
We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.
We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.
- Reducing the quantitative verification complexity.
Perspectives

We aim to consider:

- Enriching the set of possible reconfiguration forms of RecGSPNs.
- Reducing the quantitative verification complexity.
- The quantitative properties of structurally unbounded systems.
Selected Papers


Thanks!
Any Questions?