Diagnosis of supervision patterns on bounded labeled Petri nets by Model Checking

Yannick Pencolé\textsuperscript{1}, Audine Subias\textsuperscript{1,2}
\textsuperscript{1}LAAS-CNRS, Toulouse, France
\textsuperscript{2}INSA de Toulouse, France

DX-17 25-29th September 2017
Introduction

- Diagnosis of discrete event systems
- Formalism: Petri nets
- Extension of the fault diagnosis problem to the pattern diagnosis problem (more general)
- Proposed algorithm: off-line diagnosis that takes advantage and efficiency of a Model-Checking tool
Background

- Fault Diagnosis of DES: finding whether a set of fault events have effectively occurred based on a fault model and a sequence of observed events
- Lot of work on this topic (diagnosis, diagnosability analysis)
- More recently, extension of the problem to supervision pattern [Jeron, Cordier et al. 2006] (automaton-based)
- Separate the model from the diagnosis objectives
- Define and implement a generic and automatic method
- Consider the Petri net framework
- Our recent work: bring the problem to Petri nets and analyse the diagnosability of patterns by Model-Checking [Gougam, Pencolé, Subias 2017]
- This paper: provide a first diagnostic algorithm for patterns of Petri nets based on a Model-checking tool
A Petri Net (PN for short) is a 3-uple $N = \langle P, T, A \rangle$ where:

- $P$ is a finite set of nodes called places;
- $T$ is a finite set of nodes called transitions and $P \cap T = \emptyset$;
- $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs between places and transitions.
Labeled Petri Net (LPN)

Definition: LPN

A Labeled Petri Net (LPN for short) is a 5-uple $N = \langle P, T, A, \ell, \Sigma \rangle$ where:

- $P$ is a finite set of nodes called places;
- $T$ is a finite set of nodes called transitions and $P \cap T = \emptyset$;
- $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs between places and transitions.
- $\Sigma$ is a finite set of transition labels (events);
- $\ell : T \rightarrow \Sigma$ is the transition labeling function;
Labeled Prioritized Petri Net (LPPN)

Definition: LPPN

A Labeled Prioritized Petri Net (LPPN for short) is a 6-uple \( N = \langle P, T, A, \succ, \ell, \Sigma \rangle \) where:

- \( P \) is a finite set of nodes called places;
- \( T \) is a finite set of nodes called transitions and \( P \cap T = \emptyset \);
- \( A \subseteq (P \times T) \cup (T \times P) \) is the set of arcs between places and transitions.
- \( \succ \) priority relation, \( (t_1 \succ t_2) \) means that if \( t_1 \) and \( t_2 \) are both enabled then only \( t_1 \) is firable.
- \( \Sigma \) is a finite set of transition labels (events);
- \( \ell : T \rightarrow \Sigma \) is the transition labeling function;
L-type Labeled Prioritized Petri Net (LLPPN)

**Definition: LLPPN**

An L-type Labeled Prioritized Petri Net (LLPPN for short) is a 7-uple $N = \langle P, T, A, \succ, \ell, \Sigma, Q \rangle$ where:

- $P$ is a finite set of nodes called places;
- $T$ is a finite set of nodes called transitions and $P \cap T = \emptyset$;
- $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs between places and transitions.
- $\succ$ priority relation, $(t_1 \succ t_2)$ means that if $t_1$ and $t_2$ are both enabled then only $t_1$ is firable.
- $\Sigma$ is a finite set of transition labels (events);
- $\ell : T \rightarrow \Sigma$ is the transition labeling function;
- $Q$ is the set of final markings.
System example

- $\Sigma_o = \{c, d\}$
- $\Sigma_u = \{a, b\}$
- Transition $b$ has priority to Transition $a$
Example: some system runs

- \((p_0p_1): \varepsilon\)
- \((p_0p_1) \xrightarrow{b} (p_0p_3): b\)
- \((p_0p_1) \xrightarrow{b} (p_0p_3) \xrightarrow{a} (p_1p_3): ba\)
- \((p_0p_1) \xrightarrow{b} (p_0p_3) \xrightarrow{a} (p_1p_3) \xrightarrow{d} (p_4, p_5): bad\)
- ...

Full set of runs: language \(L(\Theta)\).
A pattern $\Omega$ is an LLPPN

$$\Omega = \langle P, T, A, \succ, \ell, \Sigma, Q, M_0 \rangle$$

such that

1. $M_0 \notin Q$ i.e. the language of the pattern does not contain $\varepsilon$;
2. from any reachable marking $M$ of $R(\Omega, M_0)$ there exists a sequence of transitions $r \in T^*$ such that $M \xrightarrow{r} M'$ and $M' \in Q$;
3. from any reachable marking $M$ of $R(\Omega, M_0)$ there is no event $e \in \Sigma$ that labels more than one firable transition;
4. $\forall M \in Q, \forall M': (\exists t \in T, M \xrightarrow{t} M') \implies M' \in Q$;
5. $\succ = \emptyset$, (no priority).
Pattern: $k$ occurrences of an event $e$

$Q = \{kp_2\}$
Pattern: \( n \) occurrences of events (no fixed order)

- Pattern 1: \( n \) occurrences (\( n \) multiple faults): \( Q = \{ \forall p_{2i}, M(p_{2i}) = 1 \} \)
- Pattern 2: \( 0 < r \leq n \) occurrences among the \( n \): \( Q = \{ \sum_{p_{2i}} M(p_{2i}) = r \} \)
Pattern: two occurrences of three consecutive requests $\text{Req}_1$ without occurrences of $\text{Req}_2$

$Q = \{2p_4\}$
Pattern Diagnosis - Problem statement

Implement a diagnosis function that takes as input a sequence of observations and returns one of the following three results:

1. The pattern’s behavior occurred during the system’s evolution.
2. The pattern’s behavior did not occur during the system’s evolution.
3. It is possible that the behavior described by the pattern occurred during the evolution of the system.
Matching of a pattern

**Definition: Sequence matching**

A sequence $\rho \in \Sigma^*$ matches another sequence $\sigma \in \Sigma^*$, denoted $\rho \equiv \sigma$, if:

▶ $\sigma$ is empty ($\sigma = \varepsilon$) ; or

▶ $\sigma = s.\sigma_1$, $s \in \Sigma$, $\sigma_1 \in \Sigma^*$ and there exist two sequences $\rho_0, \rho_1 \in \Sigma^*$ such that:

1. $\rho = \rho_0 s \rho_1$ ;
2. $\rho_1 \equiv \sigma_1$.

A sequence $\rho$ matches a sequence $\sigma$ if $\rho$ contains any event of the sequence $\sigma$ and these events occur in the same order as in $\sigma$.

**Definition: Matching of a pattern**

A sequence $\rho \in \Sigma^*$ matches a pattern $\Omega$ ($\rho \equiv \Omega$) if there exists at least a sequence $\sigma$ of $\mathcal{L}(\Omega)$ that is matched by $\rho$ ($\rho \equiv \sigma$).
Matching of a pattern

\[
\begin{align*}
\rho_0 & \quad s_0 & \quad \rho_1 \\
\sigma & \\
\begin{array}{c}
\rho \\
\sigma
\end{array} & \\
\begin{array}{c}
s_0 \\
\sigma_1
\end{array}
\end{align*}
\]

\[
\begin{align*}
\rho_{10} & \quad s_1 & \quad \rho_2 \\
\sigma & \\
\begin{array}{c}
\rho \\
\sigma
\end{array} & \\
\begin{array}{c}
s_1 \\
\sigma_2
\end{array}
\end{align*}
\]
Diagnosis of a pattern

For a given pattern $\Omega$, the diagnosis problem consists in defining an $\Omega$-diagnoser function

**Definition: $\Omega$-diagnoser**

Let $\Theta$ be the model of a system based on the set of events $\Sigma = \Sigma_u \cup \Sigma_o$, an $\Omega$-diagnoser is a function

$$\Delta_\Omega : \Sigma^* \rightarrow \{\Omega-certain, \Omega-safe, \Omega-ambiguous\}$$

such that :

- $\Delta_\Omega(\sigma) = \Omega-certain$ if for any run $\rho \in \mathcal{L}(\Theta)$ that is consistent with $\sigma$ (i.e. $P_{\Sigma \rightarrow \Sigma_o}(\rho) = \sigma$, $\rho \ni \Omega$);
- $\Delta_\Omega(\sigma) = \Omega-safe$ if for any run $\rho \in \mathcal{L}(\Theta)$ that is consistent with $\sigma$, $\rho \not\ni \Omega$;
- $\Delta_\Omega(\sigma) = \Omega-ambiguous$ otherwise.
Diagnosis of a pattern set

Considering now a set of patterns $\Omega_1, \ldots, \Omega_n$, the diagnoser function of a system can be defined as

**Definition: Diagnoser**

$$\Delta : \Sigma_o^* \rightarrow \prod_{i=1}^{n} \{\Omega_i - \text{certain}, \Omega_i - \text{safe}, \Omega_i - \text{ambiguous}\}$$

such that

$$\Delta(\sigma) = (\Delta_{\Omega_1}(\sigma), \ldots, \Delta_{\Omega_n}(\sigma)).$$
Combination of the system and the pattern
Combination of the system and the pattern (2)

Combination operator: $\Theta_\Omega = \Theta \times \Omega$.

**Theorem: Pattern system product**

Let $\Theta$ be the LLPPN of a system over the alphabet $\Sigma$ and $\Omega$ be the LLPPN of a pattern:

$$\mathcal{L}(\Theta \times \Omega) = \{\rho \in \mathcal{L}(\Theta) : \rho \Vdash \Omega\}.$$

Any run of the system $\Theta$ will lead to a marking in $\Theta_\Omega$. Any run of the system $\Theta$ that matches $\Omega$ will lead to an accepting marking for $\Omega$. 
Combination of the pattern-system and the observations

1. observations: sequence of events $\sigma = o_1 o_2 o_3 o_1 o_4 \ldots$.
2. transform the sequence into a LLPPN $O$ (a sequence of place/transitions), final place $p_{obs}$.
3. synchronise the transitions of $O$ with the transitions of $\Theta_\Omega$ with the same events.
4. Resulting LLPPN: $\Theta_\Omega \parallel O$

Theorem: observation synchronisation

Any run of $\Theta$ that generates $\sigma$ is a run of $\Theta_\Omega \parallel O$ that leads to a marking $M$ such that $M(p_{obs}) = 1$. 

LAAS CNRS
Model checking: principle

- Representation of a system (set of states $S$, initial state $S_0$) by a Kripke structure: $M$
- Property to check: logical formula $\varphi$ (in LTL, CTL, etc)
- Checking an LTL property,
  - Do we have for any path $\pi$ from $S_0$
    $$ M, \pi \models \varphi? $$
- In practice a model-checker returns:
  - TRUE if $\varphi$ is checked,
  - FALSE + counter-example (a system run where $\varphi$ does not hold)
Kripke structure

Definition: Classical Kripke Structure

A Kripke structure over a set of atomic propositions $P$ is a state machine $M = (S, S_0, R, L)$ such that:

1. $S$ is a set of states;
2. $S_0 \subseteq S$ is a set of initial states;
3. $R \subseteq S \times S$ is a total ordered transition relation, for any state $s \in S$, there exists at least a state $s' \in S$ such that $R(s, s')$ is true;
4. $L : S \rightarrow 2^P$ is a labelling function: each state is labelled with the set of atomic propositions that are true in this state.
Some checkings

Propositions $P = \{p_1, p_2, p_3\}$ Initial state $s_1$

Example

Formulas $\varphi$ (Linear Temporal Logic (LTL))

- $\bigcirc p_2$ : in any path, in any successor of $s_1$, $p_2$ is true? TRUE.
- $\Diamond p_3$ : in any path, is there a state with property $p_3$? FALSE. Counterexample: $s_1 \rightarrow s_2 \rightarrow s_2 \ldots$.
- $\Box p_2$ : in any path, is $p_2$ true in any state? FALSE. Counterexample: $s_1 \rightarrow s_3 \rightarrow s_4$ et $M, s_4 \not\models p_2$.
- $\Box(p_1 \Rightarrow \Diamond p_2)$ : in any path, is $p_1 \Rightarrow \Diamond p_2$ true in any state? TRUE. ($p_1 \Rightarrow \Diamond p_2$ means $p_2$ will be eventually true if $p_1$ is true).
TINA : Model-checking tools for (Time) Petri Net

- TINA : (Time) Petri Net Analyser
- Simulation, Invariance Analysis, Model-Checking
- Convert the Petri Net into an Enriched Kripke Structure
- Use the State/Event LTL logic as an input language to write $\psi$
A formula $\psi$ is a SE-LTL formula if it is a universally quantified formula

$$\psi ::= \forall \phi$$

such that

$$\phi ::= r \mid \neg \phi \mid \phi \lor \phi \mid \phi \land \phi \mid \bigcirc \phi \mid \Box \phi \mid \Diamond \phi \mid \phi U \phi$$

$$r ::= e \mid e \triangle e$$

$$e ::= p \mid a \mid c \mid e \nabla e$$

with $p$ a place symbol, $a$ a transition symbol, $c \in \mathbb{N}$, $\triangle \in \{=, <, >, \leq, \geq\}$ and $\nabla \in \{+, -, *, /\}$. The operators $\bigcirc$ (next), $\Box$ (always), $\Diamond$ (eventually) and $U$ (until) have their usual LTL semantics.
Diagnosis by model checking

- Given $M$ a marking of $O \parallel \Theta_{\Omega}$
- $M\upharpoonright \Omega$ restriction of $M$ to $\Omega$
- $M\upharpoonright O$ restriction of $M$ to $O$

Here are the questions:

1. Is it always true ($\square$) that if the system generates $\sigma$ ($M\upharpoonright O \in Q_O$) then ($\Rightarrow$) it matches the pattern ($M\upharpoonright \Omega \in Q_{\Omega}$): 

$$\phi_{CERTAIN} \equiv \square((M\upharpoonright O \in Q_O) \Rightarrow (M\upharpoonright \Omega \in Q_{\Omega}))$$

2. Is it always true ($\square$) that if the system generates $\sigma$ ($M\upharpoonright O \in Q_O$) then ($\Rightarrow$) it does not match the pattern ($M\upharpoonright \Omega \not\in Q_{\Omega}$): 

$$\phi_{SAFE} \equiv \square((M\upharpoonright O \in Q_O) \Rightarrow (M\upharpoonright \Omega \not\in Q_{\Omega}))$$
Case study: conveyors/lift
Monitoring of the occurrence of multiple events

\[ \varphi_{CERTAIN}^3 = \square((p_{obs} = 1) \Rightarrow (p_{21} = 1 \land p_{22} = 1)). \]
## Results

<table>
<thead>
<tr>
<th>Obs.</th>
<th>$\phi^3_{\text{CERT.}}$</th>
<th>$\phi^3_{\text{SAFE}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Pr1$</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>$Pr2$</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>$D$</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>$U$</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>$D$</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>$ELReq2$</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>$ERReq2$</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>$ELReq1$</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>$ERReq1$</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>$U$</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
Pattern: two occurrences of three consecutive requests $\text{Req}_1$ without occurrences of $\text{Req}_2$
### Some results

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Size</th>
<th>nb($Req_1$)</th>
<th>counter</th>
<th>$nS(\Gamma)$</th>
<th>$nT(\Gamma)$</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_1</td>
<td>20</td>
<td>5</td>
<td>1</td>
<td>92</td>
<td>107</td>
<td>11</td>
</tr>
<tr>
<td>S_2</td>
<td>200</td>
<td>5</td>
<td>1</td>
<td>1028</td>
<td>1223</td>
<td>11</td>
</tr>
<tr>
<td>S_3</td>
<td>2000</td>
<td>5</td>
<td>1</td>
<td>10388</td>
<td>12383</td>
<td>235</td>
</tr>
<tr>
<td>S_4</td>
<td>5000</td>
<td>5</td>
<td>1</td>
<td>25988</td>
<td>30983</td>
<td>1105</td>
</tr>
<tr>
<td>S_5</td>
<td>2000</td>
<td>2</td>
<td>10</td>
<td>107534</td>
<td>128379</td>
<td>1993</td>
</tr>
<tr>
<td>S_6</td>
<td>2000</td>
<td>5</td>
<td>10</td>
<td>10388</td>
<td>12383</td>
<td>228</td>
</tr>
<tr>
<td>S_7</td>
<td>2000</td>
<td>10</td>
<td>10</td>
<td>10388</td>
<td>12383</td>
<td>239</td>
</tr>
<tr>
<td>S_8</td>
<td>2000</td>
<td>2</td>
<td>10</td>
<td>7201</td>
<td>8400</td>
<td>156</td>
</tr>
<tr>
<td>S_9</td>
<td>3500</td>
<td>10</td>
<td>10</td>
<td>17009</td>
<td>20266</td>
<td>577</td>
</tr>
<tr>
<td>S_{10}</td>
<td>1000</td>
<td>5</td>
<td>2</td>
<td>907987</td>
<td>2155290</td>
<td>15745</td>
</tr>
<tr>
<td>S_{11}</td>
<td>1000</td>
<td>5</td>
<td>10</td>
<td>1201381</td>
<td>2877901</td>
<td>21018</td>
</tr>
<tr>
<td>S_{12}</td>
<td>2000</td>
<td>10</td>
<td>8</td>
<td>2737559</td>
<td>6482932</td>
<td>65066</td>
</tr>
<tr>
<td>S_{13}</td>
<td>2000</td>
<td>20</td>
<td>8</td>
<td>5999626</td>
<td>14245740</td>
<td>244379</td>
</tr>
<tr>
<td>S_{14}</td>
<td>1750</td>
<td>20</td>
<td>8</td>
<td>4812356</td>
<td>11444801</td>
<td>148629</td>
</tr>
</tbody>
</table>
Conclusion/Perspectives

- Generic framework for pattern modeling and model checking technics
- Translation of the diagnosis problem into a model-checking problem

- Extend the expressivity of patterns
- Time Petri nets