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

Yannick Pencolé[1], Audine Subias[1,2]
[1]LAAS-CNRS, Toulouse, France
[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 oberved 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,Subias2017]
- This paper : provide a first diagnostic algorithm for patterns of Petri nets based on a Model-checking tool

Petri Net (PN)

Definition: PN

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 $P \cap T = \varnothing$;
- $A \subseteq (P \times T) \cup (T \times P)$ 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 $P \cap T = \varnothing$;
- $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs between places and transitions.
- Σ is a finite set of transition labels (events) ;
- $\ell : T \to \Sigma$ 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 $P \cap T = \varnothing$;
- $A \subseteq (P \times T) \cup (T \times P)$ 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.
- Σ is a finite set of transition labels (events) ;
- $\ell : T \to \Sigma$ is the transition labeling function ;

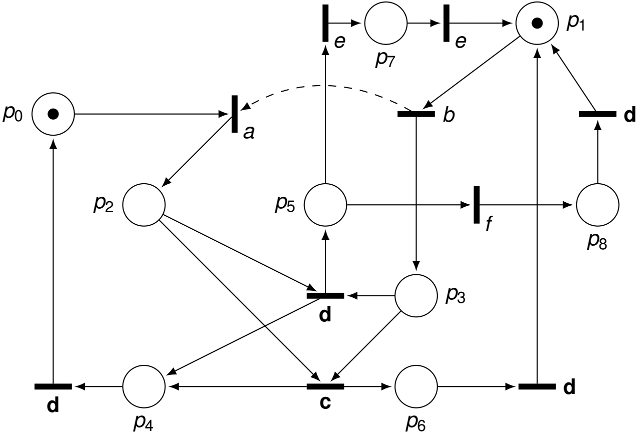
L-type Labeled Prioritized Petri Net (LLPPN)

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 $P \cap T = \varnothing$;
- $A \subseteq (P \times T) \cup (T \times P)$ 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.
- Σ 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

- $(p_0 p_1) : \varepsilon$
- $(p_0 p_1) \xrightarrow{b} (p_0 p_3) : b$
- $(p_0 p_1) \xrightarrow{b} (p_0 p_3) \xrightarrow{a} (p_1 p_3) : ba$
- $(p_0 p_1) \xrightarrow{b} (p_0 p_3) \xrightarrow{a} (p_1 p_3) \xrightarrow{d} (p_4, p_5) : bad$
- $\dots$

Full set of runs : language $\mathscr{L}(\Theta)$.

Pattern

Definition: Pattern

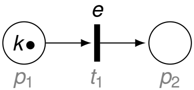
A pattern Ω is an LLPPN

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

such that

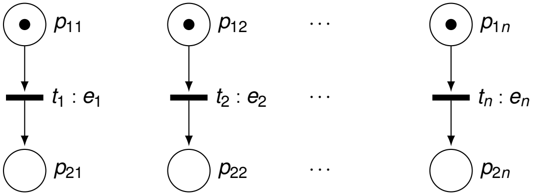
1. $M_0 \notin Q$ i.e. the language of the pattern does not contain ε ;
2. from any reachable marking M of $R(\Omega, M_0)$ there exists a sequence of transitions $r \in T^*$ such that $M \xrightarrow{r} M'$ and $M' \in Q$;
3. from any reachable marking M of $R(\Omega, M_0)$ there is no event $e \in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

Pattern : k occurrences of an event e



$Q = \{\{kp_2\}\}$

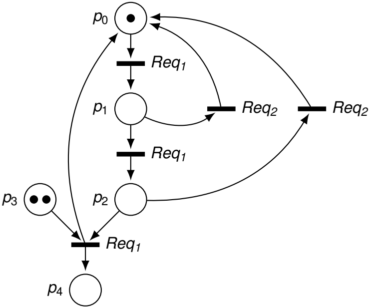
Pattern : n occurrences of events (no fixed order)



- 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 : $Q = \{\{\sum_{p_{2i}} M(p_{2i}) = r\}\}$

LAAS CNRS

Pattern : two occurrences of three consecutive requests Req_1 without occurrences of 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 $\rho \supseteq \sigma$, if :

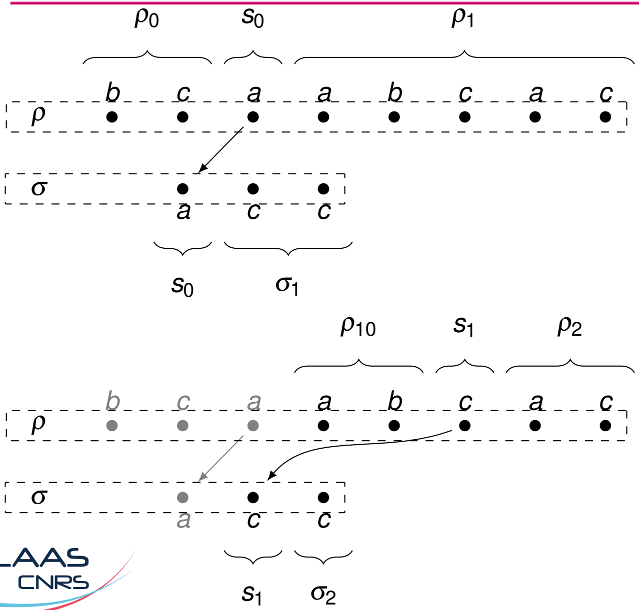
- σ is empty ($\sigma = \varepsilon$) ; or
- $\sigma = s.\sigma_1, s \in \Sigma, \sigma_1 \in \Sigma^*$ and there exist two sequences $\rho_0, \rho_1 \in \Sigma^*$ such that :
  1. $\rho = \rho_0 s \rho_1$;
  2. $\rho_1 \supseteq \sigma_1$.

A sequence ρ matches a sequence σ if ρ contains any event of the sequence σ and these events occur in the same order as in σ.

Definition: Matching of a pattern

A sequence $\rho \in \Sigma^*$ matches a pattern Ω ($\rho \supseteq \Omega$) if there exists at least a sequence σ of $\mathscr{L}(\Omega)$ that is matched by ρ ($\rho \supseteq \sigma$).

Matching of a pattern



LAAS CNRS

Diagnosis of a pattern

For a given pattern Ω, the diagnosis problem consists in defining an Ω-diagnoser function

Definition: Ω-diagnoser

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

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

such that :

- $\Delta_\Omega(\sigma) = \Omega{-}\mathit{certain}$ if for any run $\rho \in \mathscr{L}(\Theta)$ that is consistent with σ (i.e. $\mathscr{P}_{\Sigma \rightarrow \Sigma_o}(\rho) = \sigma$), $\rho \ni \Omega$;
- $\Delta_\Omega(\sigma) = \Omega{-}\mathit{safe}$ if for any run $\rho \in \mathscr{L}(\Theta)$ that is consistent with σ, $\rho \not\ni \Omega$;
- $\Delta_\Omega(\sigma) = \Omega{-}\mathit{ambiguous}$ otherwise.

Diagnosis of a pattern set

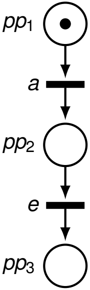
Considering now a set of patterns $\Omega_1, \ldots, \Omega_n$, the diagnoser function of a system can be defined as

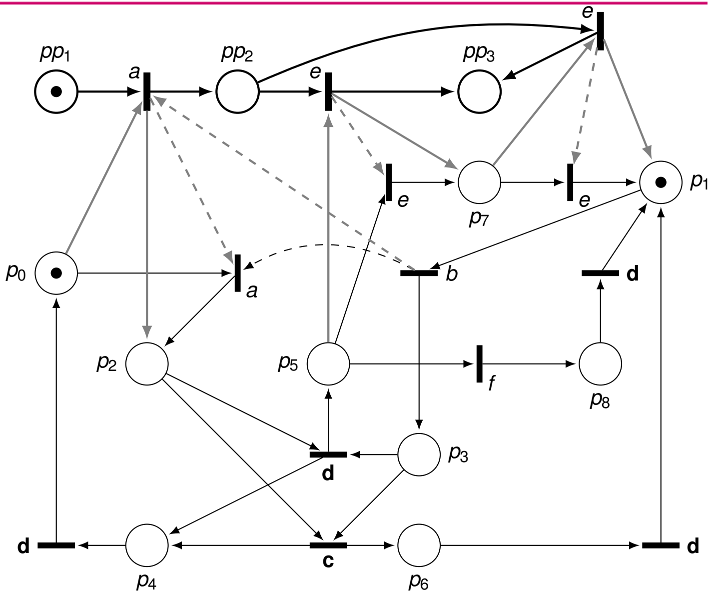
Definition: Diagnoser

$\Delta : \Sigma_o^* \to \prod_{i=1}^{n} \{\Omega_i\text{−}\mathit{certain}, \Omega_i\text{−}\mathit{safe}, \Omega_i\text{−}\mathit{ambiguous}\}$ such that $\Delta(\sigma) = (\Delta_{\Omega_1}(\sigma), \ldots, \Delta_{\Omega_n}(\sigma))$.

LAAS CNRS

Combination of the system and the pattern





Combination of the system and the pattern (2)

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

Theorem: Pattern system product

Let Θ be the LLPPN of a system over the alphabet Σ and Ω be the LLPPN of a pattern :

$$\mathscr{L}(\Theta \ltimes \Omega) = \{\rho \in \mathscr{L}(\Theta) : \rho \Supset \Omega\}.$$

Any run of the system Θ will lead to a marking in Θ_Ω. Any run of the system Θ that matches Ω will lead to an accepting marking for Ω.

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 Θ_Ω with the same events.
4. Resulting LLPPN : $\Theta_\Omega || O$

Theorem: observation synchronisation

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

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 φ (in LTL,CTL,etc)
- Checking an LTL property,
  - Do we have for any path π from S_0

$$M, \pi \models \varphi?$$

- In practice a model-checker returns :
  - TRUE if φ is checked,
  - FALSE + counter-example (a system run where φ does not hold)

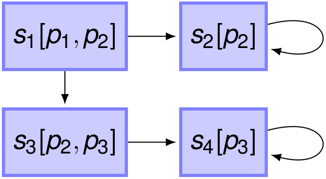
Kripke structure

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. $L : S \to 2^P$ 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 φ (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 : $s_1 \to s_2 \to s_2 \ldots$.
- $\Box p_2$: in any path, is p_2 true in any state ? FALSE. Counterexample : $s_1 \to s_3 \to s_4$ 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).

LAAS CNRS

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 ψ

SE-LTL formulas

A formula ψ is a SE-LTL formula if it is a universally quantified formula

$$\psi ::= \forall \varphi$$

such that

$$\begin{aligned}
\varphi & ::= & r \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \varphi \mathbf{U} \varphi \\
r & ::= & e \mid e \bigtriangleup e \\
e & ::= & p \mid a \mid c \mid e \bigtriangledown e
\end{aligned}$$

with p a place symbol, a a transition symbol, $c \in \mathbb{N}$, $\bigtriangleup \in \{=, <, >, \leq, \geq\}$ and $\bigtriangledown \in \{+, -, *, /\}$. 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 $O \| \Theta_\Omega$
- $M_{|\Omega}$ restriction of M to Ω
- $M_{|O}$ restriction of M to O

Here are the questions :

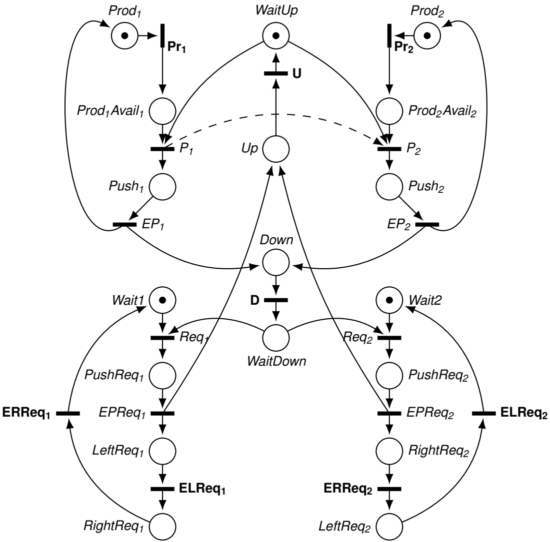
1. Is it always true ($\Box$) that if the system generates σ ($M_{|O} \in Q_O$) then ($\Rightarrow$) it matches the pattern ($M_{|\Omega} \in Q_\Omega$) :

$$\varphi_{CERTAIN} \equiv \Box((M_{|O} \in Q_O) \Rightarrow (M_{|\Omega} \in Q_\Omega))$$

2. Is it always true ($\Box$) that if the system generates σ ($M_{|O} \in Q_O$) then ($\Rightarrow$) it does not match the pattern ($M_{|\Omega} \notin Q_\Omega$) :

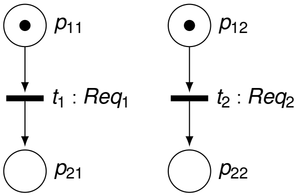
$$\varphi_{SAFE} \equiv \Box((M_{|O} \in Q_O) \Rightarrow (M_{|\Omega} \notin Q_\Omega))$$

Case study : conveyors/lift



LAAS CNRS

Monitoring of the occurrence of multiple events

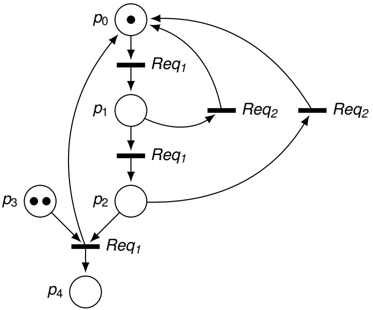


$$\varphi^3_{CERTAIN} = \Box((p_{obs} = 1) \Rightarrow (p_{21} = 1 \land p_{22} = 1)).$$

Results

Obs.	$\varphi^3_{CERT.}$?	φ^3_{SAFE} ?
Pr1	F	T
Pr2	F	T
D	F	T
U	F	F
D	F	F
ELReq2	F	F
ERReq2	F	F
ELReq1	T	F
ERReq1	T	F
U	T	F

Pattern : two occurrences of three consecutive requests Req_1 without occurrences of Req_2



Some results

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

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