Timed Diagnosability Analysis based on Chronicles

H.E. Gougam A. Subias. Y. Pencolé

CNRS; LAAS; 7, avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France (e-mail: {gougam,subias,pencole}@laas.fr)

Abstract: Automated chronicle recognition is an efficient and robust method for fault diagnosis in timed discrete-event systems (TDES). This paper addresses the problem of diagnosability of TDES with regards to such a diagnosis method. We propose a fully automated chain to a priori check whether faults can be identified with certainty based on a given set of chronicles. To deal with the time aspects inherent to the chronicles, we first propose an automated translation of chronicles into a set of Labeled Time Petri Nets with Priorities. The diagnosability analysis is then performed on the state class graph of these nets and consists in determining whether the recognition of a chronicle is exclusive or not.

Keywords: diagnosis, diagnosability, discrete event systems, chronicles, Petri nets.

1. INTRODUCTION

To ensure that a diagnosis tool is able to isolate faults with certainty in a system, it must be able to observe a particular amount of information when the system operates. It is the reason why, nowadays, the design of such a tool must take place during the design of the system itself. The capability of a diagnosis tool to isolate faults with certainty is usually called diagnosability. Diagnosability covers a set of formal properties that have been studied for many years in different fields. In the context of discrete-event systems (DES) for instance, diagnosability analysis usually consists in analyzing, off-line, the system trajectories with respect to their observability to determine whether the diagnostic tool is able to diagnose faults online (Sampath et al. (1995)).

The work presented in this article proposes to extend the diagnosability analysis to the Timed Discrete Event Systems (TDES). The aim is to improve the characterization of diagnosability for a discrete event system by taking into account the notion of finite durations between the occurrence of two events. By introducing time, the objective is to determine whether durations between events must be taken into account by the diagnostic tool to improve the overall diagnosability. As opposed to the work of Khoumsi and Ouédraogo (2009), diagnosability problem here relies on a chronicle-based approach of diagnosis (Laborie and Krivine (1997), Cordier and Dousson (2000)) where the knowledge about the underlying system is gathered in a set of chronicles. The occurrence of a fault is diagnosed by analyzing the flow of observations and matching this flow with a set of available *chronicles* (Dousson et al. (1993)). A chronicle is a partial order of events with time constraints and is associated with the occurrence of a fault. For each chronicle recognized, the diagnosis approach then automatically returns the associated fault as a diagnostic candidate that explains the observed behavior. The set of chronicles used to perform the diagnosis is designed by expertise and/or learning techniques and is a crucial step. By introducing time intervals between two events, the state-space associated to the set of possible trajectories of a TDES is usually infinite. In order to perform any diagnosability analysis, it is then necessary to use an abstraction of this state-space. Some previous works address this problem and have defined a chronicle-based diagnosability analysis as a language-based analysis (Pencolé and Subias (2009)). The challenge of such diagnosability analysis is to deal with the combinatorial explosion of the chronicle instances. To solve this issue, we propose to use Time Petri Nets and to perform the diagnosability analysis on their State Class Graph. Our proposal is to formally extend the work of Pencolé and Subias (2009) by defining a full chain of systematic steps to analyse the diagnosability of a set of chronicles with the help of time Petri nets.

The paper is organized as follows. Section 2 recalls some preliminaries and describes the principles of our chronicle-based diagnosability analysis method. Section 3 presents the formal method to translate chronicles to Labeled Time Petri Nets with Priorities. Section 4 defines the composition step of the Petri nets that characterises the state-space to explore when performing the analysis. Finally, Section 5 explains how the analysis is performed on the resulting State Class Graph to obtain diagnosability results and Section 6 illustrates this approach with an example.

2. PRELIMINARIES AND ANALYSIS PRINCIPLE

2.1 Preliminaries

The underlying system is supposed to behave like a timed model $\mathcal{M} = (E, T)$ with E the set of dated event trajectories of the system (i.e. the system language) and T the set of time constraints between the occurrence dates of the events (see Figure 2). Among the events produced by the system some of them are observable as well as their occurrence date so that an observable timed model (E_{OBS}, T_{OBS}) can be characterized out of $\mathcal{M}$ by projection. Any trajectory $\sigma_{OBS} \in E_{OBS}$ results from the observable projection $\mathrm{P_{OBS}}(\sigma)$ where σ is a trajectory of E. $\mathrm{P_{OBS}}$ is the classical projection operator of each trajectory σ that only retains in σ_{OBS} the observable events of σ(Sampath et al. (1995)). The set of temporal constraints restricted to observable events T_{OBS} is obtained by considering T as a system of inequalities where the dates of unobservable events occurrences are removed by the basic inequalities operations (see Section 6). Finally, W_{OBS} is the set of observable trajectories that the system can generate (without taking the events date into account).

In a chronicle based diagnosis approach, the knowledge that is available about the system is gathered into a set of *chronicles*, also called a *chronicle base*. A chronicle model c is a pair $(\mathcal{S}, \mathcal{T})$ where $\mathcal{S}$ is a set of observable events and $\mathcal{T}$ a set of constraints between their occurrence dates. To define a chronicle model, a human-friendly language has been developed (Dousson et al. (1993)) based on predicates like **evt**, **noevt**, **occurs**... see Section 3.2. The set of trajectories of the system leading to the recognition of the chronicle c is called the recognition language $L(c)$. Each chronicle is also associated to its observable recognition language $\mathcal{C}$, that is the set of observable projections of any trajectory of $L(c)$. Each abnormal situation or *fault* f (i.e. a fault event like in (Sampath et al. (1995)) or a fault pattern like in Jéron et al. (2006) or in Khoumsi and Ouédraogo (2009)) has a signature $Sig(f)$ that is the observable behavior of the system when the fault occurs. A chronicle model $c(f)$ is associated to a fault f when its observable recognition language $\mathcal{C}_f$ is a subset of the fault signature $\mathcal{C}_f \subseteq Sig(f)$.

2.2 General principle of the analysis

In Pencolé and Subias (2009), under the single fault assumption, checking the diagnosability of a fault f relying only on a set of chronicles requires checking whether two chronicles $c(f)$ and $c(f')$ are exclusive or not. Two chronicles are exclusive if they cannot be recognized with the same flow of event instances. It has been shown that the proposed exclusiveness analysis can be performed relying on two kinds of inputs:

- Check for the non exclusiveness of chronicles $c(f_1)$ and $c(f_2)$: if $\mathcal{C}_{f_1} \cap \mathcal{C}_{f_2} \neq \varnothing$ then f_1 and f_2 are not diagnosable.
- Check for the non exclusiveness between a chronicle $c(f)$ and a non faulty model of the monitored system $c(f_0)$: if $\mathcal{C}_f \cap \mathcal{C}_{f_0} \neq \varnothing$ then f is not diagnosable and more precisely f is not detectable.

Note that in the case where $\mathcal{C}_f = Sig(f)$ checking for the exclusiveness allows to conclude on the diagnosability property. We propose in this paper a fully automated and formal method to perform these exclusiveness tests based on the available chronicle base and the system model $\mathcal{M}$. This method relies on three main steps:

Translation: The objective of this step is to translate each chronicle model into Labeled Time Petri Net with Priorities (LTPNPr). Labels are used for modeling the events in the chronicle and priorities are introduced to manage conflicts due to time evolution. This step is developed in Section 3. Note that the objective of this step is not to model a chronicle instance or the recognition language of the chronicle model: the Petri Net represents the chronicle model and the part of the recognition language that is relevant to diagnosability. The chronicle model gives the shortest words of the recognition language that are considered as a faulty (or normal) manifestation. The modeling of these words is sufficient for exclusiveness analysis: at least one of these words must be recognized to claim the fault occurred. Several works address the modeling problem of chronicle recognition relying on modeling tools such as colored Petri nets (Bertrand et al. (2007)).

Product: The exclusiveness test aims to check that the chronicles cannot be recognized by a common trajectory of events. This step aims to construct from the LTPNPr model of each translated chronicle a unique LTPNPr (called *product*) that models the possible common behaviors with synchronized events (see Section 4).

Exclusiveness test: The exclusiveness analysis must deal with an important number of trajectories that may induce the chronicle recognition what is called chronicle instances. These chronicles instances correspond to the marked behaviors of the Petri Net. Thus, coping with the time aspects in terms of delays requires to face to unlimited state space as the complete enumeration of the possible instances of each chronicle is not realistic. Our proposal is to consider a time abstraction of the different instances of a chronicle and to perform the exclusiveness analysis on this time abstraction. The State Class Graph (SCG) of Time Petri Nets gives this finite abstraction.

3. TRANSLATION OF CHRONICLES

3.1 Labeled Time Petri Net with Priorities

Time Petri Nets (TPNs) is a prominent tool to model TDES as several effective analysis methods have been proposed (P.M. Merlin (1976), Berthomieu and Vernadat (2003)). TPNs extend Petri nets with temporal intervals associated to transitions. Firing delay ranges are associated to transitions. A TPN is a tuple $\langle P, T, Pre, Post, m_0, I_s \rangle$ in which $\langle P, T, Pre, Post, m_0 \rangle$ is a Petri net with P the set of places, T the set of transitions, m_0 the initial marking and Pre, $Post$ the forward and backward incidence functions. $I_s : T \to I^+$ is the Static Interval function that associates a time interval to each transition of the net. I^+ is the set of non empty real intervals with non-negative rational end-points. The left-end-point (resp. right-end-point) of the interval associated to a transition t is the static earliest (resp. latest) firing date of t. A TPN state is a couple $s = (m, I)$ in which m is a marking and $I : T \to I^+$ is a function associating a time interval to each transition enabled by m. Initially, $s_0 = (m_0, I_0)$ with I_0 the restriction of I_s to the transitions enabled by m_0. Every enabled transition must be fired in the associated interval. This interval is relative to the enabling date of the transition and depends on the date of the last transition firing. Firing a transition t after a delay θ from state $s = (m, I)$ is possible iff:

$$m \geq Pre(t) \wedge \theta \in I(t)$$
$$\wedge (\forall t' \neq t), (m \geq Pre(t') \Rightarrow \theta \leq sup(I(t'))).$$

The state $s' = (m', I')$ reached by firing the transition t is:

- the new marking m' classically defined by $m' = m - Pre(t) + Post(t)$
- for every transition t' enabled by m':
  (1) $I'(t') = I(t') - \theta$ if $t' \neq t$ and $m - Pre(t) \geq Pre(t')$: the firing of t' is not affected by the firing of t
  (2) $I'(tr) = I_s(tr)$ otherwise.

Time Petri Net with Priorities (TPNPr) is an extension of TPN in which a priority relation on transitions is defined (Berthomieu et al. (2006)). In TPNs the time elapse can only increase the number of firable transitions; the firing of a transition cannot be forbidden by the time elapse. Priorities are used to complete firing conditions. In a TPNPr a transition t may fire from a state $s = (m, I)$ if t is enabled by the marking m, firable instantly and if no transition with higher priority satisfies these conditions.

A marked LTPNPr is a tuple $\langle P, T, \mathbf{Pre}, \mathbf{Post}, \succ, \ell, I_s, m_0 \rangle$, with:

- $\langle P, T, \mathbf{Pre}, \mathbf{Post}, I_s, m_0 \rangle$: a marked Time Petri net;
- $\succ$: $T \rightarrow T$: the priority relation, irreflexive, asymmetric and transitive, defined by its graph $Gr(\succ)$;
- $\ell : T \rightarrow X \cup \{\lambda\}$: the labeling application, where X is an alphabet and λ the empty sequence

Instead of translating each chronicle to a specific LTPNPr, we propose in this section, to develop a direct and systematic method to switch from any chronicle to the corresponding LTPNPr. For this, we consider several basic patterns from which a chronicle model can be composed and several systematic combination templates of these patterns.

3.2 Basic chronicle patterns

We consider six basic patterns derived from the chronicles' description language (Dousson et al. (1993)).

$\mathbf{evt(a,t)} \wedge \mathbf{t} \in [\alpha, +\infty[$**:** an event a occurs after α time units;
$\mathbf{evt(a,t)} \wedge \mathbf{t} \in [\alpha, \beta[$**:** an event a occurs between α and β time units;
$\mathbf{evt(a,t)} \wedge \mathbf{noevent(b, [0,t[)}$**:** an event a occurs without any prior event b;
$\mathbf{noevt(a,} [\alpha, \beta[\mathbf{)}$**:** no event a occurs between α and β time units;
$\mathbf{evt(a,t)} \wedge \mathbf{occurs((m, +\infty), b, [0,t[)}$**:** an event a occurs after at least m events b;
$\mathbf{evt(a,t)} \wedge \mathbf{occurs((m,n), b, [0,t[)}$**:** an event a occurs after at least m and at most n events b;

Due to the lack of space, we only detail here the translation of the basic pattern $\mathbf{evt(a,t)} \wedge \mathbf{occurs((m,n), b, [0,t[)}$. The other translations are simpler and can be found in (Gougam (2011)). The associated LTPNPr is given below and the graphical representation is given by figure 1:

- $P = \{p_{init}, p_1, p_2, p_3, p_4, p_{ok}\}$.
- $T = \{t_1, t_2, t_3, t_4, t_{ok}\}$.
- $\mathbf{Pre} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{m} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{n+1-m} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{Post} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} \end{bmatrix}$.
- $Gr(\succ) = \{(t_2, t_1), (t_4, t_3), (t_4, t_{ok})\}$.
- $\begin{cases} \ell(t_1) = b \\ \ell(t_2) = \lambda \\ \ell(t_3) = b \\ \ell(t_4) = \lambda \\ \ell(t_{ok}) = a \end{cases}, \quad \begin{cases} I_s(t_1) = [0, +\infty[\\ I_s(t_2) = [0, 0] \\ I_s(t_3) = [0, +\infty[\\ I_s(t_4) = [0, 0] \\ I_s(t_{ok}) = [0, +\infty[\end{cases}.$
- $m_0(p) = \begin{cases} 1 & \text{if } p = p_{init} \\ 0 & \text{otherwise} \end{cases}.$

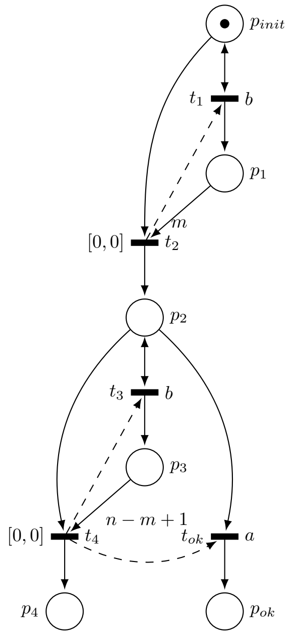


Fig. 1. $\mathbf{evt(a,t)} \wedge \mathbf{occurs((m,n), b, [0,t[)}$

The chronicle is recognized if all the "p_{ok}" places are marked. Initially, the place p_{init} is marked and m events of type b are expected to fire the transition t_2. Then, if at least $n-m+1$ events b occur — i.e. a total of at least $n+1$ events b — the chronicle is not recognized. On the other side, if an event a occurs before the $n-m+1$ events b, then the chronicle is recognized. Finally, priorities ensure compliance of the LTPNPr with the chronicle bounds.

3.3 Pattern combinations

We consider three pattern combinations.

sequence(...): n fully ordered patterns, for example a sequence of two events
$\mathbf{C1}$: $\mathbf{evt(a, t_1)} \wedge \mathbf{evt(b, t_2)} \wedge \mathbf{t_2 - t_1 \geq 4}$.
divergence(...) an initial shared pattern precedes n parallel patterns, for example $\mathbf{C2}$: $\mathbf{evt(a, t_0)} \wedge \mathbf{evt(b, t_3)} \wedge \mathbf{C1} \wedge \mathbf{t_1 - t_0 \geq 4} \wedge \mathbf{t_3 - t_0 \geq 5}$.
convergence(...) n parallel patterns precede a final shared pattern, for example $\mathbf{C3}$: $\mathbf{evt(a, t_0)} \wedge \mathbf{evt(b, t_3)} \wedge C1 \wedge \mathbf{t_3 - t_0 \geq 2} \wedge \mathbf{t_2 - t_3 \geq 5}$.

We propose a systematic way to represent each of them. Let n chronicle models $c_i, i \in \{1, \ldots, n\}$ be represented by a LTPNPr $\langle P_i, T_i, Pre_i, Post_i, \succ_i, \ell_i, I_{s_i}, m_{0i} \rangle$ with $p_{init_i} \in P_i$ and $p_{ok_i} \in P_i$. This LTPNPr of c_i comes from the representation of a basic pattern or the result of a previous combination.

To get $\mathbf{sequence}(c_1, \ldots, c_n)$, the place p_{ok_i} is *merged* with $p_{init_{i+1}}$, formally:

- $P = \bigcup_{i=1}^{n} P_i \backslash \{p_{ok1}, ..., p_{okn-1}\}$ and $T = \bigcup_{i=1}^{n} T_i$.
- $\mathbf{Pre}(p,t) = \begin{cases} \mathbf{Pre}_i(p_{ok_i}, t) & \text{if } (p = p_{init\,i+1}) \text{ and } (t \in T_i), \forall i \in \{1,...,n-1\} \\ \mathbf{Pre}_i(p,t) & \text{if } (p,t) \in P_i \times T_i, \forall i \in \{1,...,n\} \\ 0 & \text{otherwise} \end{cases}$.
- $\mathbf{Post}(p,t) = \begin{cases} \mathbf{Post}_i(p_{ok_i}, t) & \text{if } (p = p_{init\,i+1}) \text{ and } (t \in T_i), \forall i \in \{1,...,n-1\} \\ \mathbf{Post}_i(p,t) & \text{if } (p,t) \in P_i \times T_i, \forall i \in \{1,...,n\} \\ 0 & \text{otherwise} \end{cases}$.
- $Gr(\succ) = \bigcup_{i=1}^{n} Gr(\succ_i)$.
- $\ell(t) = \ell_i(t)$ if $(t \in T_i), \forall i \in \{1,...,n\}$, with: $X = \bigcup_{i=1}^{n} X_i$.
- $I_s(t) = I_{s_i}(t)$ if $(t \in T_i), \forall i \in \{1,...,n\}$.
- $m_0(p) = \begin{cases} 1 & \text{if } p = p_{init1} \\ 0 & \text{otherwise} \end{cases}$.

divergence$(c_1, \ldots, c_n)$ requires adding a place p_{init0} and a transition t_0, this latter will be the preceding transition of all p_{init_i} places:

- $P = \bigcup_{i=1}^{n} P_i \cup \{p_{init0}\}$ and $T = \bigcup_{i=1}^{n} T_i \cup \{t_0\}$.
- $\mathbf{Pre}(p,t) = \begin{cases} \mathbf{Pre}_i(p,t) & \text{if } (p,t) \in P_i \times T_i \\ 1 & \text{if } (p,t) = (p_{init0}, t_0) \\ 0 & \text{otherwise} \end{cases}$, $\forall i \in \{1,...,n\}$.
- $\mathbf{Post}(p,t) = \begin{cases} \mathbf{Post}_i(p,t) & \text{if } (p,t) \in P_i \times T_i \\ 1 & \text{if } (p,t) = (p_{init_i}, t_0) \\ 0 & \text{otherwise} \end{cases}$, $\forall i \in \{1,...,n\}$.
- $Gr(\succ) = \bigcup_{i=1}^{n} Gr(\succ_i)$;
- $\ell(t) = \begin{cases} \ell_i(t) & \text{if } t \in T_i \\ \lambda & \text{otherwise} \end{cases}, \forall i \in \{1,...,n\}$, with: $X = \bigcup_{i=1}^{n} X_i$.
- $I_s(t) = \begin{cases} I_{s_i}(t) & \text{if } t \in T_i \\ [0,0] & \text{otherwise} \end{cases}, \forall i \in \{1,...,n\}$.
- $m_0(p) = \begin{cases} 1 & \text{if } p = p_{init0} \\ 0 & \text{otherwise} \end{cases}$.

convergence$(c_1, \ldots, c_n)$ is not detailed here due to the lack of space but is defined in a very similar way, see (Gougam (2011)).

The translation of any chronicle model will then consist in recursively applying the correct pattern to the chronicle model. For instance, the translation of **C2** consists in applying **sequence(...)** to **C1** first and then **divergence(...)** on **C2**.

4. PRODUCT OF PETRI NET MODELS

As said previously the objective of this step is to represent in a single model the common behavior of two chronicles but also the independent behaviors of each chronicle. Indeed the time intervals at the end of which two chronicles can be recognized with the same event flow can take all the possible relative positions of two intervals (disjunction, overlapping, inclusion).We have then defined a specific product for LTPNPrs obtained by adding transitions labeled with synchronized events (common events) and by adding priorities relations involving these new transitions if necessary. More formally, let $N_1 = \langle P_1, T_1, \mathbf{Pre}_1, \mathbf{Post}_1, \succ_1, \ell_1, I_{s_1}, m_{10}\rangle$ and $N_2 = \langle P_2, T_2, \mathbf{Pre}_2, \mathbf{Post}_2, \succ_2, \ell_2, I_{s_2}, m_{20}\rangle$ be two LTPNPrs, with: $\forall t \in T_i, \ell_i(t) \neq \lambda \Rightarrow I_s(t) = [0, +\infty[$.

Their product $N_1 \star N_2$ is the LTPNPr $\langle P, T, \mathbf{Pre}, \mathbf{Post}, \succ, \ell, I_s, m_0\rangle$ defined as follows:

- $P = P_1 \cup P_2$;
- $T = T_1 \cup T_2 \cup T_s$ where, T_s is the set of synchronized transitions:
$$T_s = \bigcup_{e \in X_1 \cap X_2} \{t_1 \| t_2 : \exists (t_1, t_2) \in (T_1 \times T_2) \text{, with } \ell_1(t_1) = \ell_2(t_2) = e\};$$
- $\mathbf{Pre}(p,t) = \begin{cases} \mathbf{Pre}_1(p,t) & \text{if } (p,t) \in P_1 \times T_1 \\ \mathbf{Pre}_2(p,t) & \text{if } (p,t) \in P_2 \times T_2 \\ \mathbf{Pre}_1(p,t_1) & \text{if } p \in P_1 \wedge t = t_1 \| t_2 \in T_s \\ \mathbf{Pre}_2(p,t_2) & \text{if } p \in P_2 \wedge t = t_1 \| t_2 \in T_s \\ 0 & \text{otherwise} \end{cases}$;
- $\mathbf{Post}(p,t) = \begin{cases} \mathbf{Post}_1(p,t) & \text{if } (p,t) \in P_1 \times T_1 \\ \mathbf{Post}_2(p,t) & \text{if } (p,t) \in P_2 \times T_2 \\ \mathbf{Post}_1(p,t_1) & \text{if } p \in P_1 \wedge t = t_1 \| t_2 \in T_s \\ \mathbf{Post}_2(p,t_2) & \text{if } p \in P_2 \wedge t = t_1 \| t_2 \in T_s \\ 0 & \text{otherwise} \end{cases}$;
- $Gr(\succ) = Gr(\succ_1) \cup Gr(\succ_2) \cup Gr_s$, with:
$$Gr_s = \bigcup_{t_s = t_1 \| t_2} \left\{ \bigcup_{i=1}^{2} \left\{ \bigcup_{(t_i,t) \in Gr(\succ_i)} \{(t_s, t)\} \cup \bigcup_{(t,t_i) \in Gr(\succ_i)} \{(t, t_s)\} \right\} \cup \{(t_s, t_1), (t_s, t_2)\} \right\};$$
- $\ell(t) = \begin{cases} \ell_1(t) & \text{if } t \in T_1 \\ \ell_2(t) & \text{if } t \in T_2 \\ \ell_1(t_1) & \text{if } t = t_1 \| t_2 \in T_s \end{cases}$, with: $X = X_1 \cup X_2$;
- $I_s(t) = \begin{cases} I_{s_1}(t) & \text{if } t \in T_1 \\ I_{s_2}(t) & \text{if } t \in T_2 \\ [0, \infty[& \text{if } t \in T_s \end{cases}$;
- $m_0(p) = \begin{cases} m_{10}(p) & \text{if } p \in P_1 \\ m_{20}(p) & \text{if } p \in P_2 \end{cases}$.

An example of such a product is presented on Figure 4.

5. EXCLUSIVENESS ANALYSIS AND DIAGNOSABILITY RESULTS

Let us consider $SC_f = \{c_1(f), \ldots, c_n(f)\}$ a set of chronicles associated to a fault f and a set of chronicles $SC_{f'} = \{c_1(f'), \ldots, c_n(f')\}$ associated to a fault f'. As previously explained (see Section 2) checking the non exclusiveness between at least one element of SC_f and one element of $SC_{f'}$ allows to conclude to the non diagnosability of the faults f and f'. The exclusiveness test is performed by analyzing the State Class Graph of the *product* LTPNPr built from two chronicles one wants to analyze. The State Class Graph (SCG) is obtained by the grouping of the LTPNPr states in terms of *State Classes* (Berthomieu and Vernadat (2003)). The SCG allows to abstract the time from the behavior of a TPN, therefore the transitions are not labeled with time information (see Figure 5).

From the SCG we first extract W_{OK} the set of trajectories leading to the recognition of the two chronicles (i.e.

leading to the marking of the p_{ok} places). Then, each of these trajectories is compared to W_{OBS} in order to conclude about the exclusiveness or not of the chronicles. If $W_{OK} \bigcap W_{OBS} = \varnothing$ then the two chronicles are exclusive. If furthermore the faulty behavior associated to f (resp f') is totally recognized by SC_f (resp $SC_{f'}$) then system is diagnosable. If $W_{OK} \bigcap W_{OBS} \neq \varnothing$ the two chronicles are not exclusive and the two faults f and f' are not diagnosable. Moreover, the solution of the inequalities system $T_{OBS} \wedge T_c$ gives the precise intervals where the two chronicles are not exclusive. With T_c the time constraints on the paths leading to the recognition of both chronicles.

6. EXAMPLE

We have the system of figure 2, where a and b are normal observable events, σ_{uo} is a normal unobservable event and f_1, f_2 are unobservable faults. We want to study its diagnosability with the following base of chronicles:

- c_0: $\cdot \xrightarrow{[0,8]} a$ (i.e. an event a occurs before 8 time units), associated with the normal behaviour;
- c_1: $\cdot \xrightarrow{[1,5]} b$(i.e. an event b occurs between 1 and 5 time units), associated with f_1;
- c_2: $\cdot \xrightarrow{[3,6]} b$ (i.e. an event b occurs after 3 and before 6 time units), associated with f_2.

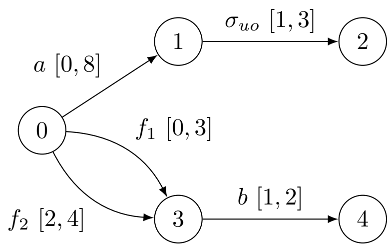


Fig. 2. The Model of the System

We have then: $E = \{(a, t_1), (a, t_1)(\sigma_{uo}, t_2), (f_1, t_3), (f_1, t_3)(b, t_4), (f_2, t_5), (f_2, t_5)(b, t_6)\}$ with:

$$T = \{0 \leq t_1 \leq 8, 2 \leq t_2 - t_1 \leq 3, 0 \leq t_3 \leq 3, 1 \leq t_4 - t_3 \leq 2, 2 \leq t_5 \leq 4, 1 \leq t_6 - t_5 \leq 2\}$$

To study the diagnosability of the system, we must study the detectability of every fault, thus compare the chronicles associated with the normal behaviour with every chronicle associated with a fault, and then study the diagnosability by comparing each fault to the others.

In our example, and for the sake of simplicity, we will present the last step — i.e. comparison between faults.

The projection of the model over the observable space gives:

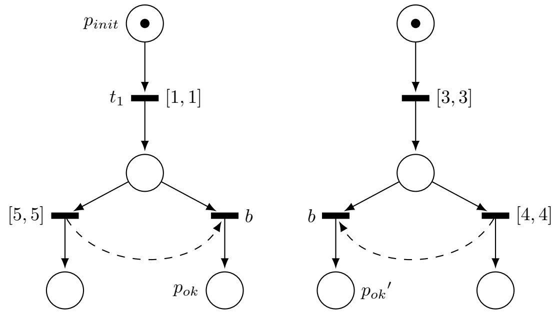
$$E_{OBS} = \{(a, t_1), (b, t_4), (b, t_6)\}$$

and:

$$T_{OBS} = \begin{cases} 0 \leq t_1 \leq 8 \\ 1 \leq t_4 \leq 5 \\ 3 \leq t_6 \leq 6 \end{cases}$$

Figure 3 gives the translation to LTPNPrs of the chronicles c_1 and c_2.

For the chronicle c_1, first the place p_{init1} is marked. After 1 temporal unit, the transition t_1 is fired. Then we have two possibilities, either an event b occurs before 5 time units and in this case the chronicle is recognized, or no event b occurs (before 5 time units) and the chronicle is not recognized.



(a) c_1 (b) c_2

Fig. 3. Translation of the Chronicles

The same reasoning applies to the chronicle c_2.

By applying the algorithm of the product previously described, we obtain the LTPNPr of the figure 4.

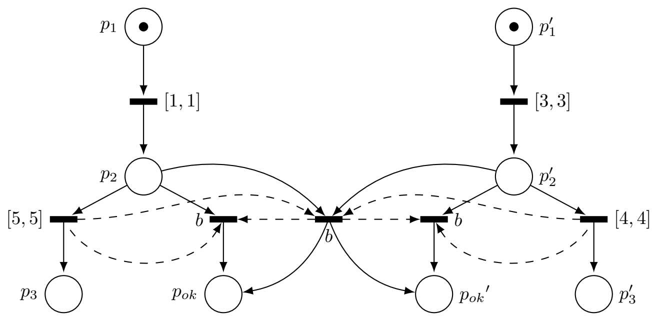


Fig. 4. Product of the Previous LTPNPrs

We simply add a new transition labelled 'b' to represent the synchronized progression of the two chronicles, and put some priorities (represented by dashed arcs) to solve conflicts. Indeed, the transition from which the arc comes out has a higher priority than the transition in which the arc comes in. Thus, in case of simultaneous activation, the transition with a higher priority is triggered.

6.1 *Results*

Figures 5 represent the class graph of the product of the LTPNPrs corresponding to c_1 and c_2.

The analysis is done through several steps.

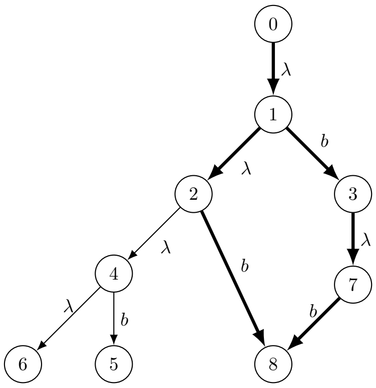
Step 1 In the example, class 8 corresponds to the marking of the places p_{ok} and p'_{ok} (figure 4) i.e. the recognition of the two chronicles. Two paths reach the class 8 (in bold on figure 5), b and bb, so: $W_{ok} = \{b, bb\}$.

Step 2 In the example: $W_{OBS} = \{a, b\}, W_{ok} = \{b, bb\}$ so

$$W_I = W_{ok} \cap W_{OBS} = \{b\} \neq \varnothing$$

Step 3 We look for the set E_{OBS} of event flow that can be generated by the system with the words associated to the paths belonging to W_{ok}, i.e. with the elements of W_I.

The system can generate two different flows 'b':



class 0: p_1, p'_1	class 1: p_1, p'_2	class 2: p_2, p'_2
class 3: p_1, p'_4	class 7: p_2, p'_4	class 8: p_4, p'_4

Fig. 5. The Class Graph and the Marking of Some Classes

- when f_1 happens:
  $$s_{1_{OBS}} = \{(b, t_4)\} \text{ with: } t_{1_{OBS}} = \{\, 3 \leq t_4 \leq 6$$
- when f_2 happens:
  $$s_{2_{OBS}} = \{(b, t_2)\} \text{ with: } t_{2_{OBS}} = \{\, 1 \leq t_2 \leq 5$$

so: $E_{OBS} = s_{1_{OBS}} \cup s_{2_{OBS}}$ and $T_{OBS} = t_{1_{OBS}} \wedge t_{2_{OBS}}$.

We have to compare T_{OBS} with T_c — the temporal constraints associated with the flow E_c leading to the recognition of both chronicles.

Step 4 The only path 'b' leading to the class 8, in our case, is given by: $E_c = \{(b, t'_1)\}$ with: $T_c = \{\, 3 \leq t'_1 \leq 6$

Now, we have to resolve the inequalities system $T_{OBS} \wedge T_c$:

$$\begin{cases} 3 \leq t_4 \leq 6 \\ 1 \leq t_2 \leq 5 \\ 3 \leq t'_1 \leq 6 \\ t_4 = t_2 = t'_1 \end{cases}$$

What leads us to: $3 \leq t'_1 \leq 5$

So, for every flow (b, t'_1) where $3 \leq t'_1 \leq 5$, the two chronicles are not exclusive, so the system is not diagnosable. For every other flow, the two chronicles are exclusive. In this case, we can conclude on the diagnosability of the system only if the observable recognition language of c_1 (resp. c_2) is the signature of the fault f_1 (resp. f_2).

7. CONCLUSION

We presented, in this paper, a systematic chronicle-based approach to analyze diagnosability of timed discrete events systems. First, we developed a method to systematically translate every chronicle to a labeled time Petri nets with priorities, and perform mutual exclusiveness test based on an adequate labeled time Petri nets with priorities product. This method allows us to refine the results of the diagnosability analyses.

A possible continuation of this work would be the study of return on design, i.e. the modification of the chronicle base or the system to suit our needs. Because the performed analysis is valid only for the particular chronicle base, meaning that changing the base can change the result of the diagnosability analysis. So it may be interesting to study the relation between the diagnosability analysis and the choice of the chronicle base, thus helping in designing the latter.

Finally, and since our method relies on the state graph class analysis (we look for some particular paths in these class graphs), we may consider another approach based on formal verification method by transposing the problem to a reachability analysis and using model checking methods to conclude about the diagnosability.

REFERENCES

Berthomieu, B., Peres, F., and Vernadat, F. (2006). Bridging the gap between timed automata and bounded time Petri nets. In *Proceedings of Formal Modeling and Analysis of Timed Systems (FORMATS'06), LCNS 4202.*

Berthomieu, B. and Vernadat, F. (2003). State class constructions for branching analyzis of systems. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'03).* Warsaw, Poland.

Bertrand, O., Carle, P., and Choppy, C. (2007). Chronicle modelling using automata and colored Petri nets. In *18th Internation Workshop on Principles of Diagnosis (DX'07)*, 243–248. Nashville, TN, USA.

Cordier, M. and Dousson, C. (2000). Alarm driven monitoring based on chronicles. In *4th Sumposium on Fault Detection Supervision and Safety for Technical Processes (SafeProcess)*, 286–291. Budapest, Hungary.

Dousson, C., Gaborit, P., and Ghallab, M. (1993). Situation recognition: representation and algorithms. In *IJCAI: International Joint Conference on Artificial Intelligence*, 166–172. Chambéry, France.

Gougam, H. (2011). Diagnosticabilité des systèmes à événement discrets à base de chroniques. Technical report, Université Paul Sabatier.

Jéron, T., Marchand, H., Pinchinat, S., and Cordier, M.O. (2006). Supervision patterns in discrete event systems diagnosis. In *WODES06: Workshop on Discrete Event Systems.*

Khoumsi, A. and Ouédraogo (2009). Diagnosis of faults in real-time discrete event systems. In *SAFEPROCESS'09*, 1557–1562.

Laborie, P. and Krivine, J.P. (1997). Automatic generation of chronicles and its application to alarm processing in power distribution systems. In *8th international workshop of diagnosis (DX97).* Mont Saint-Michel, France.

Pencolé, Y. and Subias, A. (2009). A chronicle-based diagnosability approach for discrete timed-event systems: Application to web-services. *Journal of Universal Computer Science*, 15(17), 3246–3272. doi:http://www.jucs.org/doi?doi=10.3217/jucs-015-17-3246.

P.M. Merlin, D.F. (1976). Recoverability of communication protocols: Implication of a theoretical study. *IEEE Trans. Comm.*, 24(9), 1036–1043.

Sampath, M., Sengputa, R., Lafortune, S., Sinnamohideen, K., and Teneketsis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40, 1555–1575.