Predictability of Sequence Patterns in Discrete Event Systems

Thierry Jéron, Hervé Marchand, Sahika Genc, Stéphane Lafortune

Predictability of Sequence Patterns in Discrete Event Systems

Thierry Jéron * **Hervé Marchand** * **Sahika Genc** **
Stéphane Lafortune ***

* *INRIA Rennes-Bretagne Atlantique, Rennes, France.*
First.Last@irisa.fr
** *General Electric Global Research Center, 1 Research Circle, Niskayuna, NY 12309*
*** *University of Michigan, Ann Arbor, MI, USA.*
stephane@eecs.umich.edu

Abstract: The problem of predicting the occurrences of a pattern in a partially-observed discrete-event system is studied. The system is modeled by a labeled transition system. The pattern is a set of event sequences modeled by a finite-state automaton. The occurrences of the pattern are predictable if it is possible to infer about any occurrence of the pattern before the pattern is completely executed by the system. A novel off-line algorithm to verify the property of predictability is presented. The verification is polynomial in the number of states of the system. An on-line algorithm to track the execution of the pattern during the operation of the system is also presented. This algorithm is based on the use of a diagnoser automaton. *Copyright© 2008 IFAC.*

Keywords: Labelled Transition Systems, Sequence patterns, Prediction

1. INTRODUCTION

Monitoring and fault diagnosis of dynamic systems modeled as discrete event systems has received considerable attention in the literature in the last decade. Several methodologies have been developed for different types of diagnosis problems. This paper takes the diagnosis problem one step ahead, and considers the problem of predicting (faulty) sequence patterns in the behavior of a partially-observed discrete-event system (DES). If it is possible to predict the occurrences of a pattern that describes event sequences that lead or cause the system to fail or malfunction, then the system operator can be warned in time to halt the system, take preventative measures, or otherwise to reconfigure the system.

In this paper, the system is modeled as a partially-observed Labeled Transition System (LTS). Partially-observed DES have observable (e.g., sensor readings, changes in the sensor readings, etc.) and unobservable events, which are events that are not directly recorded by the sensors attached to the system. The sequence pattern, or simply pattern, to be predicted is modeled as a regular language. Pattern consists of ordered occurrences of one or more significant observable or unobservable events; thus, occurrence or prediction of the pattern is not trivial. The problem of prediction of patterns in DES as considered here builds on and extends two prior studies: (i) the problem of *predictability* of single event occurrences in partially-observed DES studied in Genc and Lafortune [2006a] and (ii) the problem of *diagnosis* of sequence patterns in DES studied in Jéron et al. [2006] and Genc and Lafortune [2006b]. The consideration of patterns in the context of the property of predictability requires more than technical modifications to the two works mentioned above; this is true especially for the development of a polynomial-time test for pattern predictability, as will be seen later in the paper.

The notion of predictablility considered in this paper is different from prior works on other notions of predictability in Cao [1989], Buss et al. [1991], Fadel and Holloway [1999]. Other references on predictability in DES or discrete event simulation systems are Das et al. [1995], Chase and Ramadge [1990], Declerck [1998], Musolesi and Mascolo [2006], He et al. [2002], Feiler et al. [2000]. The results presented in this paper are related to the work of Jiang and Kumar Jiang and Kumar [2004] who considered the notion of *inevitability*. Indeed, in this paper, the authors are interested in diagnosing inevitability, thus in detecting that a pattern will be inevitably satisfied within a bounded number of observations *after* this inevitability. In contrast, in the context of this paper, predictability means detecting inevitability *strictly before* its occurrence. More details can be found in Jéron et al. [2007].

The contributions of the paper are

- a generalization of the definition of predictability for patterns. This notion subsumes and strictly extends the prior notion of predictability in Genc and Lafortune [2006a].
- the design of an algorithm for the verification of predictability. The verification is polynomial in the number of states of the system, whereas the one developed in Genc and Lafortune [2006a] is, in the worst case, exponential in the number of states of the system.

- and an algorithm for the construction of a predictor, which predicts, on-line, the recognition of a pattern during the evolution of the system.

The remainder of the paper is as follows: in Section 2, we define the mathematical terminology and notions used throughout the paper. In Section 3, we specify the notions of pattern and diagnosability considered in this paper. We then define the property of predictability of occurrences of a sequence pattern in Section 4. In Section 5, we propose an off-line polynomial-time algorithm for the verification of predictability. In Section 6, we present an algorithm for constructing a special type of diagnoser, named *predictor* for the purpose of on-line prediction of a sequence pattern and emphasize some properties of this predictor.

2. SYSTEM MODEL AND NOTATIONS

Let Σ be a finite alphabet of events. A *string* is a finite-length sequence of events in Σ. Given a string s, the length of s (number of events including repetitions) is denoted by $\parallel s \parallel$. The set of all strings formed by events in Σ is denoted by Σ^*. Any subset of Σ^* is called a *language* over Σ. Let L be a language over Σ. Given a string $s \in L$, L/s is called the *post-language* of L after s and defined as $L/s = \{t \in \Sigma^* \mid s.t \in L\}$.

The system in which the sequence pattern is to be predicted is modeled as an LTS. The formal definition of an LTS is as follows.

Definition 1. (LTS). *An LTS over Σ is defined by a 4-tuple $M = (Q_M, \Sigma, \rightarrow_M, q_M^0)$ where Q_M is a finite set of states, Σ is the set of* events *of M, $q_M^0 \in Q_M$ is the initial state, and $\rightarrow_M \subseteq Q_M \times \Sigma \times Q_M$ is* the partial transition relation. $\diamond$

In the remainder of this section, we consider a given LTS $M = (Q_{\mathrm{M}}, \Sigma, \rightarrow_{\mathrm{M}}, q_{\mathrm{M}}^0)$. We write $q \xrightarrow{\sigma}_{\mathrm{M}} q'$ if $(q, \sigma, q') \in \rightarrow_{\mathrm{M}}$. We extend $\rightarrow_{\mathrm{M}}$ to arbitrary sequences by setting : $q \xrightarrow{\varepsilon}_{\mathrm{M}} q$ for all states q, and $q \xrightarrow{s\sigma}_{\mathrm{M}} q'$ whenever $q \xrightarrow{s}_{\mathrm{M}} q''$ and $q'' \xrightarrow{\sigma}_{\mathrm{M}} q'$, for some $q'' \in Q_{\mathrm{M}}$. We write $q \xrightarrow{s}_{\mathrm{M}}$ if $\exists q' \in Q_{\mathrm{M}}, q \xrightarrow{s}_{\mathrm{M}} q'$. We denote by $\mathcal{L}(M) = \{s \in \Sigma^* \mid q_{\mathrm{M}}^0 \xrightarrow{s}_{\mathrm{M}}\}$ the language generated by M, which corresponds to the set of trajectories that the LTS M can execute. The *event set* of a state $q \in Q_{\mathrm{M}}$ is $\Sigma(q) \triangleq \{\sigma \in \Sigma \mid q \xrightarrow{\sigma}_{\mathrm{M}}\}$. M is *live* if $\Sigma(q) \neq \emptyset$, for each $q \in Q_{\mathrm{M}}$. A state q is *reachable* if $\exists s \in \Sigma^*, q_{\mathrm{M}}^0 \xrightarrow{s}_{\mathrm{M}} q$. We set $\Delta_{\mathrm{M}}(q, s) \triangleq \{q' \in Q_{\mathrm{M}} \mid q \xrightarrow{s}_{\mathrm{M}} q'\}$. By a slight abuse of notation, for any language $L \subseteq \Sigma^*$, $\Delta_{\mathrm{M}}(q, L) \triangleq \bigcup_{s \in L} \Delta_{\mathrm{M}}(q, s)$ and for any $Q' \subseteq Q_{\mathrm{M}}$, $\Delta_{\mathrm{M}}(Q', L) = \bigcup_{q \in Q'} \Delta_{\mathrm{M}}(q, L)$. A subset $Q' \subseteq Q_{\mathrm{M}}$ is said *stable* whenever $\Delta_{\mathrm{M}}(Q', \Sigma) \subseteq Q'$. M is *complete* whenever $\forall q \in Q_{\mathrm{M}}, \Sigma(q) = \Sigma$. We say that M is *deterministic* if, whenever $q \xrightarrow{\sigma}_{\mathrm{M}} q'$ and $q \xrightarrow{\sigma}_{\mathrm{M}} q''$, then $q' = q''$, for all $q \in Q_{\mathrm{M}}$ and all $\sigma \in \Sigma$.

We denote the set of final states by $F_{\mathrm{M}} \subseteq Q_{\mathrm{M}}$. The notions above are extended in this setting by letting the language recognized in F_{M} be

$$\mathcal{L}_{F_{\mathrm{M}}}(M) = \{s \in \Sigma^* \mid \Delta_{\mathrm{M}}(q_{\mathrm{M}}^0, s) \subseteq F_{\mathrm{M}}\}.$$

We now define the synchronous product of two LTS.

Definition 2. Let $M^i = (Q^i, \Sigma, \rightarrow_{M^i}, q_{M^i}^0)$, $i = 1, 2$, be two LTS. Their synchronous product *is $M^1 \times M^2 \triangleq (Q^1 \times Q^2, \Sigma, \rightarrow_{M^1 \times M^2}, (q_{M^1}^0, q_{M^2}^0))$, where $\rightarrow_{M^1 \times M^2} \subseteq (Q^1 \times Q^2) \times \Sigma \times (Q^1 \times Q^2)$ satisfies $(q^1, q^2) \xrightarrow{\sigma}_{M^1 \times M^2} (q'^1, q'^2)$ whenever $q^1 \xrightarrow{\sigma}_{M^1} q'^1$ and $q^2 \xrightarrow{\sigma}_{M^2} q'^2$.* $\diamond$

Clearly $\mathcal{L}(M^1 \times M^2) = \mathcal{L}(M^1) \cap \mathcal{L}(M^2)$ and for $F_i \subseteq Q^i, i = 1, 2$, we also have $\mathcal{L}_{F_1 \times F_2}(M^1 \times M^2) = \mathcal{L}_{F_1}(M^1) \cap \mathcal{L}_{F_2}(M^2)$. Also, if sets F_i are stable in M^i, $F_1 \times F_2$ is stable in $M^1 \times M^2$.

Given a set of states $E \subseteq Q_{\mathrm{M}}$ of M, we define the following sets

$$Pre_{\mathrm{M}}^{\forall}(E) = \{q \in Q \mid \forall \sigma \in \Sigma, \Delta_{\mathrm{M}}(q, \sigma) \subseteq E\}$$

$$Pre_{\mathrm{M}}^{\exists}(E) = \{q \in Q \mid \exists \sigma \in \Sigma, \Delta_{\mathrm{M}}(q, \sigma) \cap E \neq \emptyset\}$$

In words, states belonging to $Pre_{\mathrm{M}}^{\forall}(E)$ are states such that all immediate successors belong to E, while states belonging to $Pre_{\mathrm{M}}^{\exists}(E)$ are states such that at least one immediate successor belongs to E.

Given a live LTS M, let $Inev_M(E)$ be the set of states that inevitably lead to a set E in a finite number of steps. This set is given by:

$$Inev_{\mathrm{M}}(E) = \{q \in Q \mid \exists n \geq 0, s.t. \forall s \in \Sigma^*, q \xrightarrow{s}_{\mathrm{M}} \wedge \parallel s \parallel \geq n \Rightarrow \Delta_{\mathrm{M}}(q, s) \subseteq E\}$$

$Inev_M(E)$ can also be characterized by a least fixpoint (lfp) as follows: $Inev_{\mathrm{M}}(E) = lfp(\lambda X.E \cup Pre_{\mathrm{M}}^{\forall}(X))$

The set of events Σ is partitioned into Σ_o and Σ_{uo}

$$\Sigma = \Sigma_o \cup \Sigma_{uo}, \text{ and } \Sigma_o \cap \Sigma_{uo} = \emptyset,$$

where, as usual, Σ_o is the set of *observable* events while Σ_{uo} is the set of *unobservable* events. In this paper, the elements of Σ_o^* will be denoted by μ, μ'. We say that M is Σ_o-*live* if $\forall q \in Q, \exists s \in \Sigma^*.\Sigma_o, q \xrightarrow{s}$, meaning that there is no terminal loop of unobservable events.

$P : \Sigma^* \rightarrow \Sigma_o^*$ denotes the natural *projection* of trajectories onto Σ_o^*. This is extended to any language $L \subseteq \Sigma^*$ by letting $P(L) = \{P(s) \mid s \in L\}$. We adopt the terminology *traces* for *observable* trajectories; note that while standard in computer science, this notation is not standard in the DES literature. Given an LTS M, the set of traces of M is thus given by $P(\mathcal{L}(M))$. The inverse projection of a language $L \subseteq \Sigma_o^*$ is defined by $P^{-1}(L) = \{s \in \Sigma^* \mid P(s) \in L\}$.

Given a trace μ of M, we define $[\![\mu]\!]$ as the set of trajectories *compatible* with the trace μ:

$$[\![\mu]\!] \triangleq \begin{array}{l} P^{-1}(\mu) \cap \mathcal{L}(M) \cap \Sigma^*\Sigma_o \text{ if } \mu \neq \epsilon \\ \epsilon \text{ otherwise.} \end{array}$$

This means that (except for the empty trace), trajectories compatible with a trace μ are trajectories of M ending with an observable event and having trace μ. This is consistent with an on-line diagnosis where verdicts consider trajectories until the last observation, and not subsequent unobservable events.

Next, we introduce the Unobservable-Closure $\textsc{Uc}(M)$ of an LTS M, in order to abstract out unobservable events according to the semantic $[\![.]\!]$.

Definition 3. For an LTS $M = (Q_M, \Sigma, \rightarrow, q_M^0)$, the Unobservable-Closure of M *is* $\textsc{Uc}(M) = (Q_M, \Sigma_o, \rightarrow_{\textsc{Uc}}, q_M^0)$

where for any $q, q' \in Q_M$, $\sigma \in \Sigma_o$, $q \xrightarrow{\sigma}_{Uc} q'$ *in* $\text{Uc}(M)$ *whenever there exists* $s \in \Sigma_{uo}^*$ *such that* $q \xrightarrow{s\sigma}_M q'$ *in* M. $\diamond$

We get $\mathcal{L}(\text{Uc}(M)) = P(\mathcal{L}(M))$ and $\mathcal{L}_{F_M}(\text{Uc}(M)) = P(\mathcal{L}_{F_M}(M))$

Determinization of an LTS M defined on the alphabet $\Sigma = \Sigma_o \cup \Sigma_{uo}$ produces an LTS $Det(M)$ with actions in Σ_o, and deterministic on Σ_o, with same traces as the M.

Definition 4. Let $M = (Q_M, \Sigma, \rightarrow_M, q_M^0)$ *be an LTS with* $\Sigma = \Sigma_{uo} \cup \Sigma_o$. *The* determinization of M *is the LTS* $Det(M) = (\mathcal{X}, \Sigma_o, \rightarrow_d, X^0)$ *where* $\mathcal{X} = 2^{Q_M}$ *(the set of subsets of* Q_M *called* macro-states*),* $X^0 = \{q_M^0\}$ *and* $\rightarrow_d = \{(X, \sigma, \Delta_M(X, \Sigma_{uo}^*.\sigma)) \mid X \in \mathcal{X} \text{ and } \sigma \in \Sigma_o\}$. $\diamond$

Notice that this definition differs from the classical determinization of automata, but is consistent with the above semantic of observations $[\![.]\!]$: the target macro-state X' of a transition $X \xrightarrow{\sigma}_d X'$ is only composed of states q' of M which are targets of sequences of transitions $q \xrightarrow{s.\sigma} q'$ ending with an observable event σ. As a consequence, $Det(M)$ can be constructed in two steps by first constructing $\text{Uc}(M)$ and then applying the classical subset construction used in the usual determinization of automata.

From the definition of $\rightarrow_d$ in $Det(M)$, we infer that $\Delta_{\text{Det(M)}}(X^0, \mu) = \{\Delta_M(q_M^0, [\![\mu]\!])\}$, which means that the macro-state reached from X^0 by μ in $Det(M)$ is composed of the set of states that are reached from q_M^0 by trajectories of $[\![\mu]\!]$ in M.

Finally, determinization preserves observations, so we have $\mathcal{L}(Det(M)) = P(\mathcal{L}(M))$. Also for $F_M \subseteq Q_M$, we have $\mathcal{L}_{2^{F_M}}(Det(M)) = \{\mu \in \Sigma_o^* \mid [\![\mu]\!] \subseteq \mathcal{L}_{F_M}(M)\}$.

3. PATTERNS AND DIAGNOSABILITY

We now define patterns that describe stable properties, i.e., negation of safety properties; see Jéron et al. [2006] for further details. We also recall the definition of the property of diagnosability. We assume that the system is modeled by an LTS $G = (Q_G, \Sigma, \rightarrow_G, q_G^0)$ that is Σ_o-live and whose event set Σ includes a set of unobservable events Σ_{uo} with associated projection operation P.

Definition 5. A sequence pattern*, or simply pattern, is a deterministic and complete LTS,* $\Omega = (Q_\Omega, \Sigma, \rightarrow_\Omega, q_\Omega^0)$ *equiped with a distinguished stable set* $F_\Omega \subseteq Q_\Omega$. $\diamond$

As Ω is complete we get $\mathcal{L}(\Omega) = \Sigma^*$. Also note that the assumption that F_Ω is stable means that its recognized language is "extension-closed", i.e., it satisfies $\mathcal{L}_{F_\Omega}(\Omega).\Sigma^* = \mathcal{L}_{F_\Omega}(\Omega)$. In other words, $\mathcal{L}_{F_\Omega}(\Omega)$ is a language violating a safety property. In Jéron et al. [2006], examples of patterns often used in diagnosis are presented, including the cases "occurrence of a single fault event", "occurrence of one of multiple fault events in the same partition", "ordered occurrence of events", "multiple occurrences of the same fault", etc. The same approach is used in this paper in the context of the predictability problem.

Given an LTS G and a pattern Ω, we say that a trajectory $s \in \mathcal{L}(G)$ is recognized by the pattern if $s \in \mathcal{L}_{F_\Omega}(\Omega)$. Diagnosability of an LTS G with respect to a pattern Ω equipped with F_Ω, is defined as the ability to detect that a trajectory is recognized by Ω on the basis of the observed projection of the trajectory only, within a bounded number of observations. Formally, we have:

Definition 6. An LTS G *is* Ω-diagnosable *if* $\exists n \in \mathbb{N}$,
$\forall s \in \mathcal{L}_{F_\Omega}(\Omega) \cap \mathcal{L}(G) \cap \Sigma^*\Sigma_o, \forall t \in \mathcal{L}(G)/s \cap \Sigma^*\Sigma_o$,
if $\|P(t)\| \geq n$ *then* $[\![P(s.t)]\!] \subseteq \mathcal{L}_{F_\Omega}(\Omega)$. $\diamond$

Ω-diagnosability says that when a trajectory s ending with an observable event is recognized by the pattern Ω, for any extension t with enough observable events, any trajectory s' compatible with the observation $P(s.t)$ is also recognized by Ω. Details can be found in Jéron et al. [2006].

4. PREDICTABILITY

In this section, we introduce the notion of *predictability of a pattern*. G and Ω are as defined in the preceding section. Roughly speaking, a pattern is predictable if it is always possible to detect the future recognition of the pattern, strictly before this recognition, only based on the observations. We explain the idea of predictability in the following example.

Example 1. Consider the LTS G_1 shown in Fig. 1(a). The set of events is $\Sigma = \{f_1, f_2, a, b, c\}$. Let $\Sigma_{uo} = \{a, f_1, f_2\}$ be the set of unobservable events. We assume that the pattern under consideration is either the occurrence of f_1 followed by f_2 or the one of f_2 followed by f_1. This pattern is given in Fig. 1(b) and denoted by Ω in this example.

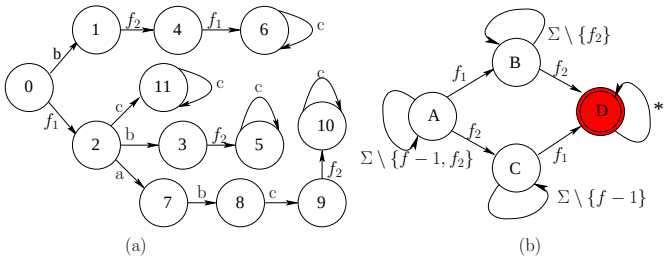

Fig. 1. The LTS G_1 of Example 1 and the Ω pattern

By a simple inspection of G_1 in Fig. 1(a), we see that there are three branches in which f_1 is followed by f_2 (or f_2 followed by f_1): $b.f_2.f_1.c^*$, $f_1.b.f_2.c^*$ and $f_1.a.b.c.f_2.c^*$. The other branch, which does not satisfy the pattern, records $f_1.c.^*$. Thus, after the observation of $b.c.c$, we know for sure that the pattern has been recognized. G_1 is thus Ω-diagnosable. Moreover, we can also predict the recognition of the pattern ahead of time. Indeed, if we do observe b then we know for sure that the pattern has not occur so far but will occur in the future. After b, the system is either is state 1,8 or 3. After the occurrence of c, then G_1 is either is 5, 6,or 9. In the first two cases, the pattern has occurred. In the third case, we need to wait for a second c to occur to be sure that it occurred (the system is then either in state 5, 6, 10, in which the pattern has surely occurred).•

We now formally define the predictability of patterns. As previously said, this generalizes the predictability definition introduced in Genc and Lafortune [2006a] by considering sequence patterns instead of single events.

Definition 7. An LTS G *is* Ω-predictable*, whenever*
$\exists n \in \mathbb{N}$, $\forall s \in \mathcal{L}(G) \cap \mathcal{L}_{F_\Omega}(\Omega) \cap \Sigma^*.\Sigma_o$,
$\exists t \in (\mathcal{L}(G) \cap \Sigma^*.\Sigma_o) \cup \{\epsilon\}$, $t < s \ \wedge \ t \notin \mathcal{L}_{F_\Omega}(\Omega)$
such that
$\forall u \in [\![P(t)]\!], \forall v \in \mathcal{L}(G)/u, \|P(v)\| \geq n \Rightarrow u.v \in \mathcal{L}_{F_\Omega}(\Omega)$

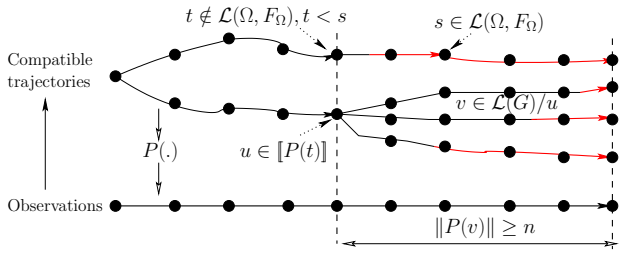

Fig. 2. Intuitive explanation of predictability

◇

The definition means that for any trajectory s recognized by the pattern, there exists a strict prefix t not recognized by the pattern, such that any trajectory u compatible with observation $P(t)$ will inevitably be extended into a trajectory $u.v$ recognized by the pattern. That is, upon observation of $P(t)$, one can already predict that the pattern, while not yet recognized, will inevitably be recognized; see Figure 2 for a conceptual representation.

Let us now relate the notion of Ω-predictability to the notion of Ω-diagnosability of Section 3.

Proposition 1. Given a system G and a pattern Ω, if G is Ω-predictable, then G is also Ω-Diagnosable.

Proof The proof of the proposition immediately follows from the respective definitions of these properties. (Further details on pattern diagnosability can be found in Jéron et al. [2006], Genc and Lafortune [2006b].) ◇

Note that the converse of this proposition is false as shown by the following example:

Example 2. We consider a variation of Example 1. The new system is given by the LTS G_2 represented in Fig. 3. The pattern Ω and the partition of the event set into observable and unobservable events are the same as in Example 1

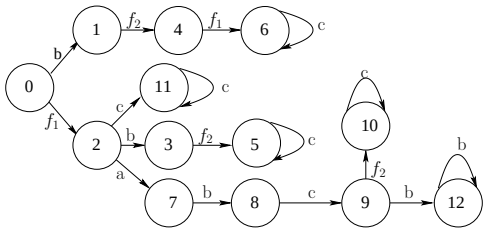

Fig. 3. The LTS G_2 of Example 2

It is easy to show that G_2 is Ω-diagnosable. However, G_2 is not Ω-predictable. Indeed, because of the sequence $f_1.a.b.c.b^*$ that is not recognized by the pattern, one can not be sure that the pattern will occur by observing the trace b. Further, if we observe the trace $b.c$, it may be too late, as the system may have trigger the sequence $f_1.b.f_2.c$, which is a sequence recognized by the pattern. •

5. VERIFICATION OF PREDICTABILITY

In this section, we present an off-line algorithm to verify the property of predictability. In Genc and Lafortune [2006b], where the problem of predictability of single event occurrences is considered, the algorithm for the verification of predictability is based on the construction of a diagnoser automaton. In this work, we propose to adapt the polynomial-time verification of diagnosability based on *verifier automata* Jiang et al. [2001], Yoo and Lafortune [2002] to the verification of predictability. In Jéron et al. [2006], a technique based on verifiers was used to verify the diagnosability of a pattern in polynomial time in the number of states of the system.

In the case of diagnosability, the verifier identifies the existence, if any, of strings that violate the definition of this property. We adopt this underlying principle in building the algorithm for verification of predictability of patterns. By definition of predictability, the trajectories that violate predictability are the ones that: (i) are accepted by the pattern, (ii) end with observable events, and (iii) have the property that none of their prefixes are sufficient to predict the occurrence of the pattern. That is, for all the prefixes of such trajectories, there exists a trajectory that produces the very same observation as the prefix but whose continuation does not contain the pattern. Thus, if we find one or more trajectories violating predictability as was just described, then we can conclude that at least one occurrence of the pattern is not predictable in the system. It is possible to transform the search for these trajectories into a search for states in an LTS that carries information on acceptance, inevitability, and projections, for trajectories leading to them.

In the remainder of this section, we decompose the verification of predictability into five steps, each corresponding to a particular operation required for the verification. We start with system $G = (Q_G, \Sigma, \rightarrow_G, q^0_G)$ and pattern $\Omega = (Q_\Omega, \Sigma, \rightarrow_\Omega, q^0_\Omega)$ equipped with F_Ω.

Algorithm for Verification of Predictability:

Step 1. Construct the synchronous product $G_\Omega = G \times \Omega = (Q_{G_\Omega}, \Sigma, \rightarrow_{G_\Omega}, q^0_{G_\Omega})$ and the final state set $F = Q_G \times F_\Omega$. By the properties of synchronous product, and using the fact that Ω is complete (thus $\mathcal{L}(\Omega) = \Sigma^*$), we get $\mathcal{L}(G_\Omega) = \mathcal{L}(G) \cap \mathcal{L}(\Omega) = \mathcal{L}(G)$ and $\mathcal{L}_F(G_\Omega) = \mathcal{L}(G) \cap \mathcal{L}_{F_\Omega}(\Omega)$. Thus, the accepted trajectories of G_Ω in F are exactly the trajectories of G accepted by Ω in F_Ω. Moreover, note that F is stable in G_Ω as both Q_G and F_Ω are stable by assumption.

Step 2. Compute $Inev_{G_\Omega}(F)$ and consider the following partition of the state space Q_{G_Ω}

$$Q_{G_\Omega} = N \cup I \cup F$$

where $I = Inev_{G_\Omega}(F) \setminus F$ is the set of states not belonging to F but from which F is unavoidable, and $N = Q_{G_\Omega} \setminus Inev_{G_\Omega}(F_{G_\Omega})$ is the set of states from which F is avoidable.

The diagram of Fig. 4 shows how G_Ω evolves from one state to another according to the set of states it belongs to. This diagram illustrates the fact that even though the system can remain forever in N-states, if it reaches an I-state, then after a finite number of steps the system will eventually evolve into an F-state.

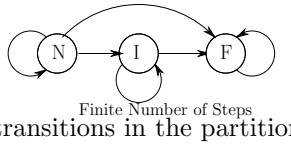

Fig. 4. Possible transitions in the partition of Q_{G_Ω}

The construction of $G_{2\Omega}$ of Example 2 is illustrated in Fig. 5, with N-states unshaded, F-states dark-shaded, and I-states light-shaded with dashed-line borders.

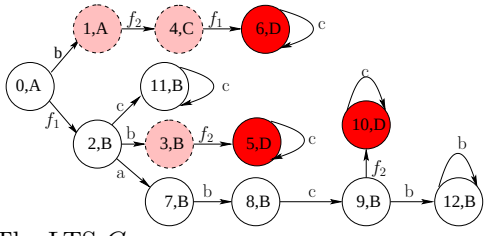

Fig. 5. The LTS $G_{2\Omega}$

Step 3. Construct the Unobservable-Closure (see Def. 3) $\text{Uc}(G_\Omega) = (Q_{G_\Omega}, \Sigma_o, \rightarrow_{\text{Uc}}, q^0_{G_\Omega})$ of G_Ω with final state set F and preservation of the partition $Q_{G_\Omega} = N \cup I \cup F$. Using this partition, label states in $(Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F)$, corresponding to the set of states having an immediate successor in F but not belonging to F. This set characterizes the maximal prefixes t of trajectories of G not recognized by Ω in F_Ω, i.e., their observation is the last chance to predict acceptance of the pattern.

By the properties of Uc, we get $\mathcal{L}(\text{Uc}(G_\Omega)) = P(\mathcal{L}(G_\Omega))$, $\mathcal{L}_F(\text{Uc}(G_\Omega)) = P(\mathcal{L}_F(G_\Omega))$, and $\mathcal{L}_F(\text{Uc}(G_\Omega)) = P(\mathcal{L}_F(G_\Omega))$, which also means that Uc preserves information on the inevitability of the occurrence of the pattern, while abstracting out internal events.

Consider Example 2 again. The Unobservable-Closure of $G_{2\Omega}$ is shown in Fig. 6 where states in $(Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F)$ are those with thick borders.

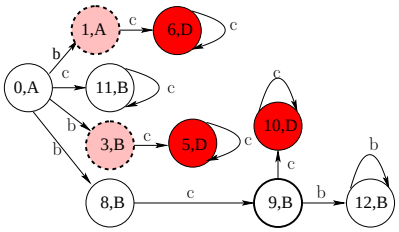

Fig. 6. The LTS $\text{Uc}(G_{2\Omega})$

Step 4. Construct the LTS $\Gamma = \text{Uc}(G_\Omega) \times \text{Uc}(G_\Omega) = (Q_{G_\Omega} \times Q_{G_\Omega}, \Sigma_o, \rightarrow_\Gamma, (q^0_{G_\Omega}, q^0_{G_\Omega}))$.

The role of Γ is to synchronize trajectories of G_Ω with same trace μ. By the definitions of Uc and synchronous product, if $\mu \in \mathcal{L}(G_\Omega)$ and $(q^0_{G_\Omega}, q^0_{G_\Omega}) \xrightarrow{\mu}_\Gamma (q, q')$, then there exists $s, s' \in [\![\mu]\!]$ s.t. $q^0_{G_\Omega} \xrightarrow{s} q$ and $q^0_{G_\Omega} \xrightarrow{s'} q'$ in G_Ω.

Step 5. Check the predictability condition given by Theorem 1 below. This completes the statement of the verification algorithm. ◇

Theorem 1. G is Ω-predictable iff $(Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F) \times N$ is not reachable from $(q^0_{G_\Omega}, q^0_{G_\Omega})$ in Γ.

A reachable state in $(Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F) \times N$ characterizes two trajectories with same observation, one where F_Ω is avoidable, and the other one where it was the last chance to predict acceptance. Observe that the symmetry in Γ allows us to simplify the condition, but in practice, reachable states in $N \times (Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \cap \setminus F)$ should also be detected.

Proof ($\Rightarrow$) Suppose that Γ has a reachable state (q, q') in $(Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F) \times N$. Note that we may have $q = q'$. On the one hand, there exists a trace $\mu \in \Sigma_o^*$, an observable event $\sigma_o \in \Sigma_o$, and states $q \in Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F$, $q_1 \in F$ such that $q^0_{G_\Omega} \xrightarrow{\mu}_{\text{Uc}} q \xrightarrow{\sigma_o}_{\text{Uc}} q_1$. On the other hand, there exists a state $q' \in N$ such that $q^0_{G_\Omega} \xrightarrow{\mu}_{\text{Uc}} q'$. By construction of $\text{Uc}(G_\Omega)$ from G_Ω, there exists $s', u' \in \Sigma^*$ with $P(s') = P(u') = \mu$ (at this point s' may be identical to u') and such that $q^0_{G_\Omega} \xrightarrow{s'}_\Gamma q \xrightarrow{\sigma_o}_\Gamma q_1$ and $q^0_{G_\Omega} \xrightarrow{u'}_\Gamma q'$. By taking $s = s'.\sigma_o$, for any $t \in \Sigma^*.\Sigma_o$ such that $t < s$, we have that $t \in \mathcal{L}(G)$ and $t \notin \mathcal{L}_{F_\Omega}(\Omega)$. But since $P(s') = P(u')$, there exists $u \leq u'$ such that $P(t) = P(u)$. Finally, $u \notin \mathcal{L}_{I \cup F}(G \times \Omega)$, otherwise this would imply that $q' \in I \cup F$ while $q' \in N$. This proves that G is not Ω-predictable.

($\Rightarrow$) Note first that Ω-predictability is equivalent to $\forall s \in \mathcal{L}_{F_\Omega}(\Omega) \cap \mathcal{L}(G) \cap \Sigma^*.\Sigma_o, \exists t \in (\mathcal{L}(G) \cap \Sigma^*.\Sigma_o) \cup \{\epsilon\},\ t < s \wedge t \notin \mathcal{L}_{F_\Omega}(\Omega)$ s.t. $[\![P(t)]\!] \subseteq \mathcal{L}_{Inev(Q_G \times F_\Omega)}(G \times \Omega)$.

Suppose now that G is not Ω-predictable. By the previous statement, $\exists s \in \mathcal{L}_{F_\Omega}(\Omega) \cap \mathcal{L}(G) \cap \Sigma^*.\Sigma_o$, such that $\forall t \in (\mathcal{L}(G) \cap \Sigma^*.\Sigma_o) \cup \{\epsilon\},\ t < s \ \wedge\ t \notin \mathcal{L}_{F_\Omega}(\Omega)$ and $[\![P(t)]\!] \not\subseteq \mathcal{L}_{Inev(Q_G \times F_\Omega)}(G \times \Omega)$. Let $s = s'.s_{uo}.\sigma_o$ with $s' \in \Sigma^*.\Sigma_o$, $s_{uo} \in \Sigma_{uo}^*$, $\Sigma_o \in \Sigma_o$. By construction of $\text{Uc}(G_\Omega)$, there exist states $q \in Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F$, $q_1 \in F$ and $q' \in N$ such that $q^0_{G_\Omega} \xrightarrow{P(s')}_{\text{Uc}} q \xrightarrow{\sigma_o}_{\text{Uc}} q_1$ and $q^0_{G_\Omega} \xrightarrow{P(s')}_{\text{Uc}} q'$. Thus, in Γ, $(q^0_{G_\Omega}, q^0_{G_\Omega}) \xrightarrow{P(s')}_\Gamma (q, q')$ with $(q, q') \in (Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F) \times N$. ◇

Figure 7 depicts Γ_2 for G_2, with the pattern Ω.

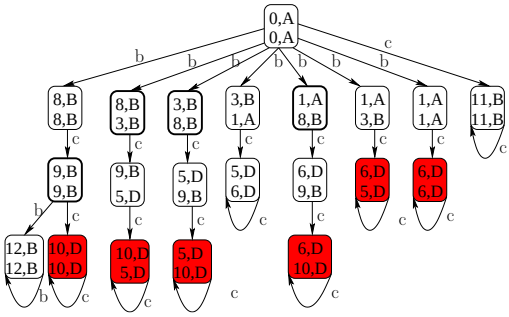

Fig. 7. The LTS Γ_2 derived from $G_{2\Omega}$

The states with thick borders in Γ_2 are states in $(Pre^{\exists}_{\text{UC}(G_\Omega)}(F) \setminus F) \times N$ and prove that G_2 is not Ω-predictable. For example, the string $b.f_2.f_1.c$ with observation $b.c$ already leads to acceptance, but acceptance is not predictable because of trajectories in $f_1.a.b.c.b^*$; Finally, note that G_2 is Ω-diagnosable, as in Γ_2, there are no cycles of states of the form $(I \cup N) \times F$ (see Jéron et al. [2006] for further details on this verification of diagnosability).

We summarize the above algorithm to verify the predictability of the occurrences of a pattern Ω in a system G and denote it by *Verif_Pred* hereafter:

Verif_Pred(G,Ω)

(i) Computation of $G_\Omega = G \times \Omega$
(ii) Computation of the state sets F, I and N in G_Ω

(iii) Computation of $\text{Uc}(G_\Omega)$ and of the state set $(Pre^{\exists}_{\text{Uc}(G_\Omega)}(F) \setminus F)$
(iv) Computation of $\Gamma = \text{Uc}(G_\Omega) \times \text{Uc}(G_\Omega)$
(v) Test of the emptiness of $(Pre^{\exists}_{\text{Uc}(G_\Omega)}(F) \setminus F) \times N$.

Proposition 2. The complexity of the verification of predictability using Verif_Pred(G,Ω) is quadratic in the product of the sizes of G and Ω.

The proof of this result is based on the following analysis of the steps within *Verif_Pred*(G,Ω) . The first step is the construction of $G_\Omega = G \times \Omega$, which is linear in $\|G\| \times \|\Omega\|$ where $\|G\|$ and $\|\Omega\|$ are the sizes of G and Ω in terms of states and transitions. The next step is the computation of $Inev_{G_\Omega}(F)$ to partition Q_{G_Ω} into $N \cup I \cup F$. This is linear in $\|G_\Omega\|$. The third step consists of constructing $\text{Uc}(G_\Omega)$ and labelling of states by $(Pre^{\exists}_{\text{Uc}(G_\Omega)}(F) \setminus F)$. It is linear in $\|G_\Omega\|$. The construction of $\Gamma = \text{Uc}(G_\Omega) \times \text{Uc}(G_\Omega)$ in the fourth step is quadratic in the size of $\text{Uc}(G_\Omega)$. Checking that no reachable state is in $(Pre^{\exists}_{\text{Uc}(G_\Omega)}(F) \setminus F) \times N$ (Step 5) can be done during this construction. Thus, the complexity of the entire verification of predictability is quadratic in the product of the sizes of G and Ω.

6. THE PREDICTOR

In this section, we explain how to construct from G and Ω a special kind of diagnoser, called predictor, that can be used on-line to predict the recognition of a pattern. We emphasize some properties of this predictor when the occurrence of the pattern Ω is predictable in G.

Consider G_Ω as defined in Section 5 (Step 1) and build $Det(G_\Omega) = (\mathcal{X}, \Sigma_o, \rightarrow_d, X^0)$ as defined in Definition 4. Remember that this can be done by reusing $\text{Uc}(G_\Omega)$ and applying a subset construction. Then, we have that $\mathcal{L}(Det(G_\Omega)) = P(\mathcal{L}(G_\Omega)) = P(\mathcal{L}(G))$. Thus, for any observation $\mu \in P(\mathcal{L}(G))$, $\Delta_{Det(G_\Omega)}(X^0, \mu) = \{\Delta_{G_\Omega}(q^0_{G_\Omega}, [\![\mu]\!])\}$.

The predictor for predictability, denoted by $\texttt{Predict}_\Omega$, is defined to be $Det(G_\Omega)$ together with a decision function on its state space that is defined as follows. For any observation μ in $P(\mathcal{L}(G))$, let the decision function of $\texttt{Predict}_\Omega(\mu)$ be:

$$\begin{cases} Yes, & \text{if } \Delta_{Det(G_\Omega)}(X^0,\mu) \subseteq F \\ Pred, & \text{if } \Delta_{Det(G_\Omega)}(X^0,\mu) \subseteq I \\ Pred\_F, & \text{if } \Delta_{Det(G_\Omega)}(X^0,\mu) \subseteq (I \cup F) \\ & \quad \wedge\ \Delta_{Det(G_\Omega)}(X^0,\mu) \cap I \neq \emptyset \\ & \quad \wedge\ \Delta_{Det(G_\Omega)}(X^0,\mu) \cap F \neq \emptyset \\ No, & \text{if } \Delta_{Det(G_\Omega)}(X^0,\mu) \subseteq N \\ Pred\_N & \text{if } \Delta_{Det(G_\Omega)}(X^0,\mu) \subseteq (I \cup N) \\ & \quad \wedge\ \Delta_{Det(G_\Omega)}(X^0,\mu) \cap I \neq \emptyset \\ & \quad \wedge\ \Delta_{Det(G_\Omega)}(X^0,\mu) \cap N \neq \emptyset \\ ?, & \text{otherwise.} \end{cases} \quad (1)$$

In order to better understand the decisions of the predictor, we consider a language point of view. Using the partition of $Q_{G_\Omega} = N \cup I \cup F$, we can partition $\mathcal{L}(G) = \mathcal{L}(G_\Omega)$ into three different languages

$$\mathcal{L}(G) = \mathcal{L}_N(G_\Omega) \cup \mathcal{L}_I(G_\Omega) \cup \mathcal{L}_F(G_\Omega)$$

Based on this partition, we then have the following results that are derived directly from the definition on the predictor:

- $\texttt{Predict}_\Omega(\mu) = Yes$ if $[\![\mu]\!] \subseteq \mathcal{L}_F(G_\Omega)$
- $\texttt{Predict}_\Omega(\mu) = Pred$ if $[\![\mu]\!] \subseteq \mathcal{L}_I(G_\Omega)$
- $\texttt{Predict}_\Omega(\mu) = Pred\_F$ if
  - $[\![\mu]\!] \subseteq \mathcal{L}_I(G_\Omega) \cup \mathcal{L}_F(G_\Omega)$, and
  - $[\![\mu]\!] \cap \mathcal{L}_I(G_\Omega) \neq \emptyset$, and $[\![\mu]\!] \cap \mathcal{L}_F(G_\Omega) \neq \emptyset$
- $\texttt{Predict}_\Omega(\mu) = No$ if $[\![\mu]\!] \subseteq \mathcal{L}_N(G_\Omega)$
- $\texttt{Predict}_\Omega(\mu) = Pred\_N$ if
  - $[\![\mu]\!] \subseteq \mathcal{L}_N(G_\Omega) \cup \mathcal{L}_I(G_\Omega)$, and
  - $[\![\mu]\!] \cap \mathcal{L}_N(G_\Omega) \neq \emptyset$, and $[\![\mu]\!] \cap \mathcal{L}_I(G_\Omega) \neq \emptyset$

Yes means that all the trajectories in $[\![\mu]\!]$ lie in $\mathcal{L}_{F_\Omega}(\Omega)$. That is, the observation μ only corresponds to trajectories recognized by the pattern. *Pred* means trajectories compatible with μ are not recognized by the pattern so far, but all their extensions will inevitably do so after a bounded number of observations. *No* simply means that all trajectories compatible with μ can still be extended without being recognized by the pattern. *Pred_F* means that the recognition of the pattern is inevitable but some trajectories compatible with the observation are already recognized by the pattern. *Pred_N* means that the observed trace corresponds to some trajectories where recognition of the pattern is inevitable, but others where this is false.

Once again, we use G_1 and G_2 of Examples 1 and 2 to illustrate these results. The predictors (as well as their verdicts) obtained by determinization of $G_{1\Omega}$ and $G_{2\Omega}$ for G_1 and G_2, respectively, with the pattern Ω are given in Fig. 8.

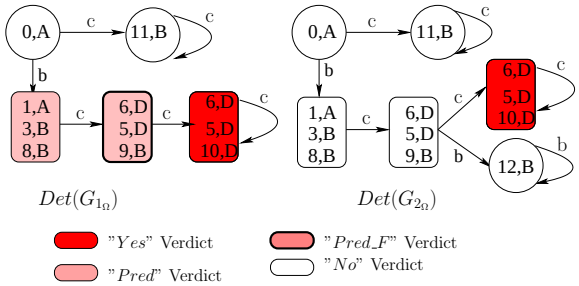

Fig. 8. The predictors

We have the following results about predictors.

Proposition 3. If G is Ω-predictable, then ∃n such that

(i) $\texttt{Predict}_\Omega(\mu) = Pred \Rightarrow (\forall \mu' \in P(\mathcal{L}(G))/\mu, \|\mu'\| \geq n \Rightarrow \texttt{Predict}_\Omega(\mu\mu') = Yes)$

(ii) $\texttt{Predict}_\Omega(\mu) = Yes \Rightarrow \exists \mu' < \mu$ *such that* $\texttt{Predict}_\Omega(\mu') = Pred$ *and* $\|\mu\| - \|\mu'\| \leq n$

(iii) $\forall \mu \in P(\mathcal{L}(G)),\ \texttt{Predict}_\Omega(\mu) \neq ?$ ◇

Part (i) of Proposition 3 simply says that when the predictor issues the decision *Pred*, after at most n observations,

the predictor will issue the decision Yes. Part (ii) emphasizes the fact that if the P_diagnoser issues the decision Yes, then in the past the predictor already issued the decision $Pred$. Part (iii) shows that the decision "?" cannot be issued when the system is Ω-predictable. Indeed, if this decision were to be issued, it would mean that there is a trajectory compatible with μ that is recognized by the pattern without having being detected, which contradicts the fact that the system is predictable.

Figure 9 explains the possible evolutions of the decisions issued by the predictor whenever the system is predictable.

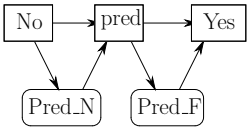

Fig. 9. Possible evolutions of the decisions of the predictor

7. CONCLUSION

We presented new results on predictability in DES. We defined the new notion of predictability of occurrences of patterns modeled as sequences of events in an LTS. We presented an off-line algorithm to verify the predictability of a pattern and an on-line algorithm to analyze the occurrences of the pattern during the operation of the system. We showed that the off-line algorithm is polynomial in the number of states of the system. This improves upon the algorithms proposed in earlier studies on predictability of single events. While this has not been done yet, our predictability verification could be implemented within the software environment DESUMA Ricker et al. [2006]. The on-line algorithm builds upon the off-line one to construct what is termed a predictor automaton; the prediction decision are obtained from an appropriate partition of the state space of the predictor.

In this study, we restricted attention to systems and patterns that can be expressed by regular languages. The definition of predictability of patterns can easilly be extended to include systems and patterns expressed by non-regular languages. However, the development of verification and on-line prediction algorithms for systems or patterns that are not modeled by regular languages remains an open problem.

REFERENCES

S. R. Buss, C. Papadimitriou, and J. Tsitsiklis. On the predictability of coupled automata: an allegory about chaos. *Complex Systems*, 5:525–539, 1991.

X.-R. Cao. The predictability of discrete event systems. *IEEE Trans. Automatic Control*, 34(11):1168–1171, November 1989.

C. Chase and P. J. Ramadge. Predictability of a class of one-dimensional supervised systems. In *Proceedings of the Fifth IEEE International Symposium on Intelligent Control*, September 1990.

S. K. Das, F. Sarkar, K. Basu, and S. Madhavapeddy. Parallel discrete event simulation in star networks with application to telecommunications. In *MASCOTS '95: Proceedings of the 3rd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 66–71, Washington, DC, USA, 1995. IEEE Computer Society. ISBN 0-8186-6902-0.

P. Declerck. Predictability and control synthesis in time deviant graphs. In *Workshop on Discrete-Event Systems*, Cagliari, Italy, August 1998.

H.K. Fadel and L.E. Holloway. Using SPC and template monitoring method for fault detection and prediction in discrete event manufacturing systems. In *Proceedings of the 1999 IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics*, pages 150 – 155, September 1999.

P. H. Feiler, B. Lewis, and S. Vestal. Improving predictability in embedded real-time systems, December 2000.

S. Genc and S. Lafortune. Predictability in discrete-event systems under partial observation. In *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Beijing, China, August 2006a.

S. Genc and S. Lafortune. Diagnosis of patterns in partially-observed discrete-event systems. In *45th IEEE Conference on Decision and Control*, San Diego, CA, December 2006b.

Q. He, M. Ammar, G. Riley, and R. Fujimoto. Exploiting the predictability of tcp's steady-state behavior to speed up network simulation. In *10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems*, pages 101 –108, 2002.

T. Jéron, H. Marchand, S. Genc, and S. Lafortune. Predictability of sequence patterns in discrete event systems. Technical Report 1834, Irisa, March 2007.

S. Jiang and R. Kumar. Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. *IEEE Transactions on Automatic Control*, 49(6): 934–945, 2004.

S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46 (8):1318–1321, 2001.

T. Jéron, H. Marchand, S. Pinchinat, and M-O. Cordier. Supervision patterns in discrete event systems diagnosis. In *Workshop on Discrete Event Systems, WODES'06*, Ann-Arbor (MI, USA), July 2006.

M. Musolesi and C. Mascolo. Evaluating context information predictability for autonomic communication. In *American Control Conference*, Minneapolis, Minnesota USA, June 2006.

L. Ricker, S. Lafortune, and S. Genc. DESUMA: A tool integrating GIDDES and UMDES. In *Software Tools, 8th International Workshop on Discrete-Event Systems*, July 2006.

T. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially-observed discrete-event systems. *IEEE Trans. on Automatic Control*, 47(9):1491–1495, Septembre 2002.