Supervision Patterns in Discrete Event Systems Diagnosis

Thierry Jéron, Hervé Marchand, Sophie Pinchinat, Marie-Odile Cordier

Proceedings of the 8th International
Workshop on Discrete Event Systems
Ann Arbor, Michigan, USA, July 10-12, 2006

Supervision Patterns in Discrete Event Systems Diagnosis

Thierry Jéron, Hervé Marchand, Sophie Pinchinat, Marie-Odile Cordier
IRISA, Campus Universitaire de Beaulieu, 35042 Rennes, France
{firstName.Name}@irisa.fr

Abstract— **In this paper, we are interested in the diagnosis of discrete event systems modeled by finite transition systems. We propose a model of supervision patterns general enough to capture past occurrences of particular trajectories of the system. Modeling the diagnosis objective by a supervision pattern allows us to generalize the properties to be diagnosed and to render them independent of the description of the system. We first formally define the diagnosis problem in this context. We then derive techniques for the construction of a diagnoser and for the verification of the diagnosticability based on standard operations on transition systems. We show that these techniques are general enough to express and solve in a unified way a broad class of diagnosis problems found in the literature, e.g. diagnosing permanent faults, multiple faults, fault sequences and some problems of intermittent faults.**

I. Introduction

Diagnosing and monitoring dynamical systems is an increasingly active research domain and model-based approaches have been proposed which differ according to the kind of models they use [13], [1], [11], [10], [3], [4]. The general diagnosis problem consists in detecting or identifying patterns of particular events on a partially observable system. This paper focuses on discrete-event systems modeled as finite state machines. In this context, patterns usually describe the occurrence of a fault [12], [13], multiple occurrences of a fault [7], the repair of a system after the occurrence of a fault [2]. The aim of diagnosis is to decide, by means of a *diagnoser*, whether or not such a pattern occurred in the system. Even if such a decision cannot be taken immediately after the occurrence of the pattern, one requires that this decision has to be taken in a bounded delay. This property is usually called *diagnosability*. This property can be checked *a priori* from the system model, and depends on both its observability and the supervision pattern.

However, the approaches in the literature suffer from some deficiencies. One observes many different notions of diagnosability and *ad hoc* algorithms for the construction of the diagnoser, and for the verification of diagnosability. As a consequence, these results are difficult to reuse for new but similar diagnosis problems. We believe that the main reason is the absence of a clear definition of the involved patterns, which would clarify the separation between the diagnosis objective and the specification of the system.

In this paper, we formally introduce the notion of supervision pattern as a means to define the diagnosis objectives: a supervision pattern is an automaton which language is the set of trajectories one wants to diagnose. The proposal is general enough to cover an important class of diagnosis objectives, including detection of permanent faults, but also transient faults, multiple faults, repeating faults, as well as quite complex sequences of events.

We then propose a formal definition of the Diagnosis Problem in this context. The essential point is a clear definition of the set of trajectories *compatible* with an observed trace. Now, the Diagnosis Problem is expressed as the problem of synthesizing a function over traces, the *diagnoser*, which decrees on the possible/certain occurrence of the pattern on trajectories compatible with the trace. The diagnoser is required to fulfil two fundamental properties: *Correctness* and *Bounded Diagnosability*. *Correctness* expresses that the diagnoser answers accurately and *Bounded Diagnosability* guarantees that only a bounded number of observations is needed to eventually answer with certainty that the pattern has occurred. *Bounded Diagnosability* is formally defined as the Ω-diagnosability of the system (where Ω is the supervision pattern), which compares to standard diagnosability by [13]. Relying on the formal framework we have developed, we then propose algorithms for both the synthesis of a diagnoser, and the verification of Ω-diagnosability. We believe that these generic algorithms as well as their correctness proofs are a lot more simple than the ones proposed in the literature.

The paper is organized as follows. In section II, we recall standard definitions and notations on labeled transition systems, as well as the notion a compatible trajectories of an observable trace. Supervision patterns are introduced in section III. The diagnosis problem and the Ω-diagnosability are then defined. Section IV is dedicated to algorithms and their associated proofs, for the construction of a correct diagnoser as well as the verification of Ω-diagnosability. Finally, Section V illustrates the approach with an example.

II. Labelled Transition Systems and Related Notions

We first recall useful standard notations: We assume given an alphabet Σ, that is a finite set $\{\sigma_0, \sigma_1, \dots\}$. The set of

1-4244-0053-8/06/$20.00 ©2006 IEEE

finite sequences over Σ is denoted by Σ^*, with ϵ for the empty sequence. In the paper, typical elements of Σ^* are $s, t, u, \ldots$ For $s = \sigma_1 \ldots \sigma_n$ and $t = \sigma'_1 \ldots \sigma'_m$ in Σ^*, $s.t = \sigma_1 \ldots \sigma_n \sigma \sigma'_1 \ldots \sigma'_m$ denotes the *concatenation of s and t* . The *length* of $s \in \Sigma^*$ is denoted $\|s\|$.

We now come to the models of systems:

Definition 1 (LTS): An LTS over Σ is defined by a 4-tuple $M = (Q, \Sigma, \rightarrow, q_0)$ where Q is a finite set of states, $q_0 \in Q$ is the initial state, *Σ is the alphabet of* events, *and $\rightarrow \subseteq Q \times \Sigma \times Q$ is* the transition relation.

In the rest of the section, we assume given an LTS $M = (Q, \Sigma, \rightarrow, q_0)$. We write $q \xrightarrow{\sigma} q'$ for $(q, \sigma, q') \in \rightarrow$. Let $q \xrightarrow{s}$ mean that $q \xrightarrow{s} q'$ for some $q' \in Q$. The *event set* of a state $q \in Q$ is $\Sigma(q) \triangleq \{\sigma \in \Sigma \mid q \xrightarrow{\sigma}\}$. We extend $\rightarrow$ to arbitrary sequences by setting : $q \xrightarrow{\epsilon} q$ always holds, and $q \xrightarrow{s\sigma} q'$ whenever $q \xrightarrow{s} q''$ and $q'' \xrightarrow{\sigma} q'$, for some $q'' \in Q$.

A state q is *reachable* if $\exists s \in \Sigma^*, q_0 \xrightarrow{s} q$.

We set $\Delta_M(q, s) \triangleq \{q' \in Q \mid q \xrightarrow{s} q'\}$. In particular $\Delta_M(q, \epsilon) \triangleq \{q\}$. By abuse of notation, for any language $L \subseteq \Sigma^*$ $\Delta_M(q, L) \triangleq \{q' \in Q \mid q \xrightarrow{s} q'$ for some $s \in L\}$, and for any $Q' \subseteq Q$, $\Delta_M(Q', L) = \bigcup_{q \in Q'} \Delta_M(q, L)$.

We say that M is *deterministic* if whenever $q \xrightarrow{\sigma} q'$ and $q \xrightarrow{\sigma} q''$, then $q' = q''$, for each $q \in Q$ and each $\sigma \in \Sigma$.

A subset $Q' \subseteq Q$ is *stable* whenever $\Delta_M(Q', \Sigma) \subseteq Q'$.

M is *alive* if $\Sigma(q) \neq \emptyset$, for each $q \in Q$. It is *complete* whenever $\Sigma(q) = \Sigma$, for each $q \in Q$.

The *language generated* by the system M is the set $\mathcal{L}(M) \triangleq \{s \in \Sigma^* \mid q_o \xrightarrow{s}\}$ which elements are called *trajectories* of M. Given a trajectory $s \in \mathcal{L}(M)$, we write

$$\mathcal{L}(M)/s \triangleq \{t \in \Sigma^* \mid s.t \in \mathcal{L}(M)\}$$

for the set of trajectories that extend s in M.

Rapidly in the paper, we will need to distinguish a subset $Q_m \subseteq Q$ to denote final states. The notions above are extended in this setting by letting $\mathcal{L}_{Q_m}(M) = \{\sigma \in \Sigma^* \mid \Delta_M(q_0, \sigma) \subseteq Q_m\}$.

A useful operation on LTS is the synchronous product that allows to intersect languages of two LTSs.

Definition 2: Let $M^i = (Q^i, q_0^i, \Sigma_i, \rightarrow_i), i = 1, 2$, be two LTSs. Their synchronous product *is $M^1 \times M^2 = (Q^1 \times Q^2, (q_0^1, q_0^2), \Sigma_1 \cup \Sigma_2, \rightarrow)$, where $\rightarrow \subseteq Q^1 \times Q^2$ satisfies $(q^1, q^2) \xrightarrow{\sigma} (q'^1, q'^2)$ whenever $q^1 \xrightarrow{\sigma}_1 q'^1$ and $q^2 \xrightarrow{\sigma}_2 q'^2$.*

Clearly $\mathcal{L}(M^1 \times M^2) = \mathcal{L}(M^1) \cap \mathcal{L}(M^2)$ and for $Q_1 \subseteq Q^1$ and $Q_2 \subseteq Q^2$, we also have $\mathcal{L}_{Q_1 \times Q_2}(M^1 \times M^2) = \mathcal{L}_{Q_1}(M^1) \cap \mathcal{L}_{Q_2}(M^2)$. Moreover, if two sets $Q_1 \subseteq Q^1$ and $Q_2 \subseteq Q^2$ are stable, $Q_1 \times Q_2$ is stable in $M^1 \times M^2$.

As we are interested in diagnosing systems - this will be formalized in the next section -, partial observation plays a central rôle. In this regard, the set of events Σ is partitioned into Σ_o and Σ_{uo} (i.e. $\Sigma = \Sigma_o \cup \Sigma_{uo}$, and $\Sigma_o \cap \Sigma_{uo} = \emptyset$), where Σ_o represents the set of *observable* events - elements of Σ_{uo} are then *unobservable* events. Typical elements of Σ_o^* will be denoted by μ, μ'.

We say that M is *Σ_o-alive* if $\forall q \in Q, \exists s \in \Sigma^*.\Sigma_o, q \xrightarrow{s}$, meaning that there is no terminal loop of unobservable events. Notice that when M has no loop of unobservable events, M is alive if and only if M is Σ_o-alive.

Let $P : \Sigma^* \rightarrow \Sigma_o^*$ be the natural *projection* of trajectories onto Σ_o^* defined by: $P(\epsilon) = \epsilon$ and $P(s\sigma) = P(s).\sigma$ if $\sigma \in \Sigma_o$, and $P(s)$ otherwise. The projection P simply erases the unobservable events from a trajectory. P extends to languages by defining, for $L \subseteq \Sigma^*$, $P(L) = \{P(s) \mid s \in L\}$. The inverse projection of L is defined by $P^{-1}(L) = \{s \in \Sigma^* \mid P(s) \in L\}$.

Now, the *language of traces of M* is

$$Traces(M) \triangleq P(\mathcal{L}(M))$$

It is the set of observable sequences of its trajectories.

From the projection P, we derive an equivalence relation between trajectories of M, written $\equiv_M$, called the *Delay-Observation equivalence* in reference to the delay-bisimulation of [9]:

Definition 3 (Delay-Observation Equivalence, $\equiv_M$): Let $\equiv_M \subseteq \mathcal{L}(M) \times \mathcal{L}(M)$ be the binary relation defined by $s \equiv_M s'$ whenever

- *$P(s) = P(s')$ and*
- *$s \in \Sigma^*.\Sigma_o$ if and only if $s' \in \Sigma^*.\Sigma_o$.*

One easily verifies that $\equiv_M$ is an equivalence relation, and we take the convention to write $[s]$ for the equivalence class of s.

Given $s \in \mathcal{L}(M)$, s naturally maps onto a trace of M, namely $P(s)$. Now, given a non empty trace μ of M, μ does not uniquely determine a Delay-Observation equivalence class as in general μ can be brought back in M in two different manners: μ can be associated with the class $[s]$ with $P(s) = \mu$ and $s \in \Sigma^*\Sigma_o$, or μ can be associated with the class $[s']$ with $P(s') = \mu$ and $s' \in \Sigma^*\Sigma_{uo}$ (notice that by Definition 3, $[s]$ and $[s']$ are different). Henceforth, we take the convention that the equivalence class denoted by a trace μ is

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

We say that $[\![\mu]\!]_M$ is the set of trajectories *compatible* with the trace μ. When clear from the context, we will use $[\![\mu]\!]$ for $[\![\mu]\!]_M$. This notion of compatible trajectory will be a central notion for diagnosis as the aim will be to infer properties on the set of trajectories $[\![\mu]\!]_M$ compatible with the observation of the trace μ. The reason for choosing this definition of $[\![\mu]\!]$ is that in the case of online diagnosis, it is natural to assume that the diagnoser is reactive to an observable move of the system.

III. Supervision Patterns and the Diagnosis Problem

In this section, we introduce the notion of *supervision patterns*, which are means to define languages we are interested in for diagnosis purpose. We then give some examples of such patterns. Finally, we introduce the diagnosis problem for such patterns.

A. Supervision Patterns

Supervision patterns are represented by particular LTSs:

Definition 4: A supervision pattern *is a 5-tuple* $\Omega = (Q_\Omega, \Sigma, \rightarrow_\Omega, q_{0_\Omega}, Q_F)$*, where* $(Q_\Omega, \Sigma, \rightarrow_\Omega, q_{0_\Omega})$ *is a deterministic and complete LTS, and* $Q_F \subseteq Q$ *is a distinguished stable subset of states.*

As Ω is complete we get $\mathcal{L}(\Omega) = \Sigma^*$. Also notice that the assumption that Q_F is stable means that its accepted language is "extension-closed", i.e. satisfies $\mathcal{L}_{Q_F}(\Omega).\Sigma^* = \mathcal{L}_{Q_F}(\Omega)$. Otherwise said, $\mathcal{L}_{Q_F}(\Omega)$ is a language violating a safety property. This choice is natural since we want to diagnose whether all trajectories compatible with an observed trace have a prefix recognized by the pattern.

In the next subsection we give some examples of supervision patterns which rephrase standard properties of interest for diagnosis.

B. Examples of supervision patterns

a) occurrence of one fault: Le $f \in \Sigma$ be a fault and consider that we are interested in diagnosing the occurrence of this fault. A trajectory $s \in \Sigma^*$ is faulty if $s \in \Sigma^*.f.\Sigma^*$. The supervision pattern Ω_f of Figure 1 exactly recognizes this language, $\mathcal{L}(\Omega_f) = \Sigma^*.f.\Sigma^*$.

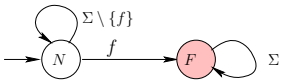


Fig. 1. Supervision pattern for one fault

b) occurrence of multiple faults: Let f_1 and f_2 be two faults that may occur in the system. Diagnosing the occurrence of these two faults in an trajectory means deciding the membership of this trajectory in $\Sigma^*.f_1.\Sigma^* \cap \Sigma^*.f_2.\Sigma^* = \mathcal{L}_{F_1}(\Omega_{f_1}) \cap \mathcal{L}_{F_2}(\Omega_{f_2})$, where $\Omega_{f_i}, i \in \{1, 2\}$ are isomorphic to the supervision pattern Ω_f described in Figure 1. The supervision pattern is then the product $\Omega_{f_1} \times \Omega_{f_2}$ which accepted language in $F_1 \times F_2$ is $\mathcal{L}_{F_1 \times F_2}(\Omega_{f_1} \times \Omega_{f_2}) = \mathcal{L}_{F_1}(\Omega_{f_1}) \cap \mathcal{L}_{F_2}(\Omega_{f_2})$.

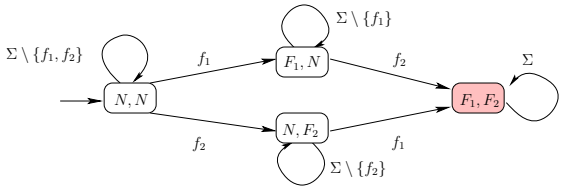


Fig. 2. Supervision pattern for two faults

More generally, the supervision pattern for the occurrence of a set of faults $\{f_1, \cdots, f_l\}$ is the product $\times_{i=1,\ldots,l}\Omega_{f_i}$, considering $\times_{i=1,\ldots,l}F_i$ as final state set.

c) ordered occurence of events: If one is interested in diagnosing different faults in a precise order, for example, f_2 after f_1, the supervision pattern should recognize the trajactories in $\Sigma^*.f_1.\Sigma^*.f_2.\Sigma^*$, which corresponds to the concatenation of the two languages $\mathcal{L}_{F_1}(\Omega_{f_1}).\mathcal{L}_F(\Omega_{f_2})$ as described by the supervision pattern given in Figure 3.

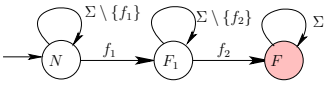


Fig. 3. Ordered occurrence of events

If f_1 corresponds to a fault event and f_2 to the reparation of this fault in the system, then we actually diagnose the reparation of the fault f_1. With this pattern, the aim is to match the *I-diagnosability* in [2].

d) multiple occurrences of the same fault: Another interesting problem is to diagnose the multiple occurrences of the same fault event f, say k times. The supervision pattern is given in Figure 4 which accepted language is $\mathcal{L}_F(\Omega_f)^k$. The aim is to match the k-diagnosability of [7].

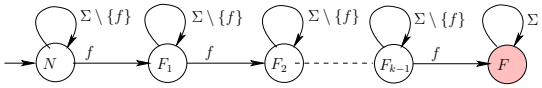


Fig. 4. k occurrences of the same fault f

e) Intermittent Fault: The supervision pattern given in Figure 5 describes the fact that a fault (occurrence of f) occurred twice without repair (occurrence of r).

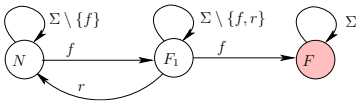


Fig. 5. Intermittent fault with repair

It is worthwhile noting that this can be generalized to a pattern recognizing the occurrence of k faults (identical or not) without repair.

C. The Diagnosis Problem

In the remainder of the paper, we consider a system whose behavior is modeled by an LTS $G = (Q, \Sigma, \rightarrow, q_0)$. The only assumption made on G is that G is Σ_o-alive. Notice that G can be non-deterministic. We also consider a supervision pattern $\Omega = (Q_\Omega, \Sigma, \rightarrow_\Omega, q_{0_\Omega}, Q_F)$ denoting the language $\mathcal{L}_{Q_F}(\Omega)$ that we want to diagnose.

We define the *Diagnosis Problem* as the problem of defining a function $\texttt{Diag}_\Omega$ on traces, whose intention is to answer the question whether trajectories compatible with observed traces are recognized or not by the supervision pattern. We do require some properties for $\texttt{Diag}_\Omega$: *Correctness* means that "Yes" and "No" answers should be accurate, while *Bounded Diagnosability* means that trajectories in $\mathcal{L}_{Q_F}(\Omega)$ should be diagnosed with finitely many observations.

The *Diagnosis problem* can be stated as follows: given an LTS G and a supervisory pattern Ω, decide whether there exists (and compute if any) a three valued function $\texttt{Diag}_\Omega : Traces(G) \rightarrow \{\text{"YES"}, \text{"NO"}, \text{"?"}\}$ decreeing, for

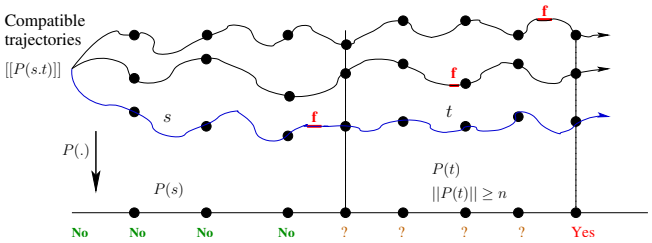
each trace μ of G, on the membership in $\mathcal{L}_{Q_F}(\Omega)$ of any trajectory in $[\![\mu]\!]$. Formally,

- (Diagnosis Correctness) The function should verify
$$\texttt{Diag}_\Omega(\mu) = \begin{cases} \text{“YES”} & \text{if } [\![\mu]\!] \subseteq \mathcal{L}_{Q_F}(\Omega) \\ \text{“NO”} & \text{if } [\![\mu]\!] \cap \mathcal{L}_{Q_F}(\Omega) = \emptyset \\ \text{“?”} & \text{otherwise.} \end{cases}$$
- (Bounded Diagnosability) As G is only partially observed, we expect in general situations where $\texttt{Diag}_\Omega(\mu) =$ “?” (as neither $[\![\mu]\!] \subseteq \mathcal{L}_{Q_F}(\Omega)$ nor $[\![\mu]\!] \cap \mathcal{L}_{Q_F}(\Omega) = \emptyset$ hold). However, we require this undetermined situation not to last in the following sense: There must exist $n \in \mathbb{N}$, the bound, such that whenever $s \in [\![\mu]\!] \cap \mathcal{L}_{Q_F}(\Omega)$, for all $t \in \mathcal{L}(G)/s \cap \Sigma^*.\Sigma_o$, if $\|P(t)\| \geq n$ then $\texttt{Diag}_\Omega(P(s.t)) =$ “YES”.

Diagnosis Correctness means that the diagnosis of a trace μ is “Yes” if all trajectories in $[\![\mu]\!]$ lie in $\mathcal{L}_{Q_F}(\Omega)$, while it is “No” if no trajectory in $[\![\mu]\!]$ lies in $\mathcal{L}_{Q_F}(\Omega)$.

Bounded Diagnosability means that when observing a trajectory in $\mathcal{L}_{Q_F}(\Omega)$, a “Yes” answer should be produced after finitely many observable events.

The following figure gives an intuitive explanation of these notions when $\Omega = \Omega_f$.



Now, if $\texttt{Diag}_\Omega$ provides a Correct Diagnosis, Bounded Diagnosability can be rephrased, by replacing $\texttt{Diag}_\Omega(P(s.t)) =$ “YES” with $[\![P(s.t)]\!] \subseteq \mathcal{L}_{Q_F}(\Omega)$. We obtain what we call the Ω-*diagnosability*. Notice that this is now a property of G with respect to Ω.

Definition 5: An LTS G is $\Omega(n)$-diagnosable, *where* $n \in \mathbb{N}$, *whenever*

$$\forall s \in \mathcal{L}_{Q_F}(\Omega) \cap \mathcal{L}(G) \cap \Sigma^*.\Sigma_o, \forall t \in \mathcal{L}(G)/s \cap \Sigma^*.\Sigma_o,$$
$$\text{if } \|P(t)\| \geq n \text{ then } [\![P(s.t)]\!] \subseteq \mathcal{L}_{Q_F}(\Omega).$$

G is said Ω-diagnosable *if it is* $\Omega(n)$-diagnosable *for some* $n \in \mathbb{N}$.

Ω-diagnosability says that when a trajectory s ending with an observable event is recognized by the supervision 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.

The remark before Definition 5 is formalized by :

Proposition 1: If $\texttt{Diag}_\Omega$ *computes a Correct Diagnosis, then G is Ω-diagnosable if and only if the Bounded Diagnosability Property holds for* $\texttt{Diag}_\Omega$.

As to show the unifying framework based on supervision patterns, we here consider the very particular supervision pattern Ω_f of Section III-B, originally considered by [12], [13] with the associated notion of f-diagnosability. Let us first recall this notion.

Let G be an LTS which is alive and has no loop of unobservable event. G is *f-diagnosable* whenever

$$\exists N \in \mathbb{N}, \forall s \in \Sigma^*.f, \forall t \in \mathcal{L}(G)/s, \text{ if } \|t\| \geq N, \\ \text{then } \forall u \in \mathcal{L}(G), P(u) = P(s.t) \Rightarrow u \in \Sigma^*.f.\Sigma^* \quad (1)$$

The following proposition relates f-diagnosability with Ω_f-diagnosability:

Proposition 2: Let G be an LTS and assume that G is alive and has no loop of internal events. Then G is f-diagnosable if and only if G is Ω_f-diagnosable.

Proof We first make the following remarks:

(a) $u \in \Sigma^*.f.\Sigma^*$ is equivalent to $u \in \mathcal{L}_{Q_F}(\Omega_f)$;
(b) $s \in \Sigma^*.f$ implies $s \in \mathcal{L}_{Q_F}(\Omega_f)$;

Assume G is f-diagnosable, and that $N \in \mathbb{N}$ fulfills (1). We prove that G is $\Omega_f(N)$-diagnosable: consider $s \in \mathcal{L}_{Q_F}(\Omega_f) \cap \Sigma^*\Sigma_o$ and let $t \in \mathcal{L}(G)/s \cap \Sigma^*.\Sigma_o$ with $\|P(t)\| \geq N$; note that therefore $\|t\| \geq N$. It is easy to show that s decomposes into $s = s'.s''$ where $s' \in \Sigma^*.f$, with additionally $s''.t \in \mathcal{L}(G)/s'$. Now, $\|t\| \geq N$ implies $\|s''.t\| \geq N$, which, by (1), entails that for any $u \in \mathcal{L}(G)$ with $P(u) = P(s.t)$, we have $u \in \Sigma^*.f.\Sigma^* = \mathcal{L}_{Q_F}(\Omega_f)$. This implies in particular that $[\![P(s.t)]\!] \subseteq \mathcal{L}_{Q_F}(\Omega_f)$.

Reciprocally, assume G is $\Omega_f(n)$-diagnosable, for some n. Let m be the length of the longest unobservable trajectory in G (which exists by assumption), and consider $N = (n+1) * m$. Consider $s \in \Sigma^*.f$ and $t \in \mathcal{L}(G)/s$ with $\|t\| \geq N$ (thus $\|P(t)\| \geq n+1$). We have to prove that for any $u \in \mathcal{L}(G)$ with $P(u) = P(s.t)$, we have $u \in \Sigma^*.f.\Sigma^*$. Let $t = t_1.t_2.t_3$ with $t_1 \in \Sigma_{uo}^*.\Sigma_o$, $t_2 \in \Sigma^*.\Sigma_o$, and $t_3 \in \Sigma_{uo}^*$. Let $s' = s.t_1$. As Q_F is stable and $s \in \mathcal{L}_{Q_F}(\Omega_f)$, $s' \in \mathcal{L}_{Q_F}(\Omega_f) \cap \Sigma^*.\Sigma_o$. We have $t_2 \in \mathcal{L}(G)/s' \cap \Sigma^*.\Sigma_o$ with $\|P(t_2)\| \geq n$. By $\Omega_f(n)$-diagnosability, for all $u \in \mathcal{L}(G)$ with $P(u) = P(s.t)$, we have $u \in \Sigma^*.f.\Sigma^* = \mathcal{L}_{Q_F}(\Omega_f)$. $\diamond$

IV. Algorithms for the Diagnosis Problem

We now propose algorithms for the Diagnosis Problem based on standard operations on LTSs. In a first stage we base the construction of the $\texttt{Diag}_\Omega$ function on the synchronous product of G and Ω and its determinisation, and prove that the function $\texttt{Diag}_\Omega$ computes a Correct Diagnosis. Next, we propose an algorithm allowing to check for the Ω-diagnosability of an LTS, thus ensuring the Bounded Diagnosis Property of the function $\texttt{Diag}_\Omega$. Hence achieving the decision of the Diagnosis Problem.

A. Computing a candidate for the function $\texttt{Diag}_\Omega$

We propose a computation of the function $\texttt{Diag}_\Omega$: given an LTS G and a supervision pattern Ω, we first consider the synchronous product G_Ω of G and Ω (see Definition 2). Next we perform on G_Ω a second operation (see Definition 6) which associates to G_Ω a deterministic LTS written $Det(G_\Omega)$. We then show how $Det(G_\Omega)$ provides a function $\texttt{Diag}_\Omega$ delivering a Correct Diagnosis.

Let us first introduce a determinisation function.

Definition 6: Let $M = (Q, \Sigma, \rightarrow, q_0)$ *be an LTS with* $\Sigma = \Sigma_{uo} \cup \Sigma_o$. *The* determinisation of M *is the LTS* $Det(M) = (\mathcal{X}, \Sigma_o, \rightarrow_d, X_0)$ *where* $\mathcal{X} = 2^Q$ *(the set of subsets of* Q *called* macro-states*),* $X_0 = \{q_0\}$ *and* $\rightarrow_d = \{(X, \sigma, \Delta_M(X, \Sigma^*_{uo}.\sigma) \mid X \in \mathcal{X} \text{ and } \sigma \in \Sigma_o\}$.

Notice that for this definition the target macro-state X' of a transition $X \stackrel{\sigma}{\rightarrow}_d X'$ is only composed of states q' of M which are targets of sequences of transitions $q \stackrel{s.\sigma}{\rightarrow} q'$ ending with an observable event σ. The reason for this definition is the coherency with $[\![.]\!]$. In fact, from the definition of $\rightarrow_d$ in $Det(M)$, we infer that $\Delta_{Det(M)}(X_0, \mu) = \{\Delta_M(q_0, [\![\mu]\!])\}$, 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_0 by trajectories of $[\![\mu]\!]$ in M.

Finally, determinisation preserves traces, so we have $\mathcal{L}(Det(M)) = \mathit{Traces}(Det(M)) = \mathit{Traces}(M)$.

We now explain the construction of the diagnoser from G and Ω. Let us first consider the synchronous product $G_\Omega = G \times \Omega$ (see Definition 2). We then get $\mathcal{L}(G_\Omega) = \mathcal{L}(G) \cap \mathcal{L}(\Omega) = \mathcal{L}(G)$ as Ω is complete (thus $\mathcal{L}(\Omega) = \Sigma^*$). We also get $\mathcal{L}_{Q \times Q_F}(G_\Omega) = \mathcal{L}(G) \cap \mathcal{L}_{Q_F}(\Omega)$ meaning that the trajectories of G accepted by Ω are exactly the accepted trajectories of G_Ω. Finally note that $Q \times Q_F$ is stable in G_Ω as both Q and Q_F are stable by assumption.

We now apply determinisation to G_Ω. We have $\mathit{Traces}(Det(G_\Omega)) = \mathit{Traces}(G_\Omega) = Traces(G)$ thus for all $\mu \in Traces(G)$, $\Delta_{Det(G_\Omega)}(X_0, \mu) = \{\Delta_{G_\Omega}(q_0, [\![\mu]\!]\}$.

We now establish the following fundamental results on the construction $Det(G_\Omega)$:

Proposition 3: For any $\mu \in Traces(G) = Traces(G_\Omega)$,

$$\Delta_{Det(G_\Omega)}(X_0, \mu) \subseteq Q \times Q_F \iff [\![\mu]\!] \subseteq \mathcal{L}_{Q_F}(\Omega) \quad (2)$$

$$\Delta_{Det(G_\Omega)}(X_0, \mu) \cap Q \times Q_F = \emptyset \iff [\![\mu]\!] \cap \mathcal{L}_{Q_F}(\Omega) = \emptyset \quad (3)$$

(2) means that all trajectories compatible with a trace μ are accepted by Ω if and only μ leads to a macro-state only composed of marked states in G_Ω.

(3) means that all trajectories compatible with μ are not accepted by Ω if and only if μ leads to a macro-state only composed of unmarked states in G_Ω.

Proof The proof of (2) is established by the following sequence of equivalences

$$\begin{array}{ll} \Delta_{Det(G_\Omega)}(X_0, \mu) \subseteq Q \times Q_F & \iff \\ \{\Delta_{G_\Omega}(q_0, [\![\mu]\!]\} \subseteq Q \times Q_F & \iff \\ [\![\mu]\!] \subseteq \mathcal{L}_{Q \times Q_F}(G_\Omega) & \iff \\ [\![\mu]\!] \subseteq \mathcal{L}(G) \cap \mathcal{L}_{Q_F}(\Omega) & \end{array}$$

Similarly, for the proof of (3) we have

$$\begin{array}{ll} \Delta_{Det(G_\Omega)}(X_0, \mu) \cap Q \times Q_F = \emptyset & \iff \\ \{\Delta_{G_\Omega}(q_0, [\![\mu]\!]\} \cap Q \times Q_F = \emptyset & \iff \\ [\![\mu]\!] \cap \mathcal{L}_{Q \times Q_F}(G_\Omega) = \emptyset & \iff \\ [\![\mu]\!] \cap \mathcal{L}(G) \cap \mathcal{L}_{Q_F}(\Omega) = \emptyset & \iff \\ [\![\mu]\!] \cap \mathcal{L}_{Q_F}(\Omega) = \emptyset \text{ as } [\![\mu]\!] \subseteq \mathcal{L}(G) & \end{array}$$

We have now the material to define the function $\mathtt{Diag}_\Omega$ and to obtain the Correctness Diagnosis Property, following directly from Proposition 3.

Theorem 1: Let $Det(G_\Omega)$ *be the LTS built as above, and let* $\mathtt{Diag}_\Omega(\mu)$ *be:*

$$\begin{cases} \text{“YES”}, & \text{if } \Delta_{Det(G_\Omega)}(X_0, \mu) \subseteq Q \times Q_F \\ \text{“NO”}, & \text{if } \Delta_{Det(G_\Omega)}(X_0, \mu) \cap Q \times Q_F = \emptyset \\ \text{“?”}, & \text{otherwise.} \end{cases} \quad (4)$$

$\mathtt{Diag}_\Omega$ *computes a Correct Diagnosis.*

B. Verifying the Bounded Diagnosis Property of $\mathtt{Diag}_\Omega$

As we have established the Correctness of $\mathtt{Diag}_\Omega$, according to Proposition 1, the Bounded Diagnosability Property of $\mathtt{Diag}_\Omega$ is provided by the Ω-diagnosability of G. We now propose an algorithm for deciding Ω-diagnosability (Definition 5).

This algorithm is adapted from [5], [14]. The idea is that G is not Ω-diagnosable if there exists an arbitrarily long trace μ, such that two trajectories compatible with μ disagree on $\mathcal{L}_{Q_F}(\Omega)$ membership. We first introduce the Delay-Observational-Closure $\mathrm{OBS}(G_\Omega)$ that preserves the information about $\mathcal{L}_{Q_F}(\Omega)$ membership while abstracting away unobservable events. Next, a self-product $\mathrm{OBS}(G_\Omega) \times \mathrm{OBS}(G_\Omega)$ allows to extract from a trace μ pairs of trajectories of G_Ω and to check their $\mathcal{L}_{Q_F}(\Omega)$ membership agreement.

Definition 7: For an LTS $M = (Q, \Sigma, \rightarrow, q_0)$, *the* Delay-Observational-Closure of M *is* $\mathit{OBS}(M) = (Q, \Sigma_o, \rightarrow_o, q_0)$ *where* $q \stackrel{\sigma}{\rightarrow}_o q'$ *whenever* $q \stackrel{s\sigma}{\rightarrow} q'$ *in* M *for some* $s \in \Sigma^*_{uo}$ *and* $\sigma \in \Sigma_o$.

By definition, for all $\mu \in Traces(M)$, $q_0 \stackrel{\mu}{\rightarrow}_o q'$ in $\mathrm{OBS}(G_\Omega)$ if and only $\exists s \in [\![\mu]\!]$ s.t. $q \stackrel{s}{\rightarrow} q'$ in M.

Consider now $\mathrm{OBS}(G_\Omega) = (Q', \Sigma_o, \rightarrow_o, q_0)$ and let $\Gamma = \mathrm{OBS}(G_\Omega) \times \mathrm{OBS}(G_\Omega)$ be the LTS $(Q' \times Q', \Sigma_o, \rightarrow_\Gamma, (q_0, q_0)$.

By definition of OBS and synchronous product, if $\mu \in Traces(G)$ and $(q_0, q_0) \stackrel{\mu}{\rightarrow}_\Gamma (q, q')$ there exists $s, s' \in [\![\mu]\!]$ s.t. $q_0 \stackrel{s}{\rightarrow} q$ and $q_0 \stackrel{s}{\rightarrow} q'$ in G_Ω.

We say that $(q, q') \in Q' \times Q'$ is Ω-*determined* whenever $q \in Q \times Q_F \iff q' \in Q \times Q_F$. Otherwise, (q, q') is said *undetermined*.

A path in Γ is called an n-*undetermined path* if it contains $n + 1$ consecutive Ω-undetermined states (thus n events between them). A path in Γ is an *undetermined cycle* if it is a cycle which states are all undetermined.

We now show the relation between $\Omega(n)$-diagnosability and the existence of n-undetermined paths.

Lemma 1: G is $\Omega(n)$-diagnosable if and only if there is no reachable n-undetermined path in Γ.

Proof Suppose there is no reachable n-undetermined path. Let $s \in \mathcal{L}_{Q_F}(\Omega) \cap \Sigma^*.\Sigma_o$ and $t \in \mathcal{L}(G)/s \cap \Sigma^*.\Sigma_o$ with $||P(t)|| \geq n$. We should prove that for all $u \in [\![P(s.t)]\!]$, $u \in \mathcal{L}_{Q_F}(\Omega)$. Let $\mu = P(s.t)$. Any path $(q_0, q_0) \stackrel{\mu}{\rightarrow}_\Gamma (q, q')$ in Γ can be decomposed into a path $(q_0, q_0) \stackrel{\mu_1}{\rightarrow}_\Gamma (r, r') \stackrel{\mu_2}{\rightarrow}_\Gamma (q, q')$ with $\mu_1 = P(s)$ and $\mu_2 = P(t)$. We have $||\mu_2|| = ||P(t)|| \geq n$ thus $(r, r') \stackrel{\mu_2}{\rightarrow}_\Gamma (q, q')$ is a path with at least n events. By hypothesis, one of the states along this path, say

(q_d, q'_d) is determined, and as $s \in \mathcal{L}_{Q_F}(\Omega)$, (q_d, q'_d) is surely in $(Q \times Q_F)^2$. Now as $Q \times Q_F$ is stable, $(q, q') \in (Q \times Q_F)^2$. It is then clear that for all $u \in [\![P(s.t)]\!]$, $u \in \mathcal{L}_{Q_F}(\Omega)$.

Conversely, suppose that there exists an n-undetermined path $p = (r, r') \xrightarrow{\mu_2}_{\Gamma} (q, q')$ in Γ with $||\mu_2|| = n$ (i.e. all states on the path are undetermined). If this path is reachable there is also a path $(q_0, q_0) \xrightarrow{\mu_1}_{\Gamma} (r, r')$. As (r, r') is undetermined, there exists $s, s' \in [\![\mu_1]\!]$ with $s \in \mathcal{L}_{Q_F}(\Omega) \cap \Sigma^*.\Sigma_o$, and $s' \notin \mathcal{L}_{Q_F}(\Omega)$. There also exists $t \in \mathcal{L}(G)/s \cap \Sigma^*\Sigma_o$ with $P(t) = \mu_2$. As all states in path p are undetermined, there exists $t' \in \mathcal{L}(G)/s' \cap \Sigma^*.\Sigma_o$ with $P(t') = \mu_2$, but $s'.t' \notin \mathcal{L}_{Q_F}(\Omega)$. We thus have $s \in \mathcal{L}_{Q_F}(\Omega) \cap \Sigma^*.\Sigma_o$ and $t \in \mathcal{L}(G)/s \cap \Sigma^*.\Sigma_o$ with $||P(t)|| \geq n$, and $s'.t' \in [\![P(s.t)]\!]$ with $s'.'t' \notin \mathcal{L}_{Q_F}(\Omega)$. This proves that G is not $\Omega(n)$-diagnosable.

Theorem 2: G is Ω-diagnosable if and only if there exists n such that Γ contains no reachable n-undetermined path.

Based on theorem 2 and on the fact that Γ is finite state, we conclude that

Corollary 1: G is not Ω-diagnosable if and only if Γ contains a reachable undetermined cycle.

Using Proposition 1 and the construction of Γ, verifying diagnosability amounts to check the existence of reachable undetermined cycles in Γ, retrieving the idea of the algorithm of [14], [5].

By Corollary 1 and Lemma 1, we get

Corollary 2: If G is Ω-diagnosable, then G is $\Omega(n+1)$-diagnosable, and not $\Omega(n)$-diagnosable where n is the length of the longest undetermined path of Γ.

We now summarize the procedure to determine whether G is Ω-diagnosable: We perform a depth first search on Γ which either exhibits an undetermined cycle or ends by having computed the length of the longest undetermined sequence. Obviously, this has linear cost in the size of Γ.

V. Supervision Example

The example we discuss here and given in Figure 6 illustrates the approach presented above. In this example, we simply model the moves of a person in a building composed of an *office* (I), a *library* (B), a *reception* (A) and a *coffee-shop* (C). The doors from one part of the building to another can be taken in only one direction. Transitions t_i model the crossings of the doors. Some doors are secured by access-cards, possibly allowing the observation that a person crosses the door. We consider the supervision pattern Ω (Figure 6) which expresses the fact that going twice to the coffee-shop without going to the library is a behaviour that has to be supervised.

Following the different steps described in the previous sections, the product $G_\Omega = G \times \Omega$ is used to label the states of G with respect to the supervision pattern Ω. The corresponding LTS is described in Figure 7.

Let us first assume that only the access-cards, corresponding to the events t_1, t_2 are activated and thus observable i.e. $\Sigma_o = \{t_1, t_2\}$. Notice that the system then has internal

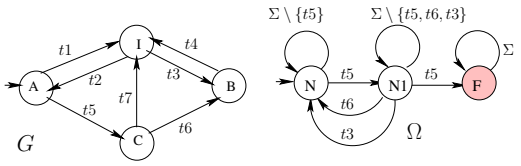


Fig. 6. G and the corresponding supervision pattern Ω

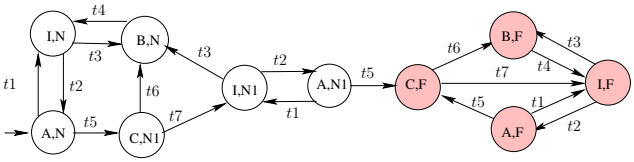


Fig. 7. G_Ω

events loops. The observable system $\text{OBS}(G_\Omega)$ is given in Figure 8.

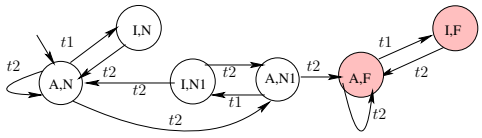


Fig. 8. $\text{OBS}(G_\Omega)$ for $\Sigma_o = \{t_1, t_2\}$

It is easy to check that the LTS $\Gamma = \text{OBS}(G_\Omega) \times \text{OBS}(G_\Omega)$ (not represented here) has an undetermined reachable cycle: ((A,N),(A,N)) $\xrightarrow{t_2}_{\Gamma}$ ((A,N),(A,N1)) $\xrightarrow{t_2}_{\Gamma}$ ((A,N),(A,F)) $\xrightarrow{t_2^*}_{\Gamma}$ ((A,N),(A,F)), thus G is not Ω-diagnosable when the set of observable events is $\Sigma_o = \{t_1, t_2\}$.

However, if the access-card t_4 is activated (i.e. $\Sigma_o = \{t_1, t_2, t_4\}$), the observable system $\text{OBS}(G_\Omega)$ is given by the LTS represented in Figure 9.

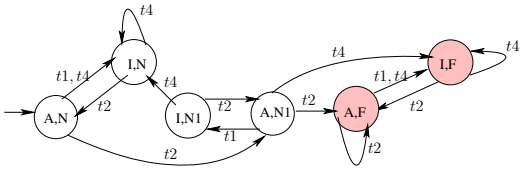


Fig. 9. $\text{OBS}(G_\Omega)$ for $\Sigma_o = \{t_1, t_2, t_4\}$

One can check that $\text{OBS}(G_\Omega)$ is deterministic. Thus $\Gamma = \text{OBS}(G_\Omega) \times \text{OBS}(G_\Omega)$ is isomorphic to $\text{OBS}(G_\Omega)$, and has no undetermined cycle. Consequently, G is Ω-diagnosable for $\Sigma_o = \{t_1, t_2, t_4\}$. Note that we also have that $Det(G_\Omega) = \text{OBS}(G_\Omega)$. Thus $\text{OBS}(G_\Omega)$ actually corresponds to the diagnoser.

VI. Conclusion

The present paper advocates the use of supervision patterns for the description of diagnosis objectives. A supervision pattern is an automaton, like the ones used in many different domains (verification, model-based testing, pattern

matching, etc), in order to unambiguously denote a formal language.

As illustrated in the paper, the fault-occurrence diagnosis is a particular case of pattern diagnosis, but patterns are also useful to describe more general objectives, as shown in subsections III-A and section V. The concept of supervision pattern is even more attractive in the sense that patterns can be composed using usual combinators inherited from language theory (union, intersection, concatenation, etc.).

We are interested in diagnosing the occurrence of trajectories violating a safety property, which by definition can be violated on a finite prefix. It is then natural to assume that patterns recognize "extension-closed" languages, in the sense that if a trajectory of the system belongs to the language, so does any extension of this trajectory. This is technically achieved by the *stability* assumption on the automaton Ω. Adopting the behavioral properties point of view on the patterns leads to the attempt to diagnose any linear time pure past temporal formulas [8]. It is clear that the properties we consider do not meet the LTL definable properties handled by [6].

In the worry of exposing a fairly general framework for diagnosis issues, the Diagnosis Problem is presented in a rather denotational spirit, as opposed to the operational spirit we find in the literature: we put the emphasis on the diagnosis function $\texttt{Diag}_\Omega$ with its correctness and boundedness diagnosability property. Correctness is an essential property that ensures the accuracy of the diagnosis. Moreover, verifying the diagnosability property of the system with respect to the supervision pattern guarantees that when using $\texttt{Diag}_\Omega$ online, an occurrence of the pattern will eventually be diagnosed, and that this eventuality can be quantified. It is the standard notion of "Diagnosability", but seen here as a mere mean to achieve a satisfactory diagnosis function; we are aware that this point of view differs from other classical approaches. The definition of diagnosability as proposed here is automata-based, with G and Ω, but could as well be expressed in a language-based framework.

We now turn to technical aspects of the approach. We have insisted on what the semantics of a trace is: a trace denotes the set of trajectories which project onto this trace and that necessarily end up with an observable event. Consequences of this choice are manifold in the definitions of Ω-diagnosability, $Det(G_\Omega)$ and $\text{OBS}(G)$. We could have chosen another semantics, impacting on the related definitions accordingly: for example we could have considered the set of trajectories which project onto this trace. What is mostly important is the accurate match between the semantics for traces and the other definitions: hence we avoid displeasing discrepancies to determine precisely the Diagnosability Bound, and even better, we have a clear proof for the correctness of the synthesis algorithm. However, we believe our choice is the most natural when admitting that the diagnosis function implemented online as an output verdict is reactive to an observable move of the system.

A more sophisticated diagnosis than the one explained here can be derived from our construction - this is fairly standard: for example, we can take advantage of knowing that the Diagnosability bound is exactly n. Assume that, after a trace μ, the function $\texttt{Diag}_\Omega$ produces "?" on $n+2$ consecutive events, then necessarily the trajectories compatible with μ cannot have met the pattern.

We terminate the discussion with future work perspectives aiming two independent objectives. The first objective is to extend the algorithms to more expressive classes of systems, such as infinite systems where data informations is exploited; this would enlarge significantly the applicability of the methods. The second objective is to relax the stability assumption, or equivalently to turn to languages which are not "extension-closed", intending to encompass frameworks like [2] for intermittent faults.

REFERENCES

[1] P. Baroni, G. Lamperti, P. Pogliano, and M. Zanella. Diagnosis of active systems. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 274–278, 1998.

[2] O. Contant, S. Lafortune, and D. Teneketzis. Diagnosis of intermittent faults. *Discrete Event Dynamic Systems: Theory and Applications*, 14(2):171–202, 2004.

[3] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Event Dynamic System : Theory and Applications*, 10(1/2):33–86, January 2000.

[4] E. Fabre, A. Benveniste, C. Jard, L. Ricker, and M. Smith. Distributed state reconstruction for discrete event systems. In *IEEE Control and Decision Conference (CDC)*, Sydney, 2000.

[5] 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.

[6] 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.

[7] S. Jiang, R. Kumar, and H.E. Garcia. Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Transactions on Robotics and Automation*, 19(2):310–323, April 2003.

[8] F. Laroussinie and Ph. Schnoebelen. A hierarchy of temporal logics with past. *Theoretical Computer Science*, 148(2):303–324, September 1995.

[9] R. Milner. A modal characterisation of observable machine-behaviour. In *CAAP*, pages 25–34, 1981.

[10] Y. Pencolé and M-O. Cordier. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence Journal*, 164(1-2):121–170, 2005.

[11] L. Rozé and M.-O. Cordier. Diagnosing discrete-event systems : an experiment in telecommunication networks. In *4th International Workshop on Discrete Event Systems*, pages 130–137, 1998.

[12] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[13] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Failure diagnosis using discrete event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, March 1996.

[14] 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, September 2002.