Optimal observability for diagnosability

Laura Brandán Briones[+], **Alexander Lazovik**[++] **and Philippe Dague**[+] *
LRI, Univ.Paris-Sud, CNRS and INRIA Saclay - Île-de-France
Parc Club Orsay Université, Orsay, F-91893, France.
{laura.brandan,alexander.lazovik,philippe.dague}@lri.fr

Abstract

The diagnosability property recognizes if a system model can be unambiguously diagnosable; that is, if all faults can be detected using only the information given by the observable events. Usually, large and complex systems require an automatic fault detection and isolation, but to specify the minimal observability degree of a system to be diagnosable is not a trivial task.

In this paper we give the necessary and sufficient conditions for observability (the list of observable events) that a system has to maintain to be diagnosable. We concentrate on two problems: first, we transform a diagnosable system into one with minimal degree of observability and still diagnosable. Second, we transform a non-diagnosable system into diagnosable by increasing the degree of observability. We also expand the developed algorithms with several extensions: for distinguishability, for predictability and for extended fault models.

1 Introduction

Fault detection is an extremely important task, and its automatization has been studied for several years. The increasing reliability requirements on autonomous systems, especially mission-critical ones, have resulted in the development of sophisticated methods for the accurate analysis of faults. One of the most important reliability properties in large systems is their diagnosability. The diagnosability property recognizes if a system model can be unambiguously diagnosable, that is, if any occurred fault is detectable within predefined finite number of steps. This recognition is performed by observing the system, e.g., by receiving information from sensors, or by monitoring logs of the running software.

However, when a system is being set up, there is always the problem of how much observability is necessary to keep the system diagnosable? In real world this question may sound as "how many sensors should be installed?" or "how many events should be monitored?". In particular, for non-diagnosable legacy systems this question may sound as "How many new sensors or monitors should be installed, to make the system diagnosable?". More subtly, in diagnosable legacy systems: "Are there too many sensors or monitors, making unnecessary and redundant the collected information?".

This paper gives the most efficient system observability (list of observable events) that keeps the system diagnosable. Following Lin work [8] where it is proved that the most efficient system observability is not unique, we extend it in two ways: First, we give an algorithm that keeps a system diagnosable but with a minimal set of observable events. Second, we present an algorithm that builds the optimal set of observable events to ensure diagnosability in a given system.

Often, especially in legacy systems, one cannot modify anything in the running system but the level and intensity of the observation (e.g., sensors or logs of the system). In this paper we assume that we are only permitted to transform the degree of observability of the system, but not the system itself.

To deal with faults in an uniform way, we introduce a signature definition that describes the situations when a fault occur. The notion of signature is a very well known concept in Continuous Systems (CS). There the diagnosis is often performed on a snapshot of observables, i.e., an evaluation of observable variables at a given time point. If the observation implies a fault occurrence then the observation is called a *signature*. However, in discrete event systems (DES) approaches the diagnosis reasoning is typically more dynamic, i.e., with different observations for a period of time. Therefore, the signature concept has to be re-defined.

In this paper we propose a definition of signatures for DES, which allows us to deal with different fault situations in a uniform way. This definition provides us with two advantages:

1. An efficient method for reducing and expanding the observability of a given system.
2. An extra layer between the actual fault model and the algorithms, which makes possible to develop algorithms independent from the actual nature of the faults.

With signatures, we build a framework that is sufficiently general for our purpose of reducing and expanding observability, and easily extendable to a wide range of problems.

The main contribution of this paper is the analysis of dif-

*This research has been funded by the EU through the FP6 IST project 516933 WS-Diamond($^+$) and the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme($^{++}$)

ferent observability levels for DES. We concentrate on two approaches: first, we transform a diagnosable system into one with minimal observability but still diagnosable. Second, we transform a non-diagnosable system into a diagnosable one by increasing the system observability. We illustrate our propositions with an intuitive example throughout the paper.

Our algorithms work and transform only the observability (the list of observable events) of the system without changing its structure (set of transitions, events or states), which makes them ideal to apply in situations where the system is fixed, and the only change available is the level of system monitoring (e.g., via installation of additional sensors).

This paper also presents extensions to the problem of reducing and expanding observability, namely: (i) distinguishability, where it is important to differentiate the fault types; (ii) predictability, where the fault is predicted rather than detected *a posteriori*; (iii) extended fault model, where a fault is formed by a specific faulty sequence of events, that are not faults by themselves.

Organization of the paper Section 2 overviews the related work. Section 3 provides basic definitions, e.g., label transition system, diagnosability. In Section 4 we introduce the notion of signatures and correct behaviours. The framework for reducing and expanding observability are presented in Section 5, with extensions provided in Section 6. Finally, we draw the conclusions in Section 7.

2 Related work

Diagnosability study for discrete-event systems is not new. In [12] diagnosability is precisely defined and an algorithm for checking diagnosability is provided. In this paper we go beyond diagnosability checking and discuss different levels of observability for diagnosable discrete-event systems. Assuming that we have a diagnosable system we develop an algorithm to find a minimal set of observable events that still keeps the system diagnosable. We also provide an algorithm that transforms non-diagnosable system into diagnosable one by increasing the system observability. There are different ways of representing the partial observability in DES (see, for instance [9]), although, there was only limited attention to how the system observability level affects its diagnosability, e.g., [6]. In contrast to [6], we introduce the notion of a signature to abstract from the underlying failure model. It originally comes from continuous systems [13]. In this work we adopt *signatures* to represent fault executions for DES. It [15] it is proven that finding the optimal (minimal) set of sensors (in our case observable events) for a diagnosable system is NP.

Diagnosability problem is somewhat related to the problem of model checking. However, there is an important difference. Model checking algorithms verify if the (possibly infinite) executions of a system satisfy a given property. Merely, it checks if there is a fault in the system, while diagnosability verifies if the existing faults are detectable. However, there is some work, that shows how diagnosability can be represented as a model checking problem [1].

The approach of extending the fault model presented in Section 6 is not new. For example, in [5] faults are described as formulae in linear temporal logic, in [4] a notion of a supervision pattern is introduced to allow more complicated pattern-based models of failures.

Our approach makes no assumption on whether the system has control over its events, making the system being "passively" diagnosable. In contrast, in active diagnosis [11], the controller is designed to take into account the issues of diagnosability. Similar problems are also tackled in planning, where a planner can decide on the most appropriate actions to deal with faults, see for instance [7].

In this paper our work is mostly oriented to the definition of the general theoretical framework, and does not address the problems related to practical application of the proposed techniques. However, as part of the future work, we plan to evaluate the developed framework against real cases [14].

In [8] to diagnose a fault is to be able to identify in which state or set of states the system is. In contrast, we do not require information about states, we require only to be certain that a fault has occurred.

Our notions of predictability are closely related to the work in [2; 3], where the problem of predicting occurrences of a fault is addressed. We extend these works with definitions of *safe predictability* and *strong predictability*. Furthermore, we define the signature concept for these two definitions and show how our framework can be applied to predictability.

3 Discrete Event Systems

3.1 Preliminaries

Let L be any set. Then with L^* we denote the set of all finite sequences over L, with L^∞ we denote the set of all infinite sequences over L and with L^ω we denote the set of all finite and infinite sequences over L. The empty sequence is denoted by ε. We use $L^+ = L^* \setminus \{\varepsilon\}$. For $\sigma, \rho \in L^\omega$, we say that σ is a *prefix* of ρ and write $\sigma \sqsubseteq \rho$, if $\rho = \sigma\sigma'$ for some $\sigma' \in L^\omega$ (then $\sigma' = \rho - \sigma$). If σ is a prefix of ρ, then ρ is a *continuation* of σ. We call σ a *proper prefix* of ρ and ρ a *proper continuation* of σ if $\sigma \sqsubseteq \rho$, but $\sigma \neq \rho$. We denote by $\mathcal{P}(L)$ the power set of L. Given $L' \subseteq L$ and σ a sequence over L^ω we denote by $\sigma_{L'}$ the restriction of σ over L'.

3.2 Labelled transition systems

Definition 1 (LTS). *A* labelled transition system, LTS*, is a tuple $A = \langle Q, q^0, L, T\rangle$ where Q is a finite set of states; $q^0 \in Q$ is the initial state; L is a finite set of events; $T \subseteq Q \times L \times Q$ is the finite branching transition relation. We denote the components of A by Q_A, q^0_A, L_A, and T_A. We omit the subscript A if it is clear from the context.*

In Figure 1-(A) we represent $A = \langle Q, q^0, L, T\rangle$ a LTS where $Q = \{q_0, \cdots, q_8\}$, $q^0 = q_0$, $L = \{a, b, c, d, e, f_i, f_j\}$ and $T = \{(q_0, b, q_1), (q_0, a, q_2), \cdots, (q_8, a, q_8)\}$.

Definition 2 (path, trace, $|\sigma|$, $q \xrightarrow{\sigma} q'$, cycle, $\breve{\sigma}^k$). *Let $A = \langle Q, q^0, L, T\rangle$ be a LTS, then*
– A path *in A is a sequence $\pi = q_0 a_0 q_1 \ldots$ such that for all i we have $(q_i, a_i, q_{i+1}) \in T$. We denote with* paths$(q)$ *the set of paths starting in q. We use* paths(A) *for* paths(q^0)*. We denote with* paths(q, q') *the set of paths starting in q and ending in q'. We write $q \to q'$, if* paths(q, q') *is not empty and $q \to$, if there exists a state q' such that $q \to q'$.*

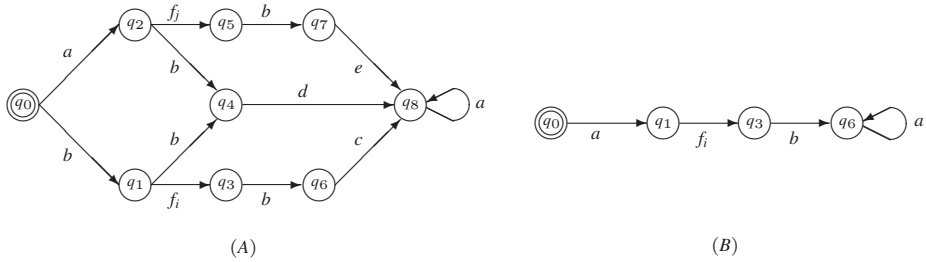

Figure 1: LTSs

– The trace σ *of a path* π*, denoted* $\mathsf{trace}(\pi)$*, is the sequence* $\sigma = a_0 a_1 \ldots$ *of events in L occurring in* π*. We write* $\mathsf{traces}(A) = \{\mathsf{trace}(\pi) \mid \pi \in \mathsf{paths}(A)\}$ *for the set of traces in A, particularly we write* $\mathsf{traces}^{\infty}(A)$ *to denote the set of infinite traces in A. In case* σ *is finite, with* $|\sigma|$ *we denote the length of the trace* σ *and we define by* $\mathsf{last}(\sigma)$ *the last event of* σ*.*
– We write $q \xrightarrow{\sigma} q'$ *if the state* q' *can be reached from the state q via the trace* σ*, i.e., if there is a path* $\pi \in \mathsf{paths}(q, q')$ *such that* $\mathsf{trace}(\pi) = \sigma$*.*
– A cycle *is a non empty element in* $\mathsf{paths}(q, q)$ *for some state q. We denote by* $\mathsf{cycle}(A)$ *all the cycles in A.*
– Given a trace $\sigma \in \mathsf{traces}(A)$*, we denote by* $\check{\sigma}$ *its postlanguage, i.e.,* $\check{\sigma} = \{\rho \in \mathsf{traces}(A) \mid \sigma \sqsubseteq \rho\}$*. Moreover, for a given natural number* $k \in \mathbb{N}$ *we denote by* $\check{\sigma}^k$ *its postlanguage with words with length equal or longer than* $|\sigma| + k$*, i.e.,* $\check{\sigma}^k = \{\rho \in \check{\sigma} \mid k \leq |\rho - \sigma|\}$*.*

We say that a *LTS A* is *live* if for all states there exists a transition initiated in that state, i.e.,

$$A = \langle Q, q^0, L, T \rangle \text{ is live, if and only if, } \forall\ q \in Q : q \rightarrow \quad (1)$$

For example, the *LTS*s from Figure 1-(A) (on the left) and 1-(B) (on the right) are live.

3.3 Observable LTSs with faults

An *observable labelled transition system with faults* is a LTS that has its set of events subdivided into observable events (L_o) and unobservable events (L_u). Moreover, there exists a subset of L_u that represents fault events (L_f).

Definition 3 (observable LTS(L_f)). *An* observable labelled transition system with faults*, denoted LTS(L_f), is a tuple* $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle$ *where* $\langle Q, q^0, L, T \rangle$ *is a LTS and:*
– The set of events L is partitioned into a set of observable events*,* L_o*, and a set of* unobservable events*,* L_u*, with* $L = L_o \cup L_u$ *and* $L_o \cap L_u = \emptyset$*.*
– There is a subset of the unobservable events, called the fault events*, denoted* L_f*.*

In a sense, an observable LTS(L_f) is about hiding the faults and some other events. From now on, we refer to LTS(L_f) as to observable LTS(L_f) unless we state the opposite.

As an example, Figure 1-(A) represents a LTS(L_f) with: $L_u = \{f_i, f_j\}$ and $L_f = \{f_i, f_j\}$.

Definition 4 (observable trace, traces^f, $\mathsf{traces}^{f,k}$, $f \in \sigma$). *Let* $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle$ *be a LTS(L_f), then:*
– The observable trace *of a trace* σ*, denoted* σ_{L_o}*, is the sequence* $a_0 a_1 \ldots$ *of events in* L_o *occurring in* σ*.*
– We denote by $\mathsf{traces}^f(A)$ *the set of traces in A that end with a fault, i.e.,* $\mathsf{traces}^f(A) = \{\sigma \in \mathsf{traces}(A) \mid \sigma \in L^* L_f\}$
– Given a natural number $k \in \mathbb{N}$ *we denote by* $\mathsf{traces}^{f,k}(A)$*, the set of traces* σ *such that there exists another trace* σ' *that ends in a fault and* σ *extends* σ' *with* $|\sigma'| + k$ *or more events, i.e.,* $\mathsf{traces}^{f,k}(A) = \{\sigma \in \mathsf{traces}(A) \mid \exists\ \sigma' \in \mathsf{traces}^f(A) \wedge \sigma \in \check{\sigma'}^k\}$.
– Given a trace σ*, we write* $f \in \sigma$ *to denote that* σ *has a fault, i.e.,* $\sigma \in L^* L_f L^{\omega}$*.*

We say that a LTS(L_f) *A* is *convergent* if it does not have cycles with non-observable events, i.e.,

$$A \text{ is } \textit{convergent}, \text{ if and only if, } \forall\, \pi \in \mathsf{cycle}(A)\ \exists\ a \in L_o : a \in \pi \quad (2)$$

3.4 Diagnosability

The diagnosability of a system means that its model, supposed to be a LTS(L_f), can be unambiguously diagnosable, where a diagnosable LTS(L_f) is defined as being able to detect a fault occurrence within a finite number of steps based only on the observable traces.

Definition 5 (diagnosability). *Let* $A = \langle Q, q^0, L, T, L_u, L_f \rangle$ *be an observable LTS(L_f), then A is* diagnosable *if the following holds,* $\exists\ n \in \mathbb{N} : \forall\ \rho \in \mathsf{traces}^{f,n}(A) :$ *if* $\alpha \in \mathsf{traces}(A) : \rho_{L_o} = \alpha_{L_o}$ *then* $f \in \alpha$*.*

The previous definition is a reformulation of the known Sampath [12] diagnosability definition, for the case with only one type of fault[1]. For example, the LTS(L_f) from Figure 1-(A), with $L_u = \{f_i, f_j\}$ is convergent and diagnosable.

Property 1. *Let* $A = \langle Q, q^0, L, T, L_u, L_f \rangle$ *be an observable LTS(L_f), then A is* diagnosable *if the following holds,* $\exists\ n \in \mathbb{N} : \forall\ \rho \in \mathsf{traces}^{f,n}(A) :$ *if* $\alpha \in \mathsf{traces}(A) :$ $\rho_{L_o} = \alpha_{L_o}$ *then* $\forall\, \alpha' \in \mathsf{traces}^{\infty}(A) : \alpha \sqsubseteq \alpha' : f \in \alpha'$*.*

The proof is given in [2].

Remark 1: For a given L_u, if the LTS(L_f)$A(L_u)$ is diagnosable, then $\forall\, L_f \subseteq L'_u \subseteq L_u$, the LTS($L_f$) $A(L'_u)$ is diagnosable.

Remark 2: A LTS(L_f)$A(L)$ is never diagnosable (from the moment it has at least one correct trace and one faulty trace).

4 Observability and signatures

4.1 Observability

Some systems can be non diagnosable even when all events, except fault events, are observable. We denote such systems

[1] In Section 6.1 we extend our approach to several types of faults

necessarily non diagnosable. A more interesting situation is when a system is diagnosable with faults being the only unobservable events. We denote such systems *possibly diagnosable*. To distinguish if there exists an observability degree, for a particular system, that makes it diagnosable, we propose thus the following definition.

Definition 6 (necessarily non/possibly diagnosable). *A LTS(L_f) A is called:*
– Necessarily non diagnosable *if the LTS(L_f) $A(L_f)$ is not diagnosable.*
– Possibly diagnosable *if the LTS(L_f) $A(L_f)$ is diagnosable.*

From now, we assume that all LTS(L_f) A, that we work with, are live, convergent and possibly diagnosable.

4.2 Signatures in DES

In this section we introduce the notion of *signatures* that originally comes from continuous systems [13]. In this work we adopt *signatures* to represent faults executions for discrete-event systems. A *signature* is a regular expression that denotes the set of traces with faults.

The *observable correct behaviour*, denoted **c**, is defined by the regular expression (with only observable events) that denotes all correct infinite traces, i.e., infinite traces that do not have faults.

Definition 7 (observable correct behaviour). *Given a LTS(L_f) A we define its* observable correct behaviour *as*

$$\mathbf{c} = \{\sigma \in L_o^\omega \mid \sigma \in \mathsf{traces}^\infty(A) : f \notin \sigma\}$$

In contrast with the observable correct behaviour, the *observable signature* is the regular expression that denotes all observable prefixes of faulty traces that are not prefix of an element of the observable correct behaviour.

To define observable signatures we use $\mathsf{traces}^{\mathbf{z}}$ to denote the set of all traces with exactly **z** events after the first occurrence of a fault. In [10] it is shown, that for an exhaustive diagnosability check, it is necessary to check traces of a maximal length $\frac{|\mathcal{Q}|^2-|\mathcal{Q}|}{2}$ (and, therefore we can choose $\mathbf{z} \leq \frac{|\mathcal{Q}|^2-|\mathcal{Q}|}{2}$), what is often impractically high. In practice, to build a correct signature, one is sometimes forced to set an upper bound for the length of faulty traces.

Definition 8 (observable signature). *Given a diagnosable LTS(L_f) $A = \langle Q, q^0, L, T, L_u, L_f\rangle$, its* observable signature *is defined as*

$$\mathbf{r} = \{\sigma \in L_o^* \mid \exists\, \alpha \in \mathsf{traces}^{\mathbf{z}}(A) : \sigma \sqsubseteq \alpha_{L_o} : \nexists \sigma'' \in \mathbf{c} : \sigma \sqsubseteq \sigma''\}$$

Note that the observable correct behaviour is not the complement of the observable signature: there exist traces that do not belong to either class.

Remark 3: A consequence of the diagnosability definition (Definition 5) is that A being diagnosable ensures that **r** is not empty (from the moment A contains at least one fault) with a $\mathbf{z} = \frac{|\mathcal{Q}|^2-|\mathcal{Q}|}{2}$.

Following the example from Figure 1-(A), with $L_u = \{f_i, f_j\}$ we can find $\mathbf{c} = abda^\infty + bbda^\infty$, $\mathbf{z} = 5$, then $\mathsf{traces}^{\mathbf{z}} = \{af_jbea^3, bf_ibca^3\}$. So, finally we can obtain the observable signature $\mathbf{r} = abe + abea + abea^2 + abea^3 + bbc + bbca + bbca^2 + bbca^3$.

In particular, we let observable signatures represent traces that do not have faults, but are prefixes of faulty traces. In Figure 1-(B) we represent a diagnosable system that is bound to have a fault with $L_u = L_f = \{f_i\}$. We obtain $\mathbf{c} = \emptyset$ because there is no infinite trace without fault. In addition, $\mathsf{traces}^{\mathbf{z}} = \{af_ibaaa\}$ then we can obtain **r** as $a + ab + aba + aba^2 + aba^3$.

5 Reducing and expanding observability

5.1 Reducing the observability

In this section we find a minimal set of observable events that still keeps the system diagnosable. We do it by reducing the set of observable events as much as possible still maintaining the diagnosability property.

The observable signatures defined in Definition 8 describe the observable part of traces with faults or traces that certainly will produce a fault. Although, for the following proofs we use a more restricted version of signature, called *long signature* and denoted **lr**. Long signatures do not contain traces that are prefix of another one, i.e.,

$$\mathbf{lr} = \{\sigma \in L_o^* \mid \sigma \in \mathbf{r} \wedge \nexists\, \sigma' \in \mathbf{r} : \sigma \sqsubset \sigma'\} \qquad (3)$$

For example, our previous signature $\mathbf{r} = abe + abea + abea^2 + abea^3 + bbc + bbca + bbca^2 + bbca^3$ is converted to the long signature $\mathbf{lr} = abea^3 + bbca^3$.

Moreover, for a trace σ we abuse the notation: $\sigma \in \mathbf{r}$, $\sigma \in \mathbf{lr}$, and $\sigma \in \mathbf{c}$ to denote that the observable trace of σ (i.e., σ_{L_o}) is in **r** or **c** respectively.

Structural differences, written $A \not\equiv B$, relate two sets of traces that do not have any trace in common, nor any prefix of a trace that belongs to the other set, except of the ε trace. Formally, structural differences are defined as follows.

Definition 9 (structural differences ($\not\equiv$)). *Let A and B be sets of traces, then $A \not\equiv B$ means*
$(\forall\, \sigma \in A : \forall\, \sigma \sqsubseteq \sigma' : \sigma' \notin B) \wedge (\forall\, \sigma \in B : \forall\ \sigma \sqsubseteq \sigma' : \sigma' \notin A)$

Lemma 1. *Let $A(L_u) = \langle Q, q^0, L, T, L_u, L_f\rangle$ be a diagnosable LTS(L_f), then if $a \in L_o : \mathbf{lr}_{L_o\setminus\{a\}} \not\equiv \mathbf{c}_{L_o\setminus\{a\}}$ and $A(L'_u) = \langle Q, q^0, L, T, L'_u, L_f\rangle$, with $L'_u = L_u \cup \{a\}$ is convergent then $A(L'_u)$ is diagnosable, and not diagnosable otherwise.*

Proof. Let $\mathbf{lr}' = \mathbf{lr}_{L_o\setminus\{a\}}$ and $\mathbf{c}' = \mathbf{c}_{L_o\setminus\{a\}}$. Note that for every $\sigma \in \mathbf{lr}$ and $\alpha \in \mathbf{c}$ it follows that $\sigma \in \mathbf{lr}'$ and $\alpha \in \mathbf{c}'$. To prove the first part, we have to show that $\exists\, n : \forall\, |\sigma'| > n : \sigma_{L'_o} = \sigma'_{L'_o}$ then $f \in \sigma', \sigma_{L'_o} = \alpha_{L'_o}$, where $L'_o = L_o\setminus\{a\}$. Let $n = z = \frac{|\mathcal{Q}|^2-|\mathcal{Q}|}{2}$. From $\forall\, \sigma \in \mathsf{traces}^{f,z}(A(L'_u)) : \sigma \in \mathsf{traces}^{f,z}(A(L_u))$, it follows $\sigma \in \mathbf{lr} \Rightarrow \sigma \in \mathbf{lr}'$. Suppose, $\exists\, \sigma_1 : f \notin \sigma_1 \wedge \sigma_{L'_o} = \sigma_{1L'_o}$. Because $\sigma_{L'_o} = \sigma_{1L'_o} \wedge \sigma \in \mathbf{lr}'$ then $\sigma_1 \in \mathbf{lr} \Rightarrow \sigma_1 \in \mathbf{lr}'$. There are two possible cases: *1.* $\exists\, \sigma_2 \in \mathsf{traces}^\infty : \sigma_1 \sqsubseteq \sigma_2 \wedge f \notin \sigma_2 \wedge \sigma_2 \in \mathbf{c}$, then $\sigma_2 \in \mathbf{c}'$. Because $\mathbf{lr}_{L'_o} \not\equiv \mathbf{c}_{L'_o}$ follows that $\sigma_1 \notin \mathbf{lr}'$. But then $\sigma \notin \mathbf{lr}'$ which contradicts with $\sigma \in \mathbf{lr}'$. *2.* $\forall\, \sigma_2 \in \mathsf{traces}^\infty : \sigma_1 \sqsubseteq \sigma_2 \wedge f \in \sigma_2$. That also means that $\exists\, \sigma_3 \notin \mathsf{traces}^\infty : \sigma_3 \sqsubseteq \sigma_2 \wedge f \in \sigma_3$. From Property 1 it follows then that the system is diagnosable.
Let us now prove, that if the assumption $\mathbf{lr}_{L'_o} \not\equiv \mathbf{c}_{L'_o}$ does not hold, then the system becomes not diagnosable if a is removed from L_o. If the assumption does not hold, then $\exists\, \sigma \in \mathbf{lr}' : \exists\, \sigma_1 \in \mathbf{c} : \sigma \sqsubseteq \sigma_1$. And from it follows either

Algorithm 1 Reducing the observability

```
1: Input: System A, observable events L_o, observable correct behavior c, observable signature r
2: Output: Minimal set of observable events L'_o
3: lr = reduceToLongSignature(r)
4: S_max = ∅
5: for all S ⊆ L_o do
6:     if checkUnObSet(A, S, c, lr) ∧ |S_max| < |S| then
7:         S_max = S
8: return L_o \ S_max
9: function checkUnObSet(A, S, c, lr)
10:    with a ∈ S do
11:        if isConvergent(A, L_o \ a) ∧
           checkUnObserve(c, lr, a) then
12:            return checkUnObSet(A, S \ a, c − {a}, lr − {a})
13:    return S = ∅
14: function checkUnObserve(c, lr, a)
15:    for all σ ∈ (c − {a}) do
16:        for all σ' ∈ prefix(σ) ∧ (| σ' | ≤ | lr |) do
17:            if σ' ∈ (lr − {a}) then
18:                return false
19:    return true
```

$\exists\,\sigma_f \in \mathbf{c}' : f \in \sigma_f$ or $\exists\,\sigma \in \mathbf{lr}' : \forall\,\sigma_1 : \sigma \sqsubseteq \sigma_1 : f \notin \sigma_1$. Let us first assume that $\exists\,\sigma_f \in \mathbf{c}' : f \in \sigma_f$. Then, $\exists\,\omega_1 \in \mathbf{c}' : \omega_1 = \sigma_{L'_o}$. From the way $\mathbf{c}'$ is build, it follows that $\exists\,\omega \in \mathbf{c} : \omega_1 = \omega - \{a\}$. From Definition 7 it follows that $\sigma_f \notin \mathbf{c}$, that is, $\exists\,\sigma \in \mathbf{c} : \sigma_{L_o} = \omega \wedge f \notin \sigma$, and, therefore, $\omega_1 = \sigma_{L'_o} = \sigma_{f_{L'_o}}$. We have just shown that two infinite traces, one having fault, and one not, have the same observable behavior, and, according to Definition 5, the system is not diagnosable. The case 2 when $\exists\,\sigma \in \mathbf{lr}' : \forall\,\sigma_1 : \sigma \sqsubseteq \sigma_1 : f \notin \sigma_1$ is proven in the same way as the first case.

For the example from Figure 1-(A) with long signature: $\mathbf{lr} = abea^3 + bbca^3$ and correct behaviour: $\mathbf{c} = abda^\infty + bbda^\infty$. So, from Lemma 1, we may drop the event b from observable events, and the system remains diagnosable for $L_u = \{f_i, f_j, b\}$ and $L_o = \{a, c, d, e\}$.

We can easily convert long signature into signature allowing any prefix of a trace in a long signature that is not a subtrace of a correct behaviour. For example, starting from the long signature $\mathbf{lr} = abea^3 + bbca^3$ from the system $A(L_u)$ with $L_u = \{f_i, f_j\}$, $L_o = \{a, b, c, d, e\}$, and correct behaviour $\mathbf{c} = abda^\infty + bbda^\infty$ we obtain $\mathbf{r} = abe + abea + abea^2 + abea^3 + bbc + bbca + bbca^2 + bbca^3$.

Theorem 1. *Let $A(L_u) = \langle Q, q^0, L, T, L_u, L_f\rangle$ be a diagnosable LTS(L_f), then we obtain a minimal set of observable events by repeatedly applying Lemma 1, $L'_o \subseteq L_o$ such that $A(L'_u) = \langle Q, q^0, L, T, L'_u, L_f\rangle$ is diagnosable, with $L'_u = L \backslash L'_o$.*

The proof for Theorem 1 is as follows. First, we derive the signature and correct behaviour from the diagnosable system A. Second, we obtain the long signature from the signature of the system. Third, we repeatedly apply Lemma 1 for all observable events until there does not exist any events in the observable events that can be converted to unobservable. Finally, we obtain back the signature from the last long signature that we obtained.

The above procedure can be performed according to different orders, depending on which observable events we choose to turn into unobservable ones. Note, that there always exists a minimal order w.r.t. the amount of observable events. The algorithm itself for reducing the observability is shown in Algorithm 1. It provides an algorithmic view for Lemma 1 and Theorem 1. Given a system, its observable events and observable signature, the algorithm returns a minimal set of observable events. It works in the following way. In line 3, it reduces the observable signature according with (3). In line 5-7, the algorithm chooses the set S with maximal cardinality, which is built by functions *checkUnObSet* and *checkUnObserve*. The functions form the set S by iteratively reducing the set of observable events as far as observable signature for **lr** and observable correct behavior **c** are still distinguishable, in the same way as it was defined by Lemma 1.

In the following example we show how we obtain the minimal set of observable events for the system A from Figure 1-(A). Starting from $A(L_u) = \langle Q, q^0, L, T, L_u, L_f\rangle$ with $L_u = \{f_i, f_j\}$, $L_o = \{a, b, c, d, e\}$, $\mathbf{r} = abe + abea + abea^2 + abea^3 + bbc + bbca + bbca^2 + bbca^3$ and $\mathbf{c} = abda^\infty + bbda^\infty$. Then, (i) we obtain $\mathbf{lr} = abea^3 + bbca^3$; (ii) we convert c into unobservable, having $\mathbf{lr} = abea^3 + bba^3$ and $\mathbf{c} = abda^\infty + bbda^\infty$; (iii) we convert b into unobservable, having $\mathbf{lr} = aea^3 + a^3$ and $\mathbf{c} = ada^\infty + da^\infty$; (iv) we convert e into unobservable, having $\mathbf{lr} = aa^3 + a^3$ and $\mathbf{c} = ada^\infty + da^\infty$; and (v) we reconstruct the signature as $\mathbf{r}' = aa + a^3 + aa^3$. It is easy to note that $A(L'_u) = \langle Q, q^0, L, T, L'_u, L_f\rangle$ with $L'_u = \{f_i, f_j, b, e, c\}$, $L_o = \{a, d\}$ is diagnosable.

5.2 Expanding the observability

In this section we present the algorithm to transform a non-diagnosable system into a diagnosable one, expanding its set of observable events.

We assume that the system is possibly diagnosable (Definition 6). Thus, if we consider the system with all events, except faults, as observable, then the system is diagnosable. We define $S_{\sigma\alpha}$ as a set of sets of events that distinguish traces σ and α. So, in a possibly diagnosable system with two traces σ and α with the same observability such that one has a fault, and afterwards it has at least n events, and the latter one without a fault, we define $S_{\sigma\alpha}$ as a set of sets of events (not from L_f) that makes σ and α distinguishable.

Definition 10 ($S_{\sigma\alpha}$). *Let $A(L_u)$ be a possibly diagnosable LTS(L_f), then $\forall\ \sigma, \alpha \in \mathsf{traces}(A) : \sigma \in \mathsf{traces}^{f,n}(A) \wedge f \notin \alpha : \sigma_{L_o} = \alpha_{L_o}$, we define*

$$S_{\sigma\alpha} = \{O \in L_o \backslash L_f \mid \sigma_{L_o \cup O} \neq \alpha_{L_o \cup O}\}$$

where n is the bound, given by the diagnosability definition, for the system $A(L_u)$.

Considering the system presented in Figure 1-(A), with $L_o = \{a, b\}$, $L_u = \{c, d, e, f_i, f_j\}$; $\sigma = bf_ibc$ and $\alpha = bbd$ (so $\sigma_{L_o} = bb$ and $\alpha_{L_o} = bb$) we have $S_{\sigma\alpha} = \{\{d\}, \{d, c\}, \{c\}\}$. Now, with $\sigma' = af_jbe$ and $\alpha' = abd$, we obtain $S_{\sigma'\alpha'} = \{\{e\}, \{d, e\}, \{d\}\}$.

A *minimal distinguishable set*, denoted by S, represents a set that includes at least one set for all $S_{\sigma\alpha}$.

Definition 11 (S). *S is a* minimal distinguishable set*, if it has minimal cardinality and for all $S_{\sigma\alpha}$ with*

Algorithm 2 Expanding the observability

1: **Input:** System A, observable events L_o
2: **Output:** Minimal set of observable events L'_o
3: **for all** $\sigma \in \mathsf{traces}^{f,n} \wedge |\sigma| \leq \mathbf{z}$ **do**
4: **for all** $\alpha \in \mathsf{traces}(A) : \alpha_{L_o} = \sigma_{L_o} \wedge f \notin \alpha$ **do**
5: $S_{\sigma\alpha} = \{O \mid \sigma_{L_o \cup O} \neq \alpha_{L_o \cup O} \wedge f \notin O\}$
6: $S = \emptyset$
7: **for all** $S_{\sigma\alpha}$ **do**
8: $S = \{B' \cup B'' \mid (B', B'') \in S \times S_{\sigma\alpha}\}$
9: **for all** $B_i, B_j \in S, B_i \subseteq B_j$ **do**
10: $S = S - B_j$
11: **return** $L_o \cup B_{min}$, where $B_{min} \in S \wedge |B_{min}| = min\,|B|, B \in S$

$\sigma, \alpha \in \mathsf{traces}(A) : \sigma \in \mathsf{traces}^{f,n}(A) \wedge f \notin \alpha : \sigma_{L_o} = \alpha_{L_o}$, *there exists* $B \in S_{\sigma\alpha} : B \subseteq S$.

For the example shown in Figure 1-(A) we have $S = \{d\}$.

Theorem 2. *Let* $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle$ *be a possible diagnosable but non-diagnosable LTS*(L_f)*, and let S defined as in Definition 11, then* $A(L'_u) = \langle Q, q^0, L, T, L'_u, L_f \rangle$ *is diagnosable, with* $L'_u = L_u \backslash S$.

Proof. Suppose $A(L'_u)$ is non-diagnosable. Therefore there exists $\sigma, \alpha \in \mathsf{traces}(A)$ such that $\sigma \in \mathsf{traces}^{f,n}(A) \wedge f \notin \alpha : \sigma_{L'_o} = \alpha_{L'_o}$. Then, using Definition 10 there exists $S_{\sigma,\alpha}$ and, moreover, from Definition 11 there exists $B \in S_{\sigma,\alpha} : B \subseteq S$, so $\sigma_{L'_o} \neq \alpha_{L'_o}$. We have a contradiction, that comes from supposing that $A(L'_u)$ is non-diagnosable.

The algorithm is provided in Algorithm 2. It transforms the system into a diagnosable one without changing its structure (set of transitions, events or states) and only by expanding its observability. Moreover, the algorithm finds a minimal set of events that should be added to the initial set of observable events. The algorithm itself is based on the definitions and the theorem above: lines 3-5 refer to Definition 10, lines 6-10 refer to Definition 11 and Theorem 2. Note, that in line 3 we use $\mathbf{z}$ to limit the maximum length of σ. For an exhaustive search $\mathbf{z}$ has to be equal to $\frac{|Q|^2-|Q|}{2}$. In practice it is often possible to provide a better bound (e.g., [10]). Note that, depending on the initial L_o, L'_o is not necessarily a minimal set of observable events making the system diagnosable. However, the minimality can always be reached by applying the algorithm presented in Section 5.1 from $L_o = L \backslash L_f$.

In our example from Figure 1-(A) with $L_o = \{a, b\}$, following Algorithm 2 we obtain $S = \{d\}$. In contrast with the minimal set of observable events for this diagnosable system that is $L_o^{min} = \{a, d\}$, as we already pointed out in Section 5.1, we obtain $A(L'_u) = \langle Q, q^0, L, T, L'_u, L_f \rangle$ with $L'_o = \{a, b, d\}$.

6 Extended models

In this section we introduce various extensions to the diagnosability model that we presented in Section 3.4. Within the framework defined in Section 4, it is possible to reuse the algorithms from Theorem 1 and Theorem 2, and, in the same time, take into account several extensions: distinguishability, predictability, and extended fault models. In the following sections we show what has to be modified in the proposed model and algorithms to deal with each particular case.

6.1 Distinguishability

The problem of *distinguishability* arises when we are interested in distinguishing different types of faults rather than in a simple indication whether a fault occurred or not.

In this section we partition the set of faults (subset of the unobservable events) into classes of faults, i.e., $\Pi_f = \{L_{f_1}, \cdots, L_{f_m}\}$, where L_{f_i} represents all faults with type f_i.

Definition 12 (observable LTS(Π_f)). *An* observable labelled transition system LTS(Π_f) with fault types*, denoted by* Π_f*,* $A(L_u) = \langle Q, q^0, L, T, L_u, \Pi_f \rangle$ *is a LTS*(L_f)*, where the set of* fault events *(*L_f*) is partitioned into* $\Pi_f = \{L_{f_1}, \cdots, L_{f_m}\}$*, i.e.,* $L_f \subseteq L_u$*,* $L_f = L_{f_1} \cup \cdots \cup L_{f_m}$ *and* $\forall\ i \neq j : L_{f_i} \cap L_{f_j} = \emptyset$.

So an observable LTS(Π_f) is a normal LTS with a clear distinction between observable and unobservable events and inside the unobservable events there is a subset of fault events subdivided into classes.

Definition 13 (traces^{f_i}, $\mathsf{traces}^{f_i,k}$, $f_i \in \sigma$). *Let* $A = \langle Q, q^0, L, T, L_u, \Pi_f \rangle$ *be a LTS*(Π_f)*, then:*
– Given a type of fault f_i*, we denote by* $\mathsf{traces}^{f_i}(A)$ *the set of traces in A that end with a fault of type* f_i*, i.e.,* $\mathsf{traces}^{f_i}(A) = \{\sigma \in \mathsf{traces}(A) \mid \sigma \in L^* L_{f_i}\}$*;*
– Given a type of fault f_i *and a natural number* $k \in \mathbb{N}$ *we denote by* $\mathsf{traces}^{f_i,k}(A)$*, the set of traces* σ *such that there exists another trace* σ' *that ends in a fault of type* f_i *and* σ *extends* σ' *with length longer or equal to the length of* σ' *plus k, i.e.,* $\mathsf{traces}^{f_i,k}(A) = \{\sigma \in \mathsf{traces}(A) \mid \exists\ \sigma' \in \mathsf{traces}^{f_i}(A) \wedge \sigma \in \breve{\sigma}'^k\}$*;*
– Given a trace σ*, we write* $f_i \in \sigma$ *to denote that* σ *has a fault of type* f_i*, i.e.,* $\sigma \in L^* L_{f_i} L^\omega$.

As follows, we re-define diagnosability and observable signatures for LTS(Π_f).

Definition 14 (diagnosability in LTS(Π_f)). *Let* $A(L_u) = \langle Q, q^0, L, T, L_u, \Pi_f \rangle$ *be a LTS*(Π_f)*, then the set* $\mathsf{traces}(A)$ *is diagnosable if the following holds,* $\forall\ 1 \leq i \leq m : \exists\ n_i \in \mathbb{N} : \forall\ \rho \in \mathsf{traces}^{f_i,n_i}(A) :$ *if* $\alpha \in \mathsf{traces}(A) : \rho_{L_o} = \alpha_{L_o}$ *then* $f_i \in \alpha$.

Definition 15 ($\mathbf{r}^i$). *Given a diagnosable LTS*(Π_f) *A and* f_i *a fault type;* $\mathbf{r}^i$ *is the* observable signature of f_i *if it observable prefixes of traces contans a fault of type* f_i *that are not prefix of a correct trace.*

$$\mathbf{r}^i = \{\sigma \in L_o^* \mid \exists\ \alpha \in \mathsf{traces}^{f_i,n_i}(A) : \sigma \sqsubseteq \alpha_{L_o} : \nexists\, \sigma'' \in \mathbf{c} : \sigma \sqsubseteq \sigma''\}$$

Definition 16 (signature in LTS(Π_f)). *Given a LTS*(Π_f) *A and* $\mathbf{r}^1, \cdots, \mathbf{r}^m$ *the set of observable signatures for fault types* $f_1 \cdots f_m$*, we define the* observable signature of *A:*

$$\mathbf{r} = \mathbf{r}^1 + \cdots + \mathbf{r}^m$$

Now we can reformulated Lemma 1 w.r.t. different faults.

Lemma 2. *Let* $A(L_u) = \langle Q, q^0, L, T, L_u, \Pi_f \rangle$ *be a diagnosable LTS*(Π_f)*, then if* $a \in L_o : \mathbf{r}_{L_o \backslash \{a\}} \not\equiv \mathbf{c}_{L_o \backslash \{a\}}$ *and* $\forall\ i \neq j : \mathbf{r}^i{}_{L_o \backslash \{a\}} \not\equiv \mathbf{r}^j{}_{L_o \backslash \{a\}}$ *then* $A(L'_u) =$

$\langle Q, q^0, L, T, L'_u, \Pi_f \rangle$, *with* $L'_u = L_u \cup \{a\}$, *is diagnosable.*

With this new lemma, Theorem 1 remains true. Moreover, if we redefine Definition 10 and Definition 11 as follows, also Theorem 2 remains true.

Definition 17 ($S_{\sigma\alpha}$, S in LTS(Π_f)). *Let $A(L_u)$ be a possibly diagnosable LTS(Π_f), then* $\forall\ 1 \leq i \leq m : \forall\ \sigma, \alpha \in \mathsf{traces}(A) : \sigma \in \mathsf{traces}^{f_i,n_i}(A) \wedge f_i \notin \alpha : \sigma_{L_o} = \alpha_{L_o}$, *we define:*
– $S_{\sigma\alpha} = \{B \subseteq L_u \backslash L_f \mid \sigma_{L_o \cup B} \neq \alpha_{L_o \cup B}\}$
– *S is a minimal cardinality set, such that*
$\forall\ S_{\sigma\alpha} : \exists\ B \in S_{\sigma\alpha} : B \subseteq S$.

6.2 Predictability

In some cases, e.g., in mission critical scenarios, it is important to achive the prediction of a possible fault situation rather than a post-fault detection. For such scenarios we have to ensure that the fault is predictable, and we come to the problem of *predictability*. Predictability study is not new, it was first introduced in [2]. However, in [2] authors investigate only case of strongly predictable systems, ignoring the notion of safe predictability.

However, since predictability is about future, and future is non-deterministic, we have two types of predictability: safe predictability and strong predictability. *Safe predictability* refers to an observation of a sequence of events that may *potentially* end in a fault; while *strong predictability* refers to cases that will end in a fault (when the fault is unavoidable).

Definition 18 (Safe predictability). *A LTS(L_f) $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle$ is* safely predictable *if the following holds:* $\forall\ \sigma \in \mathsf{traces}^f(A)$ *if* $\exists\ \alpha \in \mathsf{traces}(A) : f \notin \alpha \wedge \sigma_{L_o} = \alpha_{L_o}$ *then* $\exists\ \alpha' \in \mathsf{traces}(A) : \alpha \sqsubset \alpha' \wedge f \in \alpha'$.

Definition 19 (Strong predictability). *A LTS(L_f) $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle$ is* strongly predictable *if the following holds:* $\forall\ \sigma \in \mathsf{traces}^f(A)$ *if* $\exists\ \alpha \in \mathsf{traces}(A) : f \notin \alpha \wedge \sigma_{L_o} = \alpha_{L_o}$ *then* $\forall\ \alpha' \in \mathsf{traces}^\infty(A) :$ *if* $\alpha \sqsubset \alpha'$ *then* $f \in \alpha'$.

In our example, from Figure 1-(A), let assume we have $L_o = \{a, b\}$. Then, the system is safely predictable, since whenever we observe events a or b we know that we have the possibility to have a fault in the future. However, the system is not strongly predictable, since there is no sequence of observable events that unambiguously predicts either f_i or f_j occurrence. In particular, if we remove b from the list of observable events, the system is not safely predictable for all faults but it is safely predictable w.r.t. f_j. On the other hand, the example from Figure 1-(B) is clearly strongly predictable with a list of observable events like $L_o = \{a\}$.

Within the defined framework, as in [2; 3] where similar results are presented, strong predictability implies diagnosability and safe predictability:

Property 2. *Strong predictability implies diagnosability. It follows immediately from Property 1.*

Property 3. *Strong predictability implies safe predictability.*

A *predictability signature* is defined as a set of observable events that, if occurred, always or potentially (depends on the type of predictability) bring the execution to a fault event.

Definition 20 (safe/strong predictable signatures). *Given a $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle \in LTS(L_f)$, then:*
– *if A is* safe predictable *then its observable-safe-predictable-signatures (*safe_**pr***) is*

$$\textit{safe\_}\mathbf{pr} = \{\sigma \in L_o^* \mid \exists\ \alpha \in \mathsf{traces}^f(A) : \sigma_{L_o} \sqsubseteq \alpha_{L_o}\}$$

– *if A is* strong predictable *then its observable-strong-predictable-signatures (*strong_**pr***) is*

$$\textit{strong\_}\mathbf{pr} = \{\sigma \in L_o^* \mid \forall\ \alpha \in \mathsf{traces}^\infty(A) : \sigma_{L_o} \sqsubseteq \alpha_{L_o} : f \in \alpha\}$$

We can apply the previous algorithms directly for strong predictable systems.

Property 4. *Given a strong predictable system $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle \in LTS(L_f)$ with* **c** *as its correct behaviour (Definition 7) and strong_**pr** as its strong-predictable-signatures (Definition 20), then:*
(i) The algorithm presented in Theorem 1 reduces the set of observable events correctly, keeping the system strong predictable.
(ii) The algorithm presented in Theorem 2 expands the set of observable events correctly, keeping the system strong predictable.

We also can apply the algorithms for safe predictable systems, but it is necessary to adapt the notion of *observable correct behaviour*.

Definition 21 (secure correct behaviour). *Given a system $A(L_u) \in LTS(L_f)$ we define the* observable secure correct behaviour *as*

$$\mathbf{sc} = \{\sigma \in L_o^* \mid \sigma \in \mathsf{traces}^\infty(A) : f \notin \sigma \wedge (\forall\ \alpha \sqsubseteq \sigma : \nexists\ \alpha \sqsubseteq \alpha' : f \in \alpha')\}$$

This definition is an adaptation of that of observable correct behaviour (Definition 7 in Section 4.2). The idea is analogous to the previous one; the main novelty here is that subtraces of correct traces can never be part of observable-safe-predictable-signatures. In this way we still keep the structural difference between correct behaviours and signatures.

Property 5. *Given a safe predictable system $A(L_u) = \langle Q, q^0, L, T, L_u, L_f \rangle \in LTS(L_f)$ with* **sc** *as its secure correct behaviour (Definition 21) and safe_**pr** as its safe-predictable-signatures (Definition 20), then:*
(i) The algorithm presented in Theorem 1 reduces the set of observable events correctly, keeping the system safe predictable.
(ii) The algorithm presented in Theorem 2 expands the set of observable events correctly, keeping the system safe predictable.

We leave the properties in this section without proofs, since the proofs are analogous to Theorem 1 and Theorem 2.

6.3 Extended fault model

In this section we define an extended fault model, where a fault is formed by a specific fault sequence of events, that are not faults by themselves. Consider an example of driving a vehicle, where driving having doors open is a fault, while in

most other situations it is an absolutely legal and expected action. In this case, the fault is defined not by a faulty event but rather by a sequence of events that forms a fault. Furthermore, a fault sequence can contain any arbitrary events that do not contribute to the fault. For the vehicle example, we may have something occurred between opening the door and driving, and still, if the door is open, we are not allowed to drive. We define this problem as diagnosability problem in an *extended fault model*.

In the extended fault model the fault is defined as a sequence of events. The fault is considered to be occurred when the last event of the sequence occurs. Besides, events in the sequence are not required to occur one after another one, we may have other events happening in the meantime.

An *extended fault* is defined by a sequence of events, denoted ρ_f. The set of *fault executions* is then defined as $\{\sigma \mid \rho_f \subseteq \sigma\}$, where $\rho_f \subseteq \sigma$ means that the events from ρ_f happened in σ in order but not necessarily consecutively. So, if $\rho_f = ab$ then the trace $\sigma = cacbe$ is $\subseteq$ w.r.t. ρ_f. We denote by ρ_f^l the last event of the fault sequence ρ_f, i.e., $\rho_f^l = \mathsf{last}(\rho_f)$.

Definition 22 (Diagnosability with an extended fault model). *An extended fault model system A is called* diagnosable *w.r.t. to a fault sequence ρ_f and a set of observable events L_o if exists $n \in \mathbb{N} : \forall\ \sigma \in \mathsf{traces}^{\rho_f^l,\mathbf{z}}(A)$ if $\alpha \in \mathsf{traces}(A) : \sigma_{L_o} = \alpha_{L_o}$ then $\rho_f \in \alpha$.*

A signature of an extended fault, in a diagnosable system A, is defined by a set of observable traces that contain the extended fault, i.e., $\mathbf{r} = \{\sigma_{L_o} \mid \rho_f \in \mathsf{traces}^{\rho_f^l,n}(A)\}$.

Theorem 1 and Theorem 2 remain true also for diagnosability with extended fault model, since definitions, theorems and proofs, obtained in Section 5, work at the level of signatures, without representing the nature of signatures explicitly.

In our example we can assume that an extended fault of the system is $\rho_f = bb$, meaning that the system execution a fault if two b events are performed. It is easy to see that the system is diagnosable for the following set of observable events: $L_o = \{a, d, c, e\}$. Applying Theorem 1, we may reduce the set to $L_o = \{a\}$. From the other side, from a set of observable events $L_o = \{d\}$ (which makes the system not diagnosable), we may expand it to a set $L_o = \{d, a\}$ using Theorem 2.

7 Conclusion

In the paper we discussed different levels of observability for diagnosable discrete-event systems. We mainly studied two approaches: first, we transform a diagnosable system into one with minimal observability and still diagnosable. Second, we transform a non-diagnosable system into diagnosable by increasing the observability of the system. We presented algorithms that implement our two approaches and we illustrated our propositions with an intuitive example through the paper. Moreover, we provided several extensions to the problem of reducing and expanding of observability in diagnosable systems. Furthermore, the provided framework deals with both classical faults and an extended fault model, as well as with other extensions in a uniform way.

In the future work we plan to further investigate various extensions to the diagnosability problem and see if our framework can be extended to deal with new types of problems. We also plan to evaluate the proposed algorithms against some real cases within the WS-Diamond project [14]. As a part of the implementation and evaluation process we want to have our framework to go distributed making it faster and more efficient. The important issue of fault isolation and control is ignored within the framework presented in the paper. In the future we plan to extend the proposed approach with controllable actions that allow us to isolate faults or, at least, perform some compensation activities to repair the system from the occurred faults.

References

[1] CIMATTI, A., PECHEUR, C., AND CAVADA, R. Formal verification of diagnosability via symbolic model checking. In *IJCAI* (2003), pp. 363–369.

[2] GENC, S., AND LAFORTUNE, S. Predictability in discrete-event systems under partial observation. In *IFAC* (Beijing, China, August 2006).

[3] JÉRON, T., MARCHAND, H., GENC, S., AND LAFORTUNE, S. Predictability of sequence patterns in discrete event systems. In *IFAC World Congress* (Seoul, Korea, July 2008).

[4] JÉRON, T., MARCHAND, H., PINCHINAT, S., AND CORDIER, M.-O. Supervision patterns in discrete event systems diagnosis. 262–268.

[5] JIANG, S., AND KUMAR, R. Failure diagnosis of discrete-event systems with linear-time temporal logic specifications. *IEEE Trans. on Automatic Control 49*, 6 (2004), 934–945.

[6] JIANG, S., KUMAR, R., AND GARCIA, H. Optimal sensor selection for discrete-event systems with partial observation. 369–381.

[7] LAZOVIK, A., AIELLO, M., AND PAPAZOGLOU, M. Planning and monitoring the execution of web service requests. *Journal on Digital Libraries* (2005).

[8] LIN, F. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems: Theory and Applications 4(2)* (May 1994), 197–212.

[9] NAU, D., GHALLAB, M., AND TRAVERSO, P. *Automated task planning. Theory and practice*. M. Kaufmann, 2004.

[10] RINTANEN, J. Diagnosers and diagnosability of succinct transition systems. In *IJCAI* (2007), pp. 538–544.

[11] SAMPATH, M., LAFORTUNE, S., AND TENEKETZIS, D. Active diagnosis of discrete-event systems. *IEEE Trans. on Automatic Control 40* (1998), 908–929.

[12] SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., SINNAMOHIDEEN, K., AND TENEKETZIS, D. Diagnosability of discrete-event systems. *IEEE Trans. on Automatic Control 9*, 40 (1995), 1555–1575.

[13] TRAVÉ-MASSUYÉS, L., CORDIER, M.-O., AND PUCEL, X. Comparing diagnosability in cs and des. In *17th Int. W-p on Principles of Diagnosis (DX'06)* (2006), pp. 55–60.

[14] WS-DIAMOND. Web services - DIAgnosability, MONitoring and Diagnosis project. http://wsdiamond.di.unito.it/.

[15] YOO, T., AND LAFORTUNE, S. Np-completeness of sensor selection problems arising in partially observed discrete-event systems. 1495–1499.