Design requirements for the diagnosability of distributed discrete event systems

Pauline Ribot, Yannick Pencolé and Michel Combacau
LAAS-CNRS, Université de Toulouse
7 avenue du Colonel Roche, F-31077 Toulouse, France
{pribot,ypencole,combacau}@laas.fr

Abstract

We address the problem of fault diagnosabilty in distributed discrete-event systems. Previous works mainly propose different ways to check whether a fault is diagnosable or not. Nowadays, due to the complexity of the engineered systems, this checking is not enough and a better feedback is required in order to redesign and guarantee the diagnosability of a fault. This paper defines the problem of the automatic computation of design requirements for the diagnosability of distributed discrete event system as a cost optimization problem.

1 Introduction

We address the problem of fault diagnosis in distributed discrete-event systems. It consists in determining the occurrences of fault events from observations and system knowledge. This problem has been studied for many years, [Sampath *et al.*, 1995], [Debouk *et al.*, 2002], [Console *et al.*, 2002], [Lamperti and Zanella, 2003]. In these previous works, the objective is to model the system and to apply monitoring algorithms on it but they all share the same weakness: none of them takes into account any diagnosablility issue about the system. If a diagnosability analysis is performed on the system, then the diagnosis algorithm is more efficient and less costly because new information is taken into account before implementing it.

In this paper, we adopt the following point of view. Due to the complexity of new large systems and the new requirements such as maintenance, reliability or security, the diagnosability study must be performed at the design stage to specify a system or to modify its specification. One difficulty in the design of a distributed system is that it is usually conceived by different designers which are in charge of only a part of the system. It follows that the integration of the different parts of the system is very complex and has a direct impact on the difficulty to guarantee the diagnosability objective of the whole system.

Our aim is to design a monitoring and diagnostic architecture for the maintenance of aeronautical systems [1]. For this purpose, we try to determine design requirements of the distributed system as modifications which could be useful for the different designers to improve and to guarantee some diagnosability objectives of the whole system. This analysis relies on both the design of the distributed system itself (design and integration of components and their communication) and the design of the monitoring architecture in charge of diagnosing the system. We propose to formalize this problem as a cost optimization problem in a distributed framework. We then show the relationship between optimizing the cost of the design of the distributed system and optimizing the choice for a suitable monitoring architecture.

The paper is organized as follows. Section 2 recalls formal background on fault diagnosis and diagnosability. Section 3 defines the optimization cost problem for guaranteeing diagnosability in a distributed discrete-event system. Section 4 introduces a methodology that determines, given a pre-specified system and a set of observations, a set of design requirements for diagnosability that takes into account the inherent characteristics of a distributed system and minimizes the implementation cost.

2 Background

2.1 Formal definition of the DES formalism

Our study takes place in a discrete event system framework for model-based diagnosis as defined in [Sampath *et al.*, 1995]. This framework has been developed for several years and is used for different application types (communication networks, business processes, middleware, ...).

Let us consider a distributed system Γ composed of interacting components $\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$ which evolve by the occurrence of events. We suppose that we already have a specification of this distributed system with an initial analysis of its behavior in the case where faults occur. This analysis is provided by designers, such as FMEA (Failure Mode Effect Analysis) data for example. The distributed system can then be modelled as a set of finite state machines (see Figure 1), each finite state machine (FSM) representing the model of a component (i.e. a local model). Depending on the context, Γ shall denote the system or its model.

Definition 1 (Local model) *A local model Γ_i is a FSM $\Gamma_i = (Q_i, \Sigma_i, T_i, q_{0_i})$ where:*

- *Q_i is a finite set of states;*

[1] Archistic project in collaboration with Airbus and Enit, France.

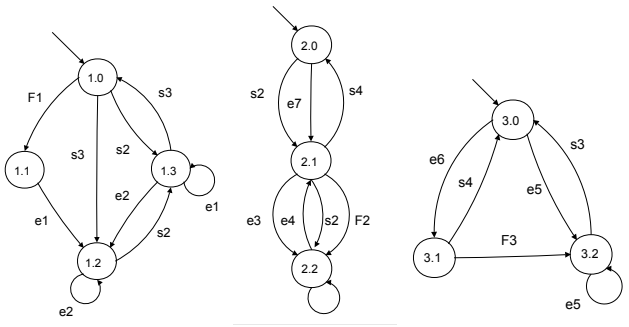

Figure 1: Models of system components : $\Gamma_1, \Gamma_2, \Gamma_3$

- Σ_i *is the set of events occurring on* Γ_i*;*
- $T_i \subseteq Q_i \times \Sigma_i \times Q_i$ *is the set of transitions;*
- q_{0_i} *is the initial state.*

Different events can occur on a component. The set Σ_i is partitioned into two disjoint sets : Σ_{l_i}, the set of events local to the component Γ_i and Σ_{c_i}, the set of interactive events which allow the communication between the different components. Local events of Σ_{l_i} can be either observable or unobservable: $\Sigma_{l_i} \subseteq \Sigma_{o_i} \cup \Sigma_{uo_i}$, where Σ_{o_i} (resp. Σ_{uo_i}) is the set of observable (resp. unobservable) events. Σ_{f_i} denotes the set of fault events occurring on the component Γ_i which are to be diagnosed. Diagnosis algorithms are based on the location of fault events on components, so we suppose that $\Sigma_{f_i} \subseteq \Sigma_{l_i}$. We authorize some interactive events to be observable : $\Sigma_{c_i} \subseteq \Sigma_{o_i} \cup \Sigma_{uo_i}$.

A subsystem γ is a non-empty set of m components of the system. The behavior of the subsystem γ is described by the prefix-closed language $L(\gamma)$ which denotes the set of all event sequences starting from the initial state. $L(\gamma) \subseteq \Sigma_\gamma^*$, where Σ_γ^* denotes the Kleene closure of the set $\Sigma_\gamma = \bigcup_{\Gamma_i \in \gamma} \Sigma_i$. In the following, we note $\Sigma_{f_\gamma} = \bigcup_{\Gamma_i \in \gamma} \Sigma_{f_i}$, $\Sigma_{c_\gamma} = \bigcup_{\Gamma_i \in \gamma} \Sigma_{c_i}$, $\Sigma_{o_\gamma} = \bigcup_{\Gamma_i \in \gamma} \Sigma_{o_i}$ and $\Sigma_{l_\gamma} = \bigcup_{\Gamma_i \in \gamma} \Sigma_{l_i}$. In the case where $\gamma = \Gamma$, previous notations are simplified as follows: $\Sigma_f = \Sigma_{f_\Gamma}$, $\Sigma_c = \Sigma_{c_\Gamma}$, $\Sigma_o = \Sigma_{o_\Gamma}$, $\Sigma_l = \Sigma_{l_\Gamma}$.

The language $L(\gamma)$ is generated by the FSM $\|\gamma\|$ obtained from the classical synchronization operation, denoted $\|$ and whose definition can be found in [Pencolé *et al.*, 2006]. The FSM $\|\gamma\|$ results from the synchronized product of m component models on the set of interactive events Σ_{c_γ} : $\|\gamma\| = \Gamma_1 \| \ldots \| \Gamma_m$. The behavior of the system is explicitly represented by $\|\Gamma\|$ (also called the global model in [Sampath *et al.*, 1995]). In the following, we will suppose that the components of the system are live and the system is deadlock-free. The consideration of problems directly linked to blocked states is not in the classical theory. New definitions and adhoc resolutions are required but they are not the topic of this paper.

The second assumption is that any model Γ_i is globally consistent. Before defining the global consistency of a local model, we need to introduce the projection operation. The projection operation can be recursively defined as follows. Let ϵ denote the empty sequence in Σ^*.

Definition 2 (Projection) *The projection operation* $P_{\Sigma'} : \Sigma^* \rightarrow \Sigma'^*$ *is such that* $P_{\Sigma'}(\epsilon) = \epsilon$ *and for all* $uv \in \Sigma^*$, $u \in \Sigma$,

$$P_{\Sigma'}(uv) = \begin{cases} uP_{\Sigma'}(v) & \text{if } u \in \Sigma' \\ P_{\Sigma'}(v) & \text{otherwise.} \end{cases}$$

Definition 3 (Global consistency) *A model* Γ_i *is globally consistent if the following condition holds:*

$$L(\Gamma_i) = P_{\Sigma_{\Gamma_i}}(L(\Gamma_0 \| \Gamma_1 \| \ldots \| \Gamma_n)).$$

A model Γ_i is globally consistent if every transition defined from any state of the model is fired in a global behavior of the system. Given any model of $\Gamma = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$, it is always possible to obtain an equivalent set of models that are globally consistent using techniques like in [Su and Wonham, 2004] for instance.

2.2 Fault diagnosis on a subsystem

The diagnosis problem is similar to the one specified in [Pencolé *et al.*, 2006]. The problem resolution requires the computation of a set of diagnosers. Let F be a fault event occurring on a component Γ_i which belongs to a subsystem γ, a diagnoser of the subsystem γ is in charge of diagnosing F. In the following, $F \in w$ denotes the occurrence of F in the event sequence w. After any sequence of observations σ emitted by γ, the diagnoser is a function that provides a diagnosis information $\Delta_\gamma(F, \sigma)$ which is one of the following three types.

- $\Delta_\gamma(F, \sigma) = F$-sure if $\forall w \in L(\gamma)$ such that $P_{\Sigma_{o_\gamma}}(w) = \sigma$, $F \in w$.
- $\Delta_\gamma(F, \sigma) = F$-safe if $\forall w \in L(\gamma)$ such that $P_{\Sigma_{o_\gamma}}(w) = \sigma$, $F \notin w$.
- $\Delta_\gamma(F, \sigma) = F$-ambiguous if $\exists w, v \in L(\gamma)$ such that $P_{\Sigma_{o_\gamma}}(w) = \sigma$ and $P_{\Sigma_{o_\gamma}}(v) = \sigma$, $F \in w$ but $F \notin v$.

The global diagnosis is given by $\{\Delta_\Gamma(F, \sigma), F \in \Sigma_f\}$.

We represent the diagnoser function as a deterministic FSM $\Delta_\gamma(F)$. This machine is built from the projection of the model $\|\gamma\|$ on the observable events in Σ_{o_γ}. Then the diagnoser describes the observable behavior of γ. The F-diagnoser $\Delta_\gamma(F)$ can be formally defined as follows.

Definition 4 (F-Diagnoser of a subsystem γ) *Given an observation set* Σ_{o_γ}*, the* F*-diagnoser* $\Delta_\gamma(F)$ *of a subsystem* γ *in charge of diagnosing a fault* F *is a deterministic finite state machine* $\Delta_\gamma(F) = (Q_{d\gamma}, \Sigma_{d\gamma}, T_{d\gamma}, q_{d\gamma_0})$ *where:*

1. $Q_{d\gamma} \subseteq 2^{Q_\gamma \times \{\{F\}, \emptyset\}}$ *is the set of states which contain the diagnosis information;*
2. $\Sigma_{d\gamma} = \Sigma_{o_\gamma}$ *is the set of events;*
3. $T_{d\gamma} \subseteq Q_{d\gamma} \times \Sigma_{d\gamma} \times Q_{d\gamma}$ *is the set of transitions;*
4. $q_{d\gamma_0} = (q_0, \emptyset)$ *is the initial state.*

The transitions of $T_{d\gamma}$ are the transitions $q_{d\gamma} \xrightarrow{e} q'_{d\gamma}$ reachable from the initial state $q_{d\gamma_0} = \{(q_0, \emptyset)\}$, with $q_{d\gamma} = \{(q_1, f_1), \dots (q_n, f_n)\}$ and $q'_{d\gamma}$ is the set $\{(q'_1, f'_1), \dots (q'_m, f'_m)\}$, such that for any $(q_i, f_i) \in q_{d\gamma}$ there exists a transition sequence $q_i \xrightarrow{uo_1} x_1 \dots x_{p-1} \xrightarrow{uo_p} x_p \xrightarrow{e} q'_j$ in the model $\|\gamma\|$ with $uo_k \in \Sigma_{uo_\gamma} \cup \Sigma_{f_\gamma}, \forall k \in \{1, \dots, p\}$, $\forall j \in \{1, \dots, m\}$ and $f' = f \cup (\{F\} \cap \{uo_1, \dots, uo_p\})$. Two examples of diagnosers are illustrated in Figure 2.

We note $Diag(q_{d\gamma})$ the diagnosis function associated to the state $q_{d\gamma}$:

1. $Diag(q_{d\gamma}) = F$-safe if $\forall (q_i, f_i) \in q_{d\gamma}$, $f_i = \{F\}$;
2. $Diag(q_{d\gamma}) = F$-sure if $\forall (q_i, f_i) \in q_{d\gamma}$, $f_i = \emptyset$;
3. $Diag(q_{d\gamma}) = F$-ambiguous otherwise.

2.3 Diagnosability

Diagnosability defines a property that measures the ability of the monitoring system to diagnose faults occurring in the supervised system. We rephrase the diagnosability definition from [Sampath *et al.*, 1995] and [Pencolé, 2005].

Definition 5 (Global diagnosability) *The fault event F is globally diagnosable in a system Γ iff*

$$\exists l \in \mathbb{N}, \forall u, v \text{ such that } |P_{\Sigma_o}(v)| \geq l \Rightarrow \Delta_\Gamma(F, \sigma) = F\text{-sure}, \quad (1)$$

where u is an event sequence in Γ from the initial state q_0 ending with the occurrence of the fault event F to a state q_f, v is an event sequence in Γ from q_f and σ is the observable sequence produced by the event sequence uv (i.e. $\sigma = P_{\Sigma_o}(uv)$).

Several tools allow the checking of diagnosability [Sampath *et al.*, 1995], [Jiang *et al.*, 2001], [Yoo and Lafortune, 2002].

3 Design for Diagnosability

The objective is to establish requirements to designers to ensure the diagnosability of the different parts of the distributed system. Two different ways can be considered to reach the objective. The first one consists in providing requirements for the design of a diagnosable system (diagnosability analysis is performed at the design stage of the system) and the second one relies on a feedback for redesigning the system and increasing the system diagnosability. Both issues are related to the problem of assistance to the design like in [Pencolé, 2005]. In this paper, we focus on the first approach.

3.1 General requirements for diagnosability

Global diagnosability implies the use of a global centralized diagnostic architecture on the whole system which may not be always feasible in practice. Moreover the global diagnosability checking requires the computation of the global model $\|\Gamma\|$ which is very complex and also not always feasible. We thus prefer to reason locally to subsystems in order to remove this limitation for large distributed systems. We want to determine subsystems that are sufficient to monitor in order to diagnose an anticipated fault event occurring on a component. We introduce the definition of local diagnosability based on the fault diagnosis on a subsystem γ (see Section 2.2). This definition can be applied to any subsystem γ.

Definition 6 (Local diagnosability) *The fault event F is locally diagnosable in a subsystem γ iff*

$$\exists l \in \mathbb{N}, \forall u, v \text{ such that } |P_{\Sigma_{o_\gamma}}(v)| \geq l \Rightarrow \Delta_\gamma(F, \sigma_\gamma) = F\text{-sure}, \quad (2)$$

where u is an event sequence in γ from the initial state q_0 ending with the occurrence of the fault event F to a state q_f, v is an event sequence in γ from q_f and σ_γ is the observable sequence produced by the event sequence uv (i.e. $\sigma_\gamma = P_{\Sigma_{o_\gamma}}(uv)$).

Global diagnosability corresponds to local diagnosability on the whole system Γ [Pencolé, 2004].

The second requirement is that the observability of the whole system has to be live (i.e. $P_{\Sigma_o}(L(\Gamma))$ is live) as explained in [Debouk *et al.*, 2002]. In practice, if we want to diagnose a fault F on a subsystem γ, we have to specify this condition locally by the observation fairness of the system. The observability of a system is *globally fair* when the observability of each component is globally fair.

Definition 7 (Observation fairness) *Observation fairness means that any observable component of the system will always emit observations after a finite number of events.*

As each observable component is live, observation fairness holds if there is no cycle of unobservable events in the component. It follows that for a subsystem γ, each event sequence in $L(\gamma)$ must always be continued by a finite event sequence which ends with an observable event of γ : $\forall w \in L(\gamma)$, $\exists p \in \mathbb{N}$ such that $\forall ww' \in L(\gamma), |w'| = p, \exists \sigma_\gamma \in \Sigma^*_{o_\gamma} \backslash \{\epsilon\}, P_{\Sigma_{o_\gamma}}(ww') = P_{\Sigma_{o_\gamma}}(w).\sigma_\gamma$.

The following property states the relationship between local and global diagnosability [Pencolé, 2004].

Property 1 *Under the assumption of observation fairness, if a fault F is locally diagnosable on a subsystem then F is globally diagnosable.*

By this property, it may be unnecessary to observe the whole system to diagnose a fault F occurring on a component of Γ. Observing only a subsystem γ can thus be sufficient.

To ensure diagnosability of the system, some modification operations have to be considered depending on the monitoring architecture. These operations consist in increasing the subsystem observability.

3.2 Types of modifications for the system design

The first operation consists in increasing the observability by selecting types, location and number of sensors. The added sensors are supposed to be reliable and without noise. This operation type tends to be similar to a sensor selection problem like in [Debouk *et al.*, 1999] or [Jiang *et al.*, 2003]. The second operation acts directly on the behavior of the subsystem. Based on the specification of a system (preexistent models), some possible operations are enumerated hereafter. All these modifications are followed by their physical signification.

1. Observing an event e (that may be a fault) which occurs in a component of the subsystem γ: $e \in \Sigma_{l_\gamma} \cap \Sigma_{o_\gamma}$. Making an event observable means to add a sensor that is able to detect the occurrence of such an event on a component in the studied subsystem.
2. Observing an event e which occurs on an other component α which interacts with γ: $e \in \Sigma_{l_\alpha} \cap \Sigma_{o_\alpha}$, where $\alpha \notin \gamma$. To observe an event on a component which interacts with the subsystem, we need to place a sensor on this component and to consider a communication protocol in the diagnostic architecture.
3. Observing an interactive event i between the subsystem γ and an other component α: $i \in \Sigma_{c_\gamma} \cap \Sigma_{c_\alpha} \cap \Sigma_{o_\gamma} \cap \Sigma_{o_\alpha}$, where $\alpha \notin \gamma$. To make an interactive event observable consists, for example, in putting a sensor on the communication bus between two components or on a middleware.
4. Observing an event $e \in \Sigma_{l_\gamma}$ only if given conditions $cond$ are verified by introducing two new events $cond{:}e \in \Sigma_{o\gamma}$ and $\neg cond{:}e \notin \Sigma_{o_\gamma}$. To observe an event in a given condition we need to have a specific sensor which can be controlled on-line depending on other information. This sensor allows an active acquisition of information like in [Thorsley and Teneketzis, 2007].
5. Adding/Deleting a transition t in the model of γ. New sensors can be generated like an alarm sensor after a given sequence of observations for example or new protocols can be implemented (communication protocol for instance). Transitions are reorganized by this new implementation and some of them may be deleted.

To each modification on the system is associated a cost which is known and listed in a modification dictionary. For example, a sensor cost depends on the type of the sensor, a pressure sensor would be more expensive that a temperature sensor. The cost of all modifications performed on the system is denoted C_D. Some modifications are infeasible. An infinite cost is associated to such modifications like for example the observation of some fault events.

3.3 Specification of the diagnostic architecture

Having sensors is not enough to ensure diagnosability. A diagnostic architecture using these sensors must be deployed upon them and have access to them. Accessing to these information resources induces a cost C_M for the monitoring system that depends on the diagnostic architecture type. The goal is to make the system diagnosable by optimizing the cost of modifications listed above (for example the cost associated to the sensor placement) but also the cost of the monitoring system (algorithm, computational resources, sensors utilization).

There mainly exist three different types of diagnostic architecture. In a centralized diagnostic architecture like in [Sampath *et al.*, 1995], all information from the components are centralized in the same place. In this architecture type, the data analysis is complex because all data are collected by the monitoring system without any processing. A lot of memory resources and communications are necessary for the monitoring system and induce the cost C_M. A centralized architecture may be not well suited for distributed systems. For large distributed systems it is more natural to adopt a decentralized or a distributed diagnostic architecture. In a decentralized diagnostic architecture, diagnosis decisions are sent from local diagnosers, which do not necessary communicate with each other. These decisions are then merged by a coordinator in order to establish a global diagnosis and to allow to make global decisions about the whole system. In a distributed architecture, there is a diagnoser per component or per subsystem that performs its own diagnosis. In order to get globally consistent diagnoses, the diagnosers need to exchange a lot of information. This communication between the diagnosers induces a bandwidth cost which is included in the cost of the monitoring algorithm C_M.

3.4 Problem formalization like a cost optimization problem

The goal is to give some requirements to the designers about how to make a system diagnosable by minimizing the total cost of operations on the system, denoted C_D, and minimizing the cost of the monitoring system algorithm C_M:

$$C_G = min \sum_{i=1}^{p} (C_{D_i} + C_{M_i}), \tag{3}$$

where p is the number of faults that may occur in the system and for which diagnosability must be guaranteed.

The challenge is to determine a trade-off between both costs (sensor placement and the cost of the associated monitoring architecture).

For example, one possible modification is to observe an event on another component in the neighbourhood. For this purpose, the monitoring system needs to use a communication protocol to get back the observation from the component. So we have to consider the cost C_M of the algorithmic procedure to communicate information to the monitoring system in addition to the cost directly associated to the sensor.

The second difficulty in the cost optimization problem is that all fault events $F \in \Sigma_f$ must be considered. Since the components of the system communicate by interactive events from Σ_c, it could be interesting to observe interactions even if this event type has a high cost because its observation could bring useful information to diagnose more than one fault event and thus reduces the monitoring cost.

4 Methodology

This section introduces a methodology that provides combining requirements and their costs for the designers. It relies on two properties : accuracy and monotony. We show how these properties can help to establish the methodology. In this section, the problem is restricted to operation of the non-active sensor placement. The only possible modification is the addition of observable events (operations from 1. to 3. in Section 3.2).

4.1 Accuracy

Independently from the diagnosability of a fault F, finding a subsystem whose diagnosis is accurate is interesting because it is a way to bound the size of the subsystem to monitor and thus the cost of the monitoring algorithm C_M. The diagnosis of a subsystem γ is said to be accurate if it is sufficient to observe it in order to provide a diagnosis which is consistent with the whole set of observations. Then the monitoring system does not require information from any other component. We are sure to have a diagnosis equivalent to the global diagnosis at any time.

Accuracy is formally defined as follows. Let $\Delta_\gamma(F)$ be the diagnoser of the subsystem γ and $\sigma_\gamma = P_{\Sigma_{o\gamma}}(\sigma)$ be the projection of the sequence $\sigma \in \Sigma_o^*$ on the observable events of γ.

Definition 8 *The diagnosis of a subsystem γ is accurate for a fault event $F \in \Sigma_{f_\gamma}$ iff the diagnoser $\Delta_\gamma(F)$ is such that*

$$\forall \sigma \in \Sigma_o^*,\ \Delta_\Gamma(F, \sigma) = \Delta_\gamma(F, \sigma_\gamma). \quad (4)$$

We can always find a subsystem whose diagnosis is accurate (diagnosis of the global system Γ is always accurate). The accuracy property is defined from the global diagnosis but there exists a sufficient local criterion to check it.

How to check accuracy

Diagnosis accuracy for a subsystem γ can be checked from a finite state machine called an interactive diagnoser. An interactive diagnoser $\Delta_\gamma^{int}(F)$ is an extension of the diagnoser $\Delta_\gamma(F)$ defined in Section 2.2. Let us suppose an observable configuration Σ_{o_γ} for the subsystem γ, the interactive diagnoser $\Delta_\gamma^{int}(F)$ results from Definition 4 (see Figure 2). The interactive diagnoser is represented by the quadruple $\Delta_\gamma^{int}(F) = (Q_{di\gamma}, \Sigma_{di\gamma}, T_{di\gamma}, q_{di\gamma_0})$, where $\Sigma_{di_\gamma} = \Sigma_{o_\gamma} \cup \Sigma_\gamma^{int}$. The set Σ_γ^{int} represents the events of γ which interact with the other components : $\Sigma_\gamma^{int} = \Sigma_{c_\gamma} \cap \Sigma_{c_{\Gamma\setminus\gamma}}$.

Like in the classical diagnoser, the states of $\Delta_\gamma^{int}(F)$ contain the diagnosis information. We recall that $Diag(q_{di})$ denotes the diagnosis information contained in the state q_{di} of $\Delta_\gamma^{int}(F)$ which is one of the following three types: F-sure, F-safe or F-ambiguous. The determination of a subsystem with an accurate diagnosis relies on a criterion introduced in [Pencolé, 2005] that we recall in Property 2. The notation $q_i \xrightarrow{w} q_j$ means that there exists an event sequence w from the state q_i which leads to the state q_j in $\Delta_\gamma^{int}(F)$. Let $\mathcal{S}(\sigma_\gamma)$ be the set of event sequences in $\Delta_\gamma^{int}(F)$ from q_{di_0} whose observable part is exactly $\sigma_\gamma \in \Sigma_{o_\gamma}^*$.

Property 2 *The diagnosis of a subsystem γ is accurate if the following criterion holds: $\forall \sigma_\gamma$, $\forall q_{di}, q'_{di} \in Q_{di\gamma}$ such as $q_{di_0} \xrightarrow{w} q_{di}$ and $q_{di_0} \xrightarrow{w'} q'_{di}$ with $w, w' \in \mathcal{S}(\sigma_\gamma)$, $Diag(q_{di}) = Diag(q'_{di})$.*

Figure 2 presents the interactive diagnoser $\Delta_\gamma^{int}(F3)$ and an accurate $F3$-diagnoser defined on the subsystem $\gamma = \{\Gamma_3\}$ for $\Sigma_{o_\gamma} = \{e5, s3\}$.

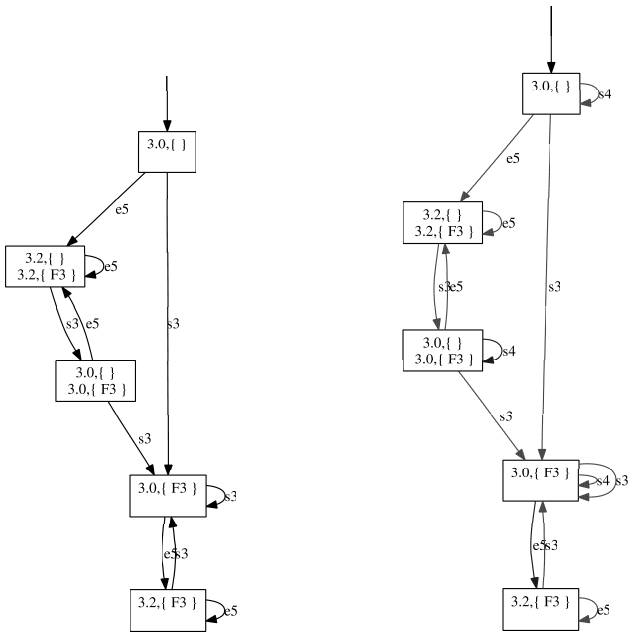

Figure 2: $\Delta_\gamma(F3)$ and $\Delta_\gamma^{int}(F3)$ on the subsystem $\gamma = \{\Gamma_3\}$ for $\Sigma_{o\gamma} = \{e5, s3\}$

Any event sequence in $\Delta_\gamma^{int}(F3)$ projected on the observable events of Σ_{o_γ} leads to states which contain the same diagnosis information, which have the same label. For example, every path from the initial state emitting the observable sequence $s3e5s3^*$ is $F3$-sure and every path emitting the sequence $e5s3e5^*$ is $F3$-ambiguous. Observations from other components could give information about the occurrence of the event $s4$ which is not observable but the diagnosis result provided by $\Delta_\gamma^{int}(F3)$ after the occurrence of $e5$ does not depend on the occurrence of $s4$. So in this case observations from other components cannot disambiguate the diagnosis.

How to make a subsystem accurate

By property 2, we notice that if all interactions of γ with other components are observable the previous criterion holds. Then a simple way to make a system accurate is to observe the interactive events [Ribot *et al.*, 2007]. Let Σ_γ^{int} be the set of events of γ that interact with the other components : $\Sigma_\gamma^{int} = \Sigma_{c_\gamma} \cap \Sigma_{c_{\Gamma\setminus\gamma}}$.

Property 3 *Let γ be a subsystem, the diagnoser $\Delta_\gamma(F)$ is F-accurate for all $F \in \Sigma_{f_\gamma}$ if $\Sigma_\gamma^{int} \subseteq \Sigma_{o_\gamma}$.*

4.2 Property monotony

The methodology consists in determining a set of modifications that guarantee observability fairness, diagnosability and accuracy. It is interesting to know if one of these properties can be preserved after the modifications realized to guarantee the other ones.

Definition 9 (Monotony) *Let $Prop$ be a boolean application ($\forall \mathbb{X}$, $Prop : P(\mathbb{X}) \mapsto \{0, 1\}$, where $P(\mathbb{X})$ represents*

the power set of $\mathbb{X}$), $Prop$ is monotonic iff

$$\forall X,\ Y,\ X \subseteq Y \subseteq \mathbb{X},\ Prop(X) \Rightarrow Prop(Y). \quad (5)$$

This definition shows that for two sets X and Y, if X is included in Y, any property which is verified by X is also verified by Y. In our study case X and Y are two sets of observable events ($X \subseteq \Sigma$, $Y \subseteq \Sigma$). The monotony of some properties like observability, normality or diagnosability has been studied in [Jiang *et al.*, 2003]. The observation fairness (Definition 7) is obviously a monotonic property. If any observable component of the system always emits an observation in a finite delay, the addition of new observations preserves this property.

Property 4 *If F is diagnosable with an observation set Σ_o satisfying the observation fairness, then F is diagnosable with an observation set $\Sigma'_o = \Sigma_o \cup \{o\}$.*

Proof: By the negation of Definition 6 (Local diagnosability), a fault F is not diagnosable for an observation set Σ'_o if there exists an infinite sequence of observations σ such that the diagnosis $\Delta_\gamma(F, \sigma)$ is F-ambiguous: there exist at least two infinite event sequences p_1 and p_2, such that F is in $p1$ and F is not in $p2$, whose projection on the observations of Σ'_o is σ. Let Σ_o be a new set of observations deprived of the event o : $\Sigma_o = \Sigma'_o \backslash \{o\}$. We prove that if F is not diagnosable with Σ'_o, then F is not diagnosable with Σ_o. If $o \notin \sigma$, we still have the infinite sequence σ such that the diagnosis is F-ambiguous. If $o \in \sigma$, as the observable configuration Σ_o satisfies the observability fairness (Definition 7), then we obtain a new infinite sequence $\sigma_{\backslash o}$ of observations deprived of o, such that the diagnosis is F-ambiguous as it is the projection of p_1 and p_2 on the observations of Σ_o. □

Property 5 *Accuracy is not a monotonic property.*

Proof: We have shown in Figure 2 that we have an accurate $F3$-diagnoser by observing $\{e5, s3\}$ on Γ_3. We consider a new observation $e6$ on the component Γ_3. Let σ_1 and σ_2 denote two global observable sequences in Σ_o^* such that $\sigma_1 = e6e7e7e5$ and $\sigma_2 = e6e7e3e5$. After observing σ_1, the global diagnosis is F-safe and after σ_2, the global diagnosis is F-sure. The projection of both event sequences σ_1 and σ_2 on events of Γ_3 is $e6e5$. Locally, σ_1 and σ_2 are not distinguishable and the local diagnosis after observing $e6e5$ is F-ambiguous. The local diagnosis is not equivalent to the global diagnosis, then the diagnosis of $F3$ is not accurate anymore by considering the observation $e6$ on Γ_3. □

Accuracy is a non monotonic property but it can be preserved in the case where all interactions of the subsystem γ with the other components are observable. In this specific case described by Property 2, accuracy still holds after the addition of a new observation.

Diagnosability property is always preserved by considering new observations whereas accuracy property is preserved only in some specific cases. The monotony can help with the sensor choice by determining some selection criteria in order to preserve diagnosability and accuracy properties.

4.3 Algorithm

This section presents an algorithm that determines a subsystem and proposes some modifications on it in order to make a fault occurring on one of its component with an accurate diagnosis and minimal costs. We consider a system of n components with an initial minimal observable configuration Σ_o. The system may be specified with an initial set of sensors which are already available. The initial observable configuration may be empty (no sensors already placed on the system), this case is illustrated by an example in 4.4.

Algorithm 1 selects a subsystem for which the total cost C_G is minimal (between the supervision cost C_M, the cost C_A to make it accurate and the cost C_D to make it diagnosable for the fault F in component Γ_i). It also returns the requirements Req as a set of modifications to perform on the subsystem. In this study, we consider modifications from the sensor placement (i.e. operations from types 1., 2. or 3. of Section 3.2). Since diagnosability is a monotonic property, we look for an observable configuration that makes the system diagnosable before considering accuracy. If the subsystem is diagnosable, the diagnosability property will be preserved by adding observable events to get accuracy. If the subsystem is not diagnosable, the optimization is performed simultaneously on both costs : C_A et C_D.

In the proposed algorithm, the expression $subsystem(\Gamma_i,\ k)$ denotes the set of subsystems composed of k components and containing Γ_i. We first need to determine if there is a solution to the sensor placement problem. The function $SolutionExistence(\gamma, F)$ relies on the Property 6.

Property 6 *If the fault event F occurring on a component of the subsystem γ is not diagnosable in γ by considering all events (except the fault events) as observable, then there is no solution for the sensor placement problem.*

If all events are observable and F is not diagnosable, no event from other components which interacts with γ will provide more information to make F diagnosable. The only way to make F diagnosable in the component is then to redesign it, by considering operations of type 4. or 5. (see Section 3.2) which is out of the topic of this methodology, in order to have F diagnosable by considering the new set of all observations.

The function $Monitoring$ induces a cost C_M for the monitoring of a subsystem depending on the implementation of the diagnosis architecture as explained in 3.3. The functions $CheckingAccuracy$ and $CheckingDiagnosability$ are boolean. The first function applies the criterion of Property 2 to determine if the considered subsystem is accurate or not and the second one uses an algorithm to check the subsystem diagnosability for a fault F as in [Jiang *et al.*, 2001], [Pencolé, 2004]. These algorithms are polynomial in the number of states in $\|\gamma\|$. The functions $MakeDiagnosable$ and $MakeAccurate$ use modifications from types 1., 2. or 3. of Section 3.2 and techniques from sensor placement as in [Jiang *et al.*, 2003] to obtain a diagnosable subsystem γ. The

Algorithm 1 Requirements and costs for diagnosability

1: *Input:* $F \in \Sigma_{f_i}, \Gamma = \{\Gamma_1, \ldots, \Gamma_n\}, \Sigma_o$
2: $C_G^0 \leftarrow \infty$; $k \leftarrow 1$; $Req = \emptyset$
3: **repeat**
4: $C_G^k \leftarrow \infty$
5: **for all** $\gamma \in subsystem(\Gamma_i,\ k)$ **do**
6: **if** $\neg SolutionExistence(\gamma, F)$ **then**
7: Go step 24
8: **end if**
9: $C_M \leftarrow Monitoring(\gamma)$; $C_D = 0$; $C_A = 0$;
10: **if** $CheckDiagnosability(\gamma, F)$ **then**
11: **if** $\neg CheckAccuracy(\gamma)$ **then**
12: $(Req, C_A) \leftarrow MakeAccurate(\gamma)$
13: **end if**
14: **else**
15: $(Req, C_A, C_D) \leftarrow MakeDiagnosable(\gamma, F) \wedge MakeAccurate(\gamma)$
16: **end if**
17: $C_G^\gamma \leftarrow C_M + C_A + C_D$
18: $C_G^k \leftarrow \min(C_G^k, C_G^\gamma)$
19: **end for**
20: $k \leftarrow k + 1$
21: **until** $(C_G^{k-1} \geq C_G^{k-2}) \wedge (k \geq 2) \wedge (k \leq n)$
22: $C_G \leftarrow C_G^{k-2}$
23: Go to step 25
24: *Output:* No solution
25: *Output:* γ; C_G; Req

function $MakeAccurate$ could also rely on Property 3. It is a simple mean to obtain an accurate subsystem. The chosen configurations of observations have to guarantee some properties. The observable configurations required to make a fault F diagnosable with an accurate diagnoser has to guarantee that properties obtained for diagnosing another fault F' are not lost, hence the importance of the property monotony.

4.4 Illustrative example

We develop an example for the specification of the system illustrated in Figure 1.

We analyze the diagnosability of the fault $F1$ on the component Γ_1. The initial observable configuration is empty: $\Sigma_o = \emptyset$. We check that a solution for the sensor placement problem exists by considering all events (except the fault $F1$) as observable in the model Γ_1. Then we determine the possible observable configurations to such that $F1$ is diagnosable and $\Delta_{\Gamma_1}(F)$ is accurate. We prefer the minimal one: $\Sigma_{o_1} = \{e1, s2, s3\}$. The global cost $C_G^{\Gamma_1}$ includes the cost of these observations, $(C_D + C_A)$, and the monitoring cost associated to the component Γ_1, C_M. We now consider the subsystems of two components containing Γ_1 and another component interacting with Γ_1 : $\|\gamma_1\| = \Gamma_1 \| \Gamma_2$ and $\|\gamma_2\| = \Gamma_1 \| \Gamma_3$. In γ_1, $F1$ is diagnosable with an accurate diagnoser for the same observable configuration: $\Sigma_{o_1} = \{e1, s2, s3\}$. But the monitoring cost for two components is obviously higher than for one component. So the global cost $C_G^{\gamma_1}$ for this subsystem is greater than $C_G^{\Gamma_1}$. Following the same reasoning for the subsystem γ_2, we find that the global cost $C_G^{\gamma_2}$ is also greater than $C_G^{\Gamma_1}$. Then we prefer to consider Γ_1 only with the observable configuration $\Sigma_{o_1} = \{e1, s2, s3\}$ and a cost $C_G^{\Gamma_1}$. Here $Req = \{e1, s2, s3\}$ because the observable configuration was empty before starting the algorithm. We could have obtained an observable configuration guaranteeing objectives with a lower global cost by considering the subsystem γ_1 and γ_2, if the new cost $(C_D + C_A)$ for these subsystems were smaller than the difference between the monitoring cost for component Γ_1 and the monitoring cost for γ_1 or γ_2.

Then we study the diagnosability of the fault $F2$ in the component Γ_2. The initial observable configuration is not empty anymore: $\Sigma_o = \{e1, s2, s3\}$. We first check the solution existence for the sensor placement problem by considering all events (except the fault $F2$) as observable in the model of Γ_2. There is no way to make $F2$ diagnosable in Γ_2 by techniques from the sensor placement. The only solution is to modify the structure of Γ_2.

The initial observable configuration is still $\Sigma_o = \{e1, s2, s3\}$ for the diagnosability analysis of $F3$ in Γ_3, because no more observations are brought by analysis of $F2$ (no solution for diagnosing $F2$). There exists a solution for the sensor placement problem in Γ_3, so we determine the observable configurations to make $F3$ diagnosable with an accurate diagnoser : $\{e5, e6, s4\} \in \Sigma_o$. The global cost $C_G^{\Gamma_3}$ includes the cost of the observations and the monitoring cost associated to the component Γ_3. We now consider the subsystems of two components containing Γ_3 and another component interacting with Γ_3: $\|\gamma_1\| = \Gamma_3 \| \Gamma_1$ and $\|\gamma_2\| = \Gamma_3 \| \Gamma_2$. In the subsystems γ_1 and γ_2, $F3$ is diagnosable with the same observable configuration as in the component Γ_3. The monitoring cost for two components is higher than for one component, so the global costs associated to these subsystems are higher than the previous one. The fault $F3$ is diagnosable in Γ_3 with $Req = \{e5, e6, s4\}$ and a global cost $C_G^{\Gamma_3}$.

Finally, the distributed system is diagnosable for the faults $F1$ and $F3$ with the observable configuration $\Sigma_o = \{e1, e5, e6, s2, s3, s4\}$ and a diagnostic architecture composed of two diagnosers : one on the component Γ_1 for diagnosing $F1$ and one on the component Γ_3 for diagnosing $F3$. Even if $F2$ cannot be diagnosable in the system, the diagnosis of $F2$ is accurate with the chosen observable configuration (all interactive events of Γ_2 are observable) so we could place another diagnoser on the component Γ_2.

5 Related Work

In the literature, there are many works about fault diagnosis and diagnosability on discrete-event systems. Our framework is based on the classical framework defined in [Sampath *et al.*, 1995]. The problem of checking diagnosability has been studied for many years and several algorithms have been proposed for centralized monitoring architectures [Jiang *et al.*, 2001], [Yoo and Lafortune, 2002]. Our proposal relies on the diagnosability of distributed discrete event systems that has been defined in [Pencolé, 2004]. None of these works propose any design feedback to increase the system diagnosability.

A possible way to provide design requirements for diagnosability is to solve the sensor placement problem. In the

sensor placement problem, it is assumed that there always exist modifications of type 1 (see section 3.1) by adding sensors that guarantee the diagnosability of the whole system. Moreover, the cost of the monitoring architecture is supposed to be negligible. In [Debouk *et al.*, 1999], the problem is to select an optimal subset of available sensors by inferring a set of sensor tests. This inference assumes that the cost of $n + 1$ sensors is always greater than the cost of n sensors (i.e. optimal = minimal) which may be a restrictive assumption from an economical point of view. [Jiang *et al.*, 2003] proposes a methodology to select a minimal set of sensors to preserve properties like normality, diagnosability and proves that this problem is NP-hard. In [Narasimhan *et al.*, 1998], the problem is defined over a qualitative model and in [Torta and Torasso, 2007] it is extended by parametrizing some discriminability relations of the system. Here also, optimality is equivalent to minimality. In [Spanache *et al.*, 2004], the selection is performed by a genetic algorithm to make a continuous system more diagnosable. The cost of the sensor takes into account the economical issue.

To our best knowledge, there is no work about improving diagnosability in a distributed framework. In [Pencolé, 2005], the question of improving diagnosabilty in a distributed discrete event systems has been introduced. The idea is to detect local undiagnosable scenarios that can be used to provide design requirements for the elimination of such scenarios.

6 Conclusion and Perspectives

This paper defines a framework based on a classical model-based diagnosis formalism in order to extend automatic diagnosability analyses and provide design requirements for a distributed dynamic system. We propose to define the problem as a cost optimization problem where not only the costs about the design of the system but also the costs about the monitoring architecture are taken into account in order to minimize the integration costs of the distributed system. Finally, we claim that the design requirements for diagnosability are closely related to the design requirements for accuracy as this property is a way to isolate a subsystem above which it is possible to design a monitoring architecture that provides a diagnosis as accurate as possible. In this paper, we have presented an algorithm which selects a subsystem for which the total cost C_G is minimal. The solution is not unique but the algorithm can be extended to provide a set of possible subsystems which have a minimal cost. Our perspectives are to develop this methodology in detail by integrating the different methods (diagnosability checkers, sensor placement selectors) and automatically provide design requirements for distributed systems. Then we would like to apply this work in the Archistic project framework for the maintenance of aeronautical systems.

References

[Console *et al.*, 2002] L. Console, C. Picardi, and M. Ribaudo. Process algebra for systems diagnosis. *Artificial Intelligence*, 142:19–51, 2002.

[Debouk *et al.*, 1999] R. Debouk, S. Lafortune, and D. Teneketzis. On an optimization problem in sensor selection for failure diagnosis. In *38th Conference on Decision and Control*, pages 4990–4995, 1999.

[Debouk *et al.*, 2002] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems*, 10(1–2):33–86, 2002.

[Jiang *et al.*, 2001] 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.

[Jiang *et al.*, 2003] S. Jiang, R. Kumar, and H. E. Garcia. Optimal sensor selection for discrete event systems under partial observation. *IEEE Transactions on Automatic Control*, 48:369–381, 2003.

[Lamperti and Zanella, 2003] G. Lamperti and M. Zanella. *Diagnosis of active systems*. Kluwer Academic Publishers, 2003.

[Narasimhan *et al.*, 1998] S. Narasimhan, P.J. Mosterman, and G. Biswas. A systematic analysis of measurement selection algorithms for fault isolation in dynamic systems. In *9th International Workshop on Principles of Diagnosis (DX'98)*, 1998.

[Pencolé *et al.*, 2006] Y. Pencolé, D. Kamenetsky, and A. Schumann. Towards low-cost diagnosis of component-based systems. In *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process*, 2006.

[Pencolé, 2004] Y. Pencolé. Diagnosability analysis of distributed event systems. In *European Conference on Artificial Intelligence (ECAI'04)*, 2004.

[Pencolé, 2005] Y. Pencolé. Assistance for the design of a diagnosable component-based system. In *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, pages 549–556, 2005.

[Ribot *et al.*, 2007] P. Ribot, Y. Pencolé, and M. Combacau. Characterization of requirements and costs for the diagnosability of distributed discrete event systems. In *5th Workshop on Advanced Control and Diagnosis (ACD'07)*, 2007.

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

[Spanache *et al.*, 2004] S. Spanache, T. Escobet, and L. Travé-Massuyès. Sensor placement optimisation using genetic algorithms. In *15th International Workshop on Principles of Diagnosis (DX'04)*, pages 179–184, 2004.

[Su and Wonham, 2004] R. Su and W.M. Wonham. Distributed diagnosis under global consistency. In *43 rd IEEE Conference on Decision and Control*, pages 525–530, 2004.

[Thorsley and Teneketzis, 2007] D. Thorsley and D. Teneketzis. Active acquisition of information for diagnosis and supervisory control of discrete event systems. *Discrete Event Dynamic Systems*, 17(4):531–583, 2007.

[Torta and Torasso, 2007] G. Torta and P. Torasso. Computation of minimal sensor sets from precompiled disciminability relations. In *18th International Worshop on Principles of Diagnosis (DX'07)*, pages 202–209, 2007.

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