

# Incremental decentralized diagnosis approach for the supervision of a telecommunication network

**Yannick Pencolé**  
IRISA, 35042 Rennes, FRANCE  
ypencole@irisa.fr

**Marie-Odile Cordier**  
IRISA, 35042 Rennes, FRANCE  
cordier@irisa.fr

**Laurence Rozé**  
IRISA, 35042 Rennes, FRANCE  
roze@irisa.fr

## Abstract

We address the problem of failure diagnosis in discrete-event systems such as telecommunication networks. We propose to extend the decentralized diagnosis approach proposed in (Pencolé 2000) in order to use it in an incremental way. The incremental approach is needed in order to provide on-line diagnosis and assist supervision operators. The difficulties about the incremental decentralized diagnosis approach are analyzed. Two solutions are proposed and discussed. This incremental approach has been experimented on telecommunication networks.

## Introduction

The problem we deal with is the supervision of complex and large discrete-event systems such as telecommunication networks. Our purpose is to help operators of such systems to diagnose failures in the system according to observed events (alarms).

In the literature, a few failure diagnosis approaches have been developed for discrete-event systems. We can divide these approaches in two types.

The first type of approaches needs a centralized information about the system to diagnose. (Sampath *et al.* 1998) has proposed the diagnoser approach which consists in the compilation of diagnostic information in a data structure (called *diagnoser*) which maps failures and observations for on-line diagnosis. As telecommunication networks are concerned, (Rozé & Cordier 1998) proposes an extension of this approach which is well-updated for the on-line diagnosis. Nevertheless, a centralized approach needs to have a global information about the system which is unrealistic due to the size of the systems we are faced with.

For complex and large systems like telecommunication networks, the impossibility to use a centralized information obliges to use a decentralized information. Moreover, such systems are naturally distributed so it is easier to model those systems with decentralized information. In this way, (Debouk, Lafortune, & Teneketzi 2000) proposes an approach for diagnosing discrete-event systems using decentralized and coordinated diagnosers. But in this approach, the computation of one

decentralized diagnoser needs a global information that we cannot have with our applications. (Baroni *et al.* 1999) and (Console, Picardi, & Ribaud 2000) propose methods based on a model-simulation approach which only needs a decentralized model, but these methods are used off-line to solve a diagnosis problem *a posteriori*.

Our motivation is to propose an approach which only needs decentralized information and can provide on-line diagnosis of a large discrete-event system (telecommunication network). In (Pencolé 2000), we have already defined a decentralized diagnosis approach that respects the constraints we are faced with. Nevertheless, in order to use this approach on-line, incrementality becomes a crucial issue in order to efficiently update already computed diagnoses by taking into account new observations received by the supervision center.

The paper is organized as follows. We first describe a simple example of a telecommunication network that will be used as a running example throughout the paper. We then recall the decentralized approach by defining a decentralized model and the different notions of diagnosis relying on the decentralized approach. In a new section, the difficulties about the incremental decentralized approach are analyzed. Two solutions are examined. The first one consists in carefully selecting the breakpoints which determine the temporal windows on which successive diagnoses are computed. The second one consists in completing the flow of observations received on a given temporal window. In both cases, it is then possible to use an easy and efficient algorithm based on the concatenation of diagnoses corresponding to successive temporal windows. This incremental approach has been experimented on telecommunication networks.

## Running example

In this section, we introduce a very simple telecommunication system (see figure 1) that we shall use as a running example throughout the paper. It is formed by two switches (*SW1* and *SW2*) which send and receive data, a control station *CS* which is in charge of managing the switches and a supervision center *SC* which is in charge of monitoring the system by receiving alarms

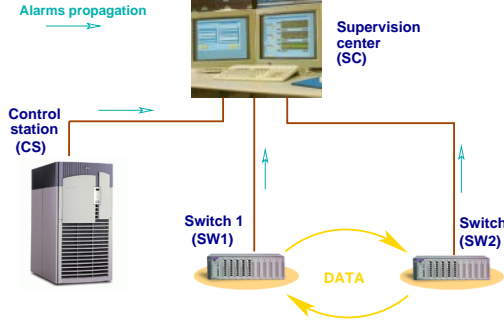


Figure 1: Running example of a simplified telecommunication network.

from  $SW1$ ,  $SW2$  and  $CS$ . For reasons of simplicity of the example, we assume that the failures only occur on the components; the connections between the components are considered as safe. A failure is defined by two events: the beginning of the failure and the end of the failure. In the example, we consider that the failures are repaired without an operator intervention.

$SW1$  and  $SW2$  can have two kinds of failures: they can boot or be blocked. Those failures are defined by their start-event ( $SWi_{blk}$ ,  $SWi_{bt}$  where  $i = \{1, 2\}$ ) and their end-event ( $SWi_{back}$ ,  $SWi_{endbt}$ ). When the switch  $SWi$  begins to block, it emits an alarm  $SWi_{stop}$ . Concerning the booting, the behaviors of the switches are different: when  $SW1$  begins to boot then it emits  $SW1_{stop}$ ; when  $SW2$  begins to boot then  $SW2$  emits  $SW2_{boot}$ . When the switch  $SWi$  begins to work well again, it emits the alarm  $SWi_{run}$ . Nevertheless, when  $SWi$  is blocked, then it does not emit any alarm when it begins to boot.

As  $SW1$  and  $SW2$ ,  $CS$  has two kinds of failures (blocked and booting) defined by the events ( $CS_{blk}$ ,  $CS_{back}$ ,  $CS_{bt}$ ,  $CS_{endbt}$ ). When  $CS$  begins to block, it emits  $CS_{stop}$ . When  $CS$  begins to boot, it also emits a  $CS_{stop}$  alarm and sends to  $SWi$  a message  $SWi_{bt}$  for the rebooting of the  $SWi$  switches. Nevertheless, when  $CS$  begins to boot whereas  $CS$  is blocked, it only sends a message to the  $SWi$  switches (no alarm is sent). When  $CS$  begins to work well again, it emits the alarm  $CS_{run}$ .

## Overview of the decentralized approach Model in a decentralized approach

The systems we consider are distributed systems composed of components which interact each other. As said before, dealing with a global model of such systems is unrealistic. This section explains how the model of the system is described in a decentralized way by means of local models, which describe the behaviors of each component of the system and the interactions between them. The formalism chosen for such models is that of communicating automata.

Each component can react to exogenous events such

as failures by changing of states and emitting observable events. The components interact by exchanging messages (named *internal events*) which occur for instance when failures propagate through the system. We make the hypothesis that no delays exist on the messages exchanged by the components.

A *component* is faced to two kinds of received events: exogenous events ( $\Sigma_{exo}^i$ ) such as failure events and internal events ( $\Sigma_{intreceived}^i$ ). A component emits two kinds of events: observable events via its communication channel ( $\Sigma_{obs}^i$ ) and internal events ( $\Sigma_{intemitted}^i$ ).

**Definition 1 (Model of a component)** A component behavior is described by a communicating finite-state machine  $\Gamma_i = (\Sigma_{in}^i, 2^{\Sigma_{out}^i}, Q_i, E_i)$  where

- $\Sigma_{in}^i$  is the set of input events ( $\Sigma_{in}^i = \Sigma_{exo}^i \cup \Sigma_{intreceived}^i$ );
- $\Sigma_{out}^i$  is the set of output events ( $\Sigma_{out}^i = \Sigma_{obs}^i \cup \Sigma_{intemitted}^i$ );
- $Q_i$  is the set of states of the component;
- $E_i \subseteq (Q_i \times \Sigma_{in}^i \times 2^{\Sigma_{out}^i} \times Q_i)$  is the set of transitions.

The model of the system is described in a decentralized way by the models of its components. Note that this model could be explicitly built by composing (via a classical operation of synchronization on the internal events) the automata of its components but it is exactly what we want to avoid due to the intractable size of such a model for large systems.

**Definition 2 (Model of a system)** The model  $\Gamma$  of a system is given by the set of models of its components  $\{\Gamma_1, \dots, \Gamma_n\}$ , a set of exogenous events ( $\Sigma_{exo}$ ), a set of observable events ( $\Sigma_{obs}$ ) and a set of internal events ( $\Sigma_{int}$ ) such that:

- $\{\Sigma_{obs}^1, \dots, \Sigma_{obs}^n\}$  is a partition of  $\Sigma_{obs}$ ;
- $\{\Sigma_{exo}^1, \dots, \Sigma_{exo}^n\}$  is a partition of  $\Sigma_{exo}$ ;
- $\{\Sigma_{intreceived}^1, \dots, \Sigma_{intreceived}^n\}$  and  $\{\Sigma_{intemitted}^1, \dots, \Sigma_{intemitted}^n\}$  are partitions of  $\Sigma_{int}$ ;
- $\forall e \in \Sigma_{int}, \exists! \Gamma_i | e \in \Sigma_{intreceived}^i \wedge \exists! \Gamma_j | e \in \Sigma_{intemitted}^j \wedge i \neq j$ .

We present in the figure 2, the model of the system described in the previous section. The observable events<sup>1</sup> are ( $CS_{stop}$ ,  $CS_{run}$ ,  $SW1_{stop}$ ,  $SW1_{run}$ ,  $SW2_{stop}$ ,  $SW2_{run}$ ,  $SW2_{boot}$ ). The exogenous events are the failure events which can occur on the system ( $CS_{blk}$ ,  $CS_{back}$ ,  $CS_{bt}$ ,  $CS_{endbt}$ ,  $SW1_{blk}$ ,  $SW1_{back}$ ,  $SW1_{endbt}$ ,  $SW2_{blk}$ ,  $SW2_{back}$ ,  $SW2_{endbt}$ ). The set of internal events permits to model the propagation of the booting from the control station to the switches ( $SW1_{bt}$ ,  $SW2_{bt}$ ).

<sup>1</sup>We suppose the existence of local sensors, one for each component. We will come back on this point in the paragraph defining a global diagnosis.

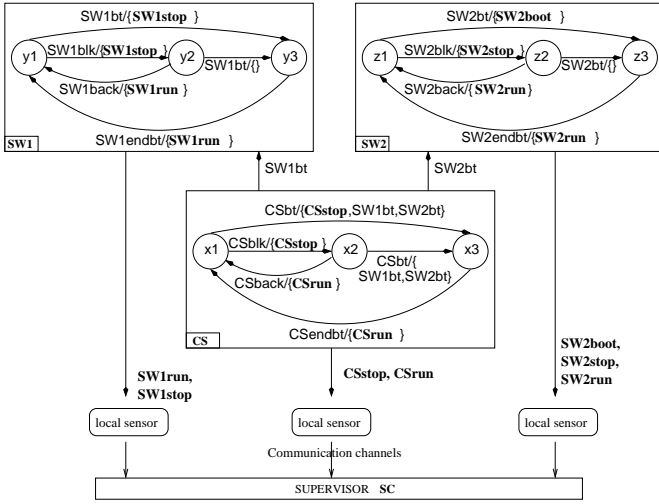


Figure 2: Model of the system.

### Diagnosis in a decentralized approach

The idea is to use a decentralized diagnosis approach by firstly computing a diagnosis for each component (*local diagnosis*) and then building a diagnosis of the whole system (*global diagnosis*) from these local diagnoses. In the following, we formally define these notions and explain how the global diagnosis is computed by *composing* the local ones.

**Local diagnosis** Let denote  $\mathcal{O}_{\Gamma_i}$  the sequence of local observations, i.e the events locally observed by a sensor plugged in the component  $\Gamma_i$ . We have  $\mathcal{O}_{\Gamma_i} \in (\Sigma_{obs}^i)^*$ . We suppose that, at the beginning of the task, the component is in one of the initial states  $X_{init}^{\Gamma_i}$ .

Given the model of the component  $\Gamma_i$ , a *local diagnosis*  $\Delta_{\Gamma_i}$  describes the subset of trajectories from  $\Gamma_i$  starting from elements of  $X_{init}^{\Gamma_i}$  which explains the sequence of *local* observations  $\mathcal{O}_{\Gamma_i}$ , i.e such that their projections on observable events correspond exactly to  $\mathcal{O}_{\Gamma_i}$ . We propose to represent a local diagnosis as a communicating finite-state machine  $\Delta_{\Gamma_i}(X_{init}^{\Gamma_i}, \mathcal{O}_{\Gamma_i})$ , shortly  $\Delta_{\Gamma_i}$ . Compared to the automaton  $\Gamma_i$ , the main syntactical difference is that each state  $Q_{\Delta_i}$  of this automaton is associated to a pair  $(s_{\Gamma_i}, \mathcal{E}_i)$  where  $s_{\Gamma_i} \in Q_i$  is a state of the component and  $\mathcal{E}_i$  is the prefix subsequence of  $\mathcal{O}_{\Gamma_i}$  explained in this state. The initial states of  $\Delta_{\Gamma_i}$  are those corresponding to  $X_{init}^{\Gamma_i}$ . The final states are those such that  $\mathcal{E}_i = \mathcal{O}_{\Gamma_i}$ . They represent the states of the component explaining the whole sequence of local observations. As in  $\Gamma_i$ , the transitions are labeled with exogenous or internal received events as input and with observed or internal emitted events as output.

**Definition 3 (Local diagnosis)** The local diagnosis  $\Delta_{\Gamma_i}(X_{init}^{\Gamma_i}, \mathcal{O}_{\Gamma_i})$  of  $\Gamma_i$  according to the sequence  $\mathcal{O}_{\Gamma_i}$  is a finite-state machine:  $(\Sigma_{in}^i, 2^{\Sigma_{out}^i}, Q_{\Delta_i}, E_{\Delta_i})$  where

- $\Sigma_{in}^i$  is the set of input events ( $\Sigma_{in}^i = \Sigma_{exo}^i \cup$

$\Sigma_{intreceived}^i$ );

- $\Sigma_{out}^i$  is the set of output events ( $\Sigma_{out}^i = \Sigma_{obs}^i \cup \Sigma_{intemitted}^i$ );
- $Q_{\Delta_i}$  is the set of states ( $Q_{\Delta_i} \subseteq Q_i \times (\Sigma_{obs}^i)^*$ );
- $E_{\Delta_i} \subseteq (Q_{\Delta_i} \times \Sigma_{in}^i \times 2^{\Sigma_{out}^i} \times Q_{\Delta_i})$  is the set of transitions.

Figure 3 gives the local diagnosis of the control station when  $X_{init}^{CS} = \{x1\}$  and  $\mathcal{O}_{CS} = [CSstop]$ .

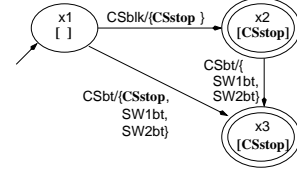


Figure 3: Local diagnosis:  $\Delta_{CS}(\{x1\}, [CSstop])$ .

The way a local diagnosis is computed is not the subject of this paper. In our case, it is done by using a diagnoser approach as detailed in (Pencolé 2000).

**Global diagnosis** We have defined local diagnoses as finite-state machines which represent the possible behaviors of components according to local observations. Let us now define a global diagnosis and explain how it is computed from local diagnoses.

The global observation  $\mathcal{O}$  corresponds to what is observed by the supervisor which collects all the sequences of observable events (alarms) sent by each of the components through their own communication channels. It is described by  $\{\mathcal{O}|\Sigma_{obs}^1, \dots, \mathcal{O}|\Sigma_{obs}^n\}$  where  $\mathcal{O}|\Sigma_{obs}^i \in (\Sigma_{Obs}^i)^*$  is the sequence of observations received from  $\Gamma_i$ .

Given a set of initial states  $X_{init}^{\Gamma}$ , a global diagnosis describes all the trajectories starting from elements of  $X_{init}^{\Gamma}$  which explain the global observation  $\mathcal{O}$ , i.e such that their projections on the observable events correspond to  $\mathcal{O}$ .

The global diagnosis is a communicating finite-state machine  $\Delta_{\Gamma}(X_{init}^{\Gamma}, \mathcal{O})$ , shortly  $\Delta_{\Gamma}$ . The states  $Q_{\Delta}$  of this automaton are pairs  $(s_{\Gamma}, \mathcal{E})$  where  $s_{\Gamma} \in Q_1 \times \dots \times Q_n$  is a state of the system and  $\mathcal{E} = \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$  where  $\mathcal{E}_i$  is the prefix of the sequence  $\mathcal{O}|\Sigma_{obs}^i$  explained in this state. The initial states of  $\Delta_{\Gamma}$  are those corresponding to  $X_{init}^{\Gamma}$  and the final states are such that  $\mathcal{E} = \mathcal{O}$ , i.e  $\forall i \mathcal{E}_i = \mathcal{O}|\Sigma_{obs}^i$ . So the final states represent the states of the components explaining the whole sequence of observations. The transitions are labeled with exogenous events as input ( $\Sigma_{exo}$ ) and with observed events as output ( $\Sigma_{obs}$ ).

**Definition 4 (Global diagnosis)** The global diagnosis  $\Delta_{\Gamma}(X_{init}^{\Gamma}, \mathcal{O})$  is a finite-state machine:  $(\Sigma_{in}, 2^{\Sigma_{out}}, Q_{\Delta}, E_{\Delta})$  where

- $\Sigma_{in}$  is the set of input events ( $\Sigma_{in} = \Sigma_{exo}$ );
- $\Sigma_{out}$  is the set of output events ( $\Sigma_{out} = \Sigma_{obs}$ );

- $Q_\Delta$  is the set of states of the diagnosis;
- $E_\Delta \subseteq (Q_\Delta \times \Sigma_{in} \times 2^{\Sigma_{out}} \times Q_\Delta)$  is the set of transitions.

**Computing global diag from local diags** In a decentralized approach, the global model of the system is not explicitly built but defined as the set of its component models. Consequently, instead of computing directly the global diagnosis from the global model, the idea is to compute the global diagnosis from the local diagnoses. As local diagnoses are represented by automata, the global diagnosis is built by composing the local diagnoses.

Let us suppose that the sequence of observations received by the supervisor from each component  $\Gamma_i$  ( $\mathcal{O}|\Sigma_{obs}^i$ ) corresponds exactly to the sequence of local observations emitted by  $\Gamma_i$  ( $\mathcal{O}_{\Gamma_i}$ ). We have then  $\mathcal{O}|\Sigma_{obs}^i = \mathcal{O}_{\Gamma_i}$ . This property will be referred as *property 1* in the following.

To compute the global diagnosis, we use the following equation where  $\odot$  is the classical composition operation between two communicating finite-state machines synchronized on the internal events exchanged between the local diagnoses:

$$\Delta_\Gamma(X_{init}^\Gamma, \mathcal{O}) = \bigodot_{i=1}^n \Delta_{\Gamma_i}(X_{init}^{\Gamma_i}, \mathcal{O}|\Sigma_{obs}^i) \quad (1)$$

Figure 4 gives the global diagnosis of the model shown in figure 2 for the initial state  $X_{init}^\Gamma = \{(x1, y1, z1)\}$  and the global observation  $\mathcal{O} = \{[CSstop], [SW1stop], []\}$ . It was obtained by the operation:  $\Delta_{CS}(\{x1\}, [CSstop]) \odot \Delta_{SW1}(\{y1\}, [SW1stop]) \odot \Delta_{SW2}(\{z1\}, [])$ .

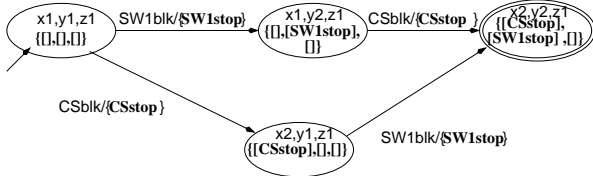


Figure 4: Global diagnosis of the system:  $\Delta_\Gamma(\{(x1, y1, z1)\}, \{[CSstop], [SW1stop], []\})$ .

**Local observations / global observation** The local observations were defined as observable events observed by local sensors, i.e sensors associated to each of the components. The global observation corresponds to what is observed by a supervisor which collects all the observations emitted by the components. In the previous paragraph, we make the supposition (property 1) that what is received by the supervisor corresponds exactly to what is emitted by each component ( $\mathcal{O}|\Sigma_{obs}^i = \mathcal{O}_{\Gamma_i}$ ).

This property is clearly satisfied when the local sensors are directly observable, or when the messages they sent are received without delay to the supervisor. In

most of the cases however, such local sensors are not directly observable, and the messages are sent to the supervisor via communication channels. The respective order in which the events sent by each component are received by the supervisor depends on the characteristics of the communication channels. In the following, we make the hypothesis that the communication channels behave as FIFO files. We have then :

**Hypothesis 1** Let  $o_1$  and  $o_2$  be observable events emitted by one component. We assume that  $o_1$  and  $o_2$  are received by the supervisor in the order of their emission by the component.

With this hypothesis on the communication channels, the property 1 relating local observations and global observation holds ( $\mathcal{O}|\Sigma_{obs}^i = \mathcal{O}_{\Gamma_i}$ ) and the global diagnosis can be built as explained before (equation 1).

An important point to remark is that property 1 holds because we consider that the set of observations received by the supervisor is complete, i.e all observable events sent by the components have been received by the supervisor (the communication channels are empty).

In the next section, we turn to the problem of the on-line computing of a diagnosis and we show that property 1 is a central issue to preserve the correctness and the efficiency of an incremental algorithm.

## Incremental diagnosis

In the on-line diagnostic approaches, the observations are considered on successive temporal windows. Having computed a global diagnosis for a given temporal window, the problem is to update it by taking into account the observations of the next temporal window.

### Notations

- $\mathcal{O}_j$  described by  $\{\mathcal{O}_j|\Sigma_{obs}^1, \dots, \mathcal{O}_j|\Sigma_{obs}^n\}$  represents all the observations that have been received from the beginning at time  $j$ .
- $\Delta_j$  is the diagnosis explaining  $\mathcal{O}_j$ .
- $\mathcal{W}_j$  denotes a temporal window.  $\mathcal{O}_{\mathcal{W}_j}$  described by  $\{\mathcal{O}_{\mathcal{W}_j}|\Sigma_{obs}^1, \dots, \mathcal{O}_{\mathcal{W}_j}|\Sigma_{obs}^n\}$  is the set of observations received during the temporal window  $\mathcal{W}_j$ . We have  $\forall i, \mathcal{O}_j|\Sigma_{obs}^i = [\mathcal{O}_{j-1}|\Sigma_{obs}^i, \mathcal{O}_{\mathcal{W}_j}|\Sigma_{obs}^i]$ .
- $\Delta_{\mathcal{W}_j}$  is the diagnosis on the temporal window  $\mathcal{W}_j$ .

### Problem of the incremental diagnosis

The problem of the incremental diagnosis is that, by randomly splitting the sequence of observations in temporal windows, we have no guarantee to have them satisfying the property 1 (see the previous section).

Let us see what happens on the example given by figure 5 (which is a part of the running example) with three components  $CS$  as  $\Gamma_1$ ,  $SW1$  as  $\Gamma_2$  and  $SW2$  as  $\Gamma_3$ . Each component has only two states and one transition. The initial states of the components are respectively  $x1$ ,

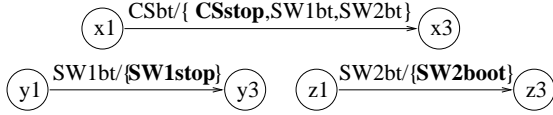


Figure 5: Simplified model of the system.

$y1$  and  $z1$ . Let us first suppose a single temporal window  $\mathcal{W}$  with  $\mathcal{O}_{\mathcal{W}} = \{[CSstop], [SW1stop], [SW2boot]\}$ . The global diagnosis, computed as described before, is given by figure 6.

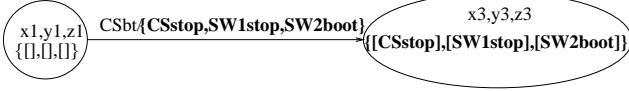


Figure 6: Global diagnosis on  $\mathcal{O}_{\mathcal{W}}$  for the simplified model.

Let us now consider two successive temporal windows with  $\mathcal{O}_{\mathcal{W}_1} = \{[CSstop], [], []\}$  and  $\mathcal{O}_{\mathcal{W}_2} = \{[], [SW1stop], [SW2boot]\}$ . The local diagnosis (see figure 7)  $\Delta_{\mathcal{W}_1, \Gamma_1}$  explains  $CSstop$  but requires synchronizations on  $SW1bt$  and  $SW2bt$  with  $\Delta_{\mathcal{W}_1, \Gamma_2}$  and  $\Delta_{\mathcal{W}_1, \Gamma_3}$ . These synchronizations are not satisfied and no diagnosis is then found. The problem is that, during  $\mathcal{W}_1$ ,  $SW1stop$  and  $SW2boot$  have been emitted by the component but are still not received by the supervision center. Both alarms will be received during  $\mathcal{W}_2$ . The property is clearly not satisfied on  $\mathcal{W}_1$ .

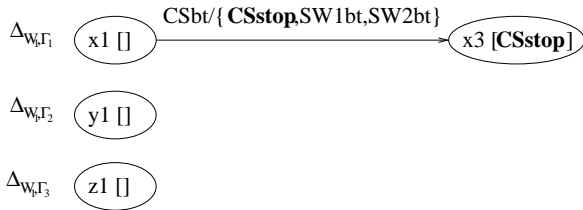


Figure 7: Local diagnoses on the window  $\mathcal{W}_1$  for the simplified model.

This example illustrates that the choice of the temporal window is very important. We firstly examine the case where, by choosing appropriate breakpoints, it is possible to ensure that each temporal window satisfies the property 1. A refinement algorithm, based on the concatenation of automata, is proposed and allows us to compute efficiently a diagnosis on successive temporal windows.

We then examine the general case and show how, by extending the definition of diagnosis, it is still possible to use the refinement algorithm.

## Incremental algorithm in the case of sound temporal windows

A first solution is to carefully choose the breakpoints in order to ensure that each temporal window satisfies property 1.

**Definition 5** A temporal window  $\mathcal{W}_j$  is said to be sound wrt a sequence of observations  $\mathcal{O}_{j-1}$  iff  $\forall o_2 \in \mathcal{O}_{\mathcal{W}_j}, \forall o_1 \in \mathcal{O}_{j-1}, o_2$  has been emitted after the reception of  $o_1$ . Two successive sound temporal windows meet at a sound breakpoint.

In the case where the temporal windows are sound, the global diagnosis  $\Delta_j$  results from the concatenation of the diagnosis  $\Delta_{j-1}$  with the diagnosis  $\Delta_{\mathcal{W}_j}$ . The only condition is that the final states of  $\Delta_{j-1}$ , noted  $X_{final}$ , are considered as the initial states for  $\Delta_{\mathcal{W}_j}$ .  $\Delta_{\mathcal{W}_j}$  is computed as before (eq 1) by  $\Delta_{\mathcal{W}_j} = \Delta_{\Gamma}(X_{final}, \mathcal{O}_{\mathcal{W}_j}) = \bigodot_{i=1}^n \Delta_{\Gamma_i}(X_{final}, \mathcal{O}_{\mathcal{W}_j} | \Sigma_{obs}^i)$ . The global diagnosis  $\Delta_j$  is computed by the application of a refinement operator (noted  $\oplus$ ) defined by the incremental algorithm 1. We have  $\Delta_j = \Delta_{j-1} \oplus \Delta_{\mathcal{W}_j}$ .

---

### Algorithm 1 Refinement operation: $\Delta_j = \Delta_{j-1} \oplus \Delta_{\mathcal{W}_j}$

---

**input:** Diagnosis on the past windows  $\Delta_{j-1}$   
**input:** Diagnosis on the current window  $\Delta_{\mathcal{W}_j}$   
 {Appending of the diagnoses}  
 $\Delta_{tmp} \leftarrow Append(\Delta_{j-1}, \Delta_{\mathcal{W}_j})$   
 {Eliminating trajectories that do not explain all the observations  $\mathcal{O}_j$ }  
**for all**  $x = (s_{\Gamma}, (\mathcal{E}_1, \dots, \mathcal{E}_n)) \in final\_states(\Delta_{tmp})$   
**do**  
   **if**  $\exists \mathcal{E}_i$  such that  $\mathcal{E}_i \neq \mathcal{O}_j | \Sigma_{obs}^i$  **then**  
      $\{x$  is not a final state in the new diagnosis. $\}$   
      $\Delta_{tmp} \leftarrow ElimTraj(\Delta_{tmp}, x)$   
**end if**  
**end for**  
**output:**  $\Delta_{\mathcal{W}_j} \leftarrow \Delta_{tmp}$

---

*Append* is an operation based on the classical concatenation of finite-state machines (Hopcroft & Ullman 1979). *ElimTraj* is the operation which eliminates the states  $x$  from which we cannot find a diagnosis for the observations of  $\mathcal{W}_j$ . *ElimTraj* also eliminates all the states that are predecessors of  $x$  and have no other successors<sup>2</sup>.

We present in the figure 8 the update of the global diagnosis of the system presented in the figure 4.

We suppose that in the new temporal window  $\mathcal{W}_j$ , we observe  $\mathcal{O}_{\mathcal{W}_j} = \{[CSrun], [], [SW2boot]\}$ . In the diagnosis of the figure 4, there is one final state  $((x2, y2, z1), \{[CSstop], [SW1stop], []\})$ . We compute the global diagnosis  $\Delta_{\{CS, SW1, SW2\}}((x2, y2, z1), \mathcal{O}_{\mathcal{W}_j})$  which is appended to the final state of the previous diagnosis. In this update, there is no elimination because

<sup>2</sup>Typically, such an elimination can be done with the help of a classical garbage collector.

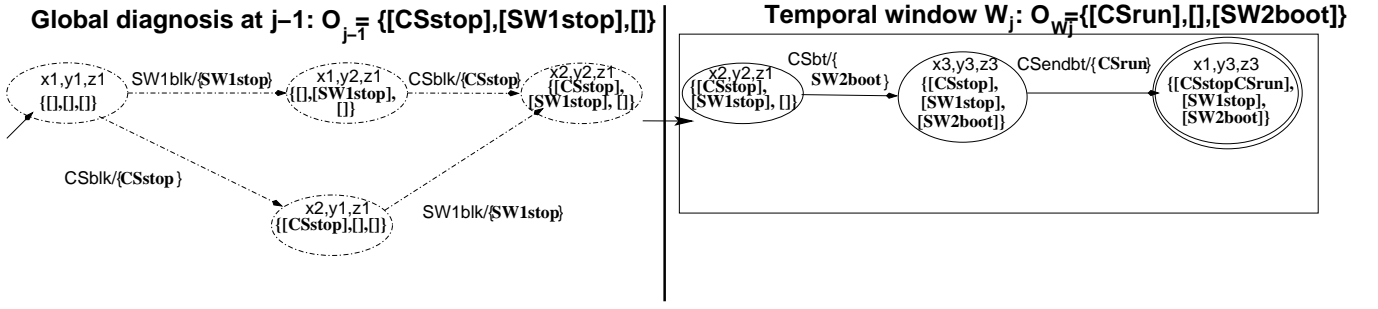


Figure 8: Update of the global diagnosis  $\Delta_j$ .

each previous final state is followed by an explanation of  $\mathcal{O}_{W_j}$ . We can remark that the resulting diagnosis do not contain trajectories which explain  $CSstop$  by the occurrence of  $CSbt$ . Because we assume the soundness of the windows, we suppose that  $CSstop$  is observed before the emission of  $SW2boot$ , so  $CSbt$  cannot explain the observation  $CSstop$ .

### Incremental algorithm in the general case

It is not always possible to select sound breakpoints. In this section, we consider the general case where it cannot be guaranteed that the temporal windows are sound. The difficulty of an incremental diagnostic algorithm consists then in taking into account two kind of observable events:

1. the observations received by the supervisor in the current temporal window;
2. the events emitted by the components during (or even before) the current temporal window which have not yet been received by the supervisor in the current temporal window. They are still in the communication channels.

In order to have an efficient way to update the current global diagnosis of the system, we want to use the refinement algorithm presented above (see algorithm 1). The idea is to complete the set of received observations by a set of potentially emitted (but not yet received) events. Therefore, we propose to compute, for each temporal window  $W_j$ , an *extended* diagnosis  $\Delta_{W_j}^{ext}$  which summarizes trajectories that explain the events observed in  $W_j$  and a set of hypothetical unreceived events during  $W_j$ .

$\Delta_{W_j}^{ext}$  is computed (see algorithm 2) by composing *extended* local diagnoses  $\Delta_{\Gamma_i}^{ext}(W_j)$  in a similar way as seen above.

**Extended local diagnosis** The *extended* local diagnosis  $\Delta_{\Gamma_i}^{ext}(W_j)$  on the window  $W_j$  depends on the states of  $\Gamma_i$  described in the final states of the extended diagnosis  $\Delta_{j-1}^{ext}$  (extended diagnosis of  $\mathcal{O}_{j-1}$ ). In such a state  $x$ ,  $x = ((x_1, \dots, x_n), (\mathcal{E}_1, \dots, \mathcal{E}_n))$ , some observations of  $\Gamma_i$  may have been supposed to be emitted and unreceived before  $W_j$ , we have  $\mathcal{E}_i = [\mathcal{O}_{j-1} | \Sigma_{obs}^i, SupposedObs_x]$ . Looking to  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$ , it

can be checked whether this supposition is satisfied or not. If it is satisfied, we have two cases:

1.  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$  is a prefix of  $SupposedObs_x$ . This means that  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$  was totally explained in the previous temporal window. Thus,  $SupposedObs_x = [\mathcal{O}_{W_j} | \Sigma_{obs}^i, unObs_x]$  where  $unObs_x$  is a sequence of observations of  $\Gamma_i$  potentially emitted before  $W_j$  and not yet observed during  $W_j$ .
2.  $SupposedObs_x$  is a prefix of  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$ . This means that  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$  was partially explained in the previous temporal window. Thus,  $\mathcal{O}_{W_j} | \Sigma_{obs}^i = [SupposedObs_x, UnExplainObs_x]$  where  $UnExplainObs_x$  is the sequence that terminates  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$  which is not yet explained.

In the first case,  $x$  is a state resulting from trajectories which already explain the observations  $\mathcal{O}_{W_j} | \Sigma_{obs}^i$ . We only need to determine local trajectories from  $x_i$  which explain hypothetical unreceived events which can follow the events of  $unObs_x$ . In the second case, we have to determine local trajectories from  $x_i$  which explain  $UnExplainObs_x$  followed by hypothetical unreceived events.

Therefore, in both cases, we have to determine hypothetical unreceived events. If we do the hypothesis that there is a bounded number  $k$  of local observations at the same time in the communication channel associated to  $\Gamma_i$ , then the sequence of hypothetical unreceived events of  $\Gamma_i$  is finite and belongs to:

**Definition 6** ( $UnRcvObs_i(k)$ ) Let  $k$  be a positive integer. We note by  $UnRcvObs_i(k)$  the sequences of observable events  $sq$  such that  $sq \in (\Sigma_{obs}^i)^*$ ,  $|sq| \leq k$ .

For example, in figure 2,  $UnRcvObs_{CS}(2) = \{[], [CSrun], [CSstop], [CSrun CSstop], [CSstop CSrun]\}$ .

Thus from the state  $x$ , we have to compute the local trajectories of  $\Gamma_i$  which explain the sequences of  $Comp_i(x, k)$  where  $Comp_i(x, k)$  is a set of observation sequences. In the case 1, each sequence  $obsSeq$  of  $Comp_i(x, k)$  is such that:

$$obsSeq \in UnRcvObs_i(k - |unObs_x|).$$

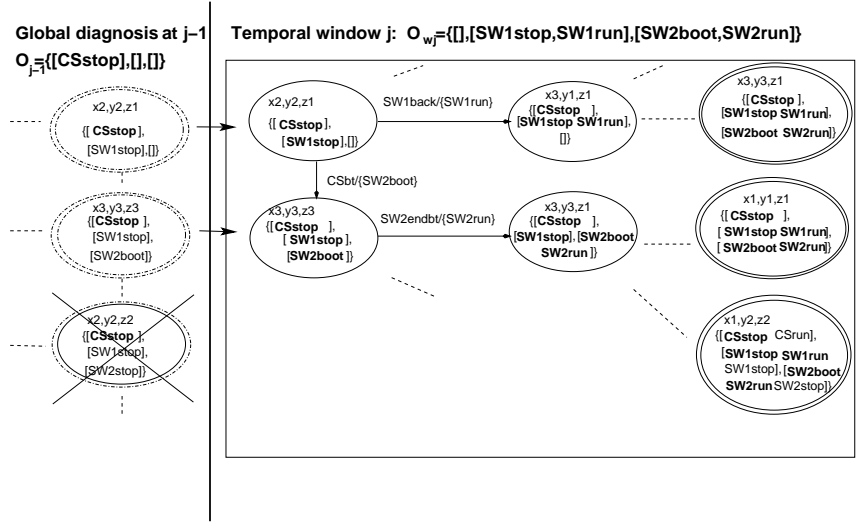


Figure 9: Update of the extended global diagnosis  $\Delta_j^{ext}$ .

In the case 2,  $obsSeq$  is such that:

$$obsSeq = [UnExplains_x, UnRcvObs]$$

$$UnRcvObs \in UnRcvObs_i(k).$$

Thus, the extended diagnosis  $\Delta_{\Gamma_i}^{ext}(\mathcal{W}_j)$  is the set of local trajectories computed from each final state  $x$  of  $\Delta_{j-1}^{ext}$  that explain  $Comp_i(x, k)$ . Because we make hypotheses about the set of unreceived events during  $\mathcal{W}_j$ , each state resulting of a trajectory that explains such events is possible, so we mark it as a final state.

---

**Algorithm 2** Extended diagnosis of  $\mathcal{W}_j$  :  $\Delta_{\mathcal{W}_j}^{ext}$

---

**input:**  $\mathcal{O}_{j-1}, \mathcal{O}_{\mathcal{W}_j}$   
**input:**  $X_{init}^j$  {Final states of the diagnosis  $\Delta_{j-1}^{ext}$ }  
**for all**  $i \in \{1, \dots, n\}$  **do**  
  {Computation of the extended diagnosis of  $\Gamma_i$ .}  
   $\Delta_{tmp} \leftarrow \emptyset$   
  **for all**  $x \in X_{init}^j$  **do**  
     $\{x = ((x_1, \dots, x_n), (\mathcal{E}_1, \dots, \mathcal{E}_n))\}$   
    {Computation of the local observation sequences to explain}  
    **for all**  $obsSeq \in Comp_i(x, k)$  **do**  
       $\Delta_{tmp} \leftarrow \Delta_{tmp} \cup \Delta_{\Gamma_i}(\{x_i\}, obsSeq)$   
    **end for**  
  **end for**  
   $\Delta_{\Gamma_i}^{ext}(\mathcal{W}_j) \leftarrow \Delta_{tmp}$   
**end for**  
{Computation of the extended diagnosis of  $\mathcal{W}_j$ .}  
**output:**  
 $\Delta_{\mathcal{W}_j}^{ext} \leftarrow \bigodot_{i=1}^n \Delta_{\Gamma_i}^{ext}(\mathcal{W}_j)$

---

**Update of the global diagnosis** The global diagnosis  $\Delta_j$  is computed, as before, by the application of the refinement operation (noted  $\oplus$ ) defined by the algorithm 1. The difference is that  $\Delta_j$  is changed in  $\Delta_j^{ext}$ . We have:

$$\Delta_j^{ext} = \Delta_{j-1}^{ext} \oplus \Delta_{\mathcal{W}_j}^{ext}.$$

After the use of the refinement algorithm on the temporal window  $\mathcal{W}_j$ , the current diagnosis  $\Delta_j^{ext}$  describes trajectories that all explain  $\mathcal{O}_j$  and some of them explain a set of complementary events supposed to have been emitted but not yet observed by the supervisor at  $\mathcal{W}_j$ . It is then clear that we have:  $\Delta_{\Gamma}(X_{init}, \mathcal{O}_j) \subseteq \Delta_j^{ext}$ .

Finally, if we consider  $\mathcal{O}_m$  as “complete”, meaning that no more observation is expected ( $\mathcal{W}_m$  is the last window), the extended diagnosis  $\Delta_{\mathcal{W}_m}^{ext}$  is computed with  $k = 0$  (no expected event). Thus, potential wrong assumptions made during the computation of the successive extended diagnoses are eliminated with help of the diagnosis of the last window. We get:  $\Delta_{\Gamma}(X_{init}, \mathcal{O}_m) = \Delta_m^{ext}$ .

**Example** We present in the figure 9, the scheme of the  $\Delta_j^{ext}$  computation. In this example, the extended diagnosis  $\Delta_{j-1}^{ext}$  explains the observation  $\mathcal{O}_{j-1} = \{\{CSstop\}, [], []\}$ . Some of its final states are represented. Some final states of  $\Delta_{j-1}^{ext}$  not only contain the occurrence of  $\mathcal{O}_{j-1}$  (in bold) but also other alarms due to the completion of observable events done during  $\mathcal{W}_{j-1}$  (here, we consider that each communication channel conveys at the most  $k = 1$  observation at the same time). Given the new temporal window  $\mathcal{W}_j$  and its observations  $\mathcal{O}_{\mathcal{W}_j} = \{[], [SW1stop, SW1run], [SW2boot, SW2run]\}$ , we compute  $\Delta_{\mathcal{W}_j}^{ext}$  by considering the final states of  $\Delta_{j-1}^{ext}$ .

$\Delta_{\mathcal{W}_j}^{ext}$  summarizes the trajectories explaining  $\mathcal{O}_{\mathcal{W}_j}$ . We also complete the diagnosis by supposing the emission of other alarms. In  $\Delta_{\mathcal{W}_j}^{ext}$ , we suppose in particular the occurrence of *CSrun*, *SW1stop* or *SW2stop*. Once  $\Delta_{\mathcal{W}_j}^{ext}$  is computed, we apply the refinement algorithm. We append the initial states of  $\Delta_{\mathcal{W}_j}^{ext}$  to the corresponding final states of  $\Delta_{j-1}^{ext}$ . Some final states of  $\Delta_{j-1}^{ext}$  are eliminated because they do not permit to find an explanation of  $\mathcal{O}_{\mathcal{W}_j}$  (in the figure, the eliminated state considers the observation of *SW2stop* whereas we observed *SW2boot*).

## Conclusion

Our motivation was to extend the decentralized diagnosis approach initially presented by (Pencolé 2000) in order to be able to analyze, *on-line*, a flow of incoming observations. In an on-line context, the observations are considered on successive temporal windows. A crucial issue is to incrementally update the current diagnosis by taking into account the observations of the next temporal window.

Two solutions have been examined. The first one consists in carefully selecting the breakpoints which determine the temporal windows. We define a property (the *soundness*) which, when satisfied by the windows, allows us to use an easy and efficient refinement algorithm based on the concatenation of automata. It is not however always possible to determine such sound breakpoints. In the general case, we propose to complete the observations by guessing what is lacking and we consequently define extended local diagnoses. The refinement algorithm can still be used for the incremental computation of the global diagnosis.

In the first case, the issue is to use domain knowledge, and especially knowledge on the properties of the communication channels, in order to split the flow of observations in sound windows. For instance, when you know the maximal delay of transmission, the absence of any alarms received by the supervisor during a delay greater than this threshold determines a sound breakpoint. In the second case, an important issue to be studied is the optimal size of the window. Small windows mean small local diagnoses but frequent computations of the current global diagnosis whereas large windows mean space-consuming local diagnoses but less computations. This point is currently investigated.

Another alternative solution to this problem of incremental decentralized diagnosis is presented in (Cordier, Pencolé, & Rozé 2001). It consists in modelling the communication channel as part of the component it is connected to. In this case, it can be shown that the soundness property is satisfied whatever the temporal windows are. The problem of the optimal size for the window becomes the central one. The counterpart is the risk of increasing the size of the local models.

## Acknowledgements

Special thanks to the anonymous referees for their valuable comments.

## References

- Baroni, P.; Lamperti, G.; Pogliano, P.; and Zanella, M. 1999. Diagnosis of large active systems. *Artificial Intelligence* 110:135–183.
- Console, L.; Picardi, C.; and Ribaudo, M. 2000. Diagnosis and diagnosability analysis using pepa. In *Proceedings of ECAI'2000*, 131–135.
- Cordier, M.-O.; Pencolé, Y.; and Rozé, L. 2001. A decentralized model-based approach for diagnosing complex systems. Submitted.
- Debouk, R.; Lafortune, S.; and Teneketzis, D. 2000. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems* 10(1-2):33–86.
- Hopcroft, J., and Ullman, J. 1979. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.
- Pencolé, Y. 2000. Decentralized diagnoser approach: application to telecommunication networks. In *Proceedings of the International Workshop on Principles of Diagnosis (DX'00)*.
- Rozé, L., and Cordier, M.-O. 1998. Diagnosing discrete event systems : An experiment in telecommunication networks. In *WODES98, Fourth Workshop on Discrete Event Systems*.
- Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamo-hideen, K.; and Teneketzis, D. 1998. Active diagnosis of discrete-event systems. *IEEE Transactions on Automatic Control* 43(7):908–929.