Decentralized diagnoser approach: application to telecommunication networks

Yannick Pencolé
IRISA
Campus de Beaulieu
35042 Rennes Cédex, FRANCE
Email: yannick.pencole@irisa.fr

Abstract

This paper presents a general method for the diagnosis of large systems, such as telecommunication networks. Because of the size of the system, the model we use is decentralized. In order to increase the efficiency of the diagnosis, the method combines two basic techniques of diagnosis: diagnosers and simulation-based techniques. We propose the construction of diagnosers based on local behaviors to compute local diagnoses. Then, we propose a coordination of local diagnoses based on a strategy which minimizes the computation for the coordination.

Introduction

The problem we deal with is the supervision of complex and large 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)[1].

Our motivation is to propose a model-based diagnosis method which can be implemented on a real system such as the largest package switching French network. We want to prove efficiency in diagnosis. There are two methods for performing supervision in model-based diagnosis: abductive techniques or simulation-based techniques.

Abductive techniques consist in the computation of failures directly from observations. One of these techniques proposed in (Sampath *et al.* 1995) is the diagnoser approach. The diagnoser approach is the compilation of diagnostic information in a data structure (called a *diagnoser*) which efficiently maps failures and observations for on-line diagnosis. The main problem of such a technique is the size of the data structure. In large systems, like telecommunication networks, it is impossible to create a *centralized diagnoser* because of the large number of states in such systems.

The simulation-based technique consists in tracking, online, the potential unobservable behavior of the system according to the observation and the model. This type of technique is proposed for example in (Baroni *et al.* 1999). One advantage of this approach is that a *decentralized model* of the system can be used. Nevertheless, the number of potential behaviors of the system according to observations may be so large that the computation may take too long to produce a useful on-line diagnosis.

We propose a combination of both techniques to set up an efficient method for diagnosing the large systems we considered. The architecture of our method is based on a decentralization of the diagnoser. We build a set of *local diagnosers* to reduce the size. In (Debouk, Lafortune, & Teneketzis 1998), an architecture of this type has been proposed; however, the diagnosers they use are still based on a global model of the system. We propose to use decentralized models and to construct *local diagnosers* based only on local behaviors described in the model. Thus, each local diagnoser efficiently produces a *local diagnosis*. Then, to obtain a *global diagnosis*, a *coordinator* computes the simulation of these diagnoses with a strategy for minimizing the computation of the overall diagnosis.

The first section of this paper introduces the telecommunication network application we are dealing with. In the second section, the behavioral model of the system is presented. Then, the general architecture of our diagnosis method is discussed, followed by the construction and coordination of the local diagnoses. Finally, we apply our method to our network application.

Application : telecommunication network

The network considered in this paper is the largest packet switching French network. This network is a hierarchical structure made up of about ten technical centers (TC) and three hundred switches (SW) (see Fig. 1). SWs route data through the network. There is one supervision center (SC) which is in charge of receiving alarms emitted by each

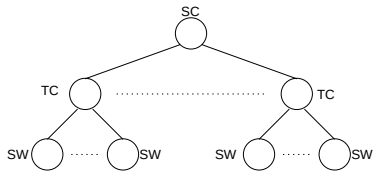


Figure 1: Hierarchical structure of the Network

[1]This work is partially supported by the RNRT project MAGDA, funded by the Ministère de la Recherche; other partners of the project are France Telecom R&D, Alcatel, Ilog, and Paris-Nord University.

equipment of the network (the SC is not in charge of supervising the data transmission itself). Analyzing observed alarms is a difficult task, due to the large number of alarms (150,000 a day). Moreover, the supervision must take the masking phenomenon into account. This phenomenon is due to the fact that alarms from hierarchically inferior equipments (e.g switches) are conveyed to the supervisor by superior equipments (such as technical centers). If one of those equipments is down, an alarm from inferior equipment is not received by the supervisor. Thus, there are some alarms the supervisor does not observe.

Our purpose is to interpret the observed alarms for the supervision operator. Earlier work for this application includes the GASPAR project[2] (Bibas *et al.* 1996) (Rozé 1997).

Model of the supervised network

Global Model

The considered model describes the behavior of the system in the case of failures on the system. A failure is modeled as a set of unobervable events that are received by the system (for instance, "begin of failure 1", "end of failure 1" are such events). When one of the failure events (Σ_{fail}) is received, we suppose that the system instantaneously reacts and potentially emits some of the observable events (Σ_{obs}). We model this behavior as a communicating finite-state machine (Brand & Zafiropulo 1983). Formally, our communicating finite-state machine Γ is:

$$\Gamma = (\Sigma_{fail}, 2^{(\Sigma_{obs})^\star}, Q, E)$$

where

- Σ_{fail} is the set of *exogenous events* (failure events) of the system;
- Σ_{obs} is the set of *observable events* of the system;
- Q is the set of global states of the system; and
- $E \subseteq (Q \times \Sigma_{fail} \times 2^{(\Sigma_{obs})^\star} \times Q)$ is the set of transitions: a transition is also composed of one input failure event (from Σ_{fail}) and a set of observations (from Σ_{obs}) in reaction to the input failure event.

With large systems, the size of such a model is potentially large. As an example, in our application, a simplified model of the network already contains $2^{10}4^{300}$ states. So a realistic diagnosis method for large systems cannot be based on such a model. We propose to base the diagnosis production on a decentralized model.

Decentralized Model of the network

The systems we consider are distributed systems. The system is composed of sub-systems which interact with each other. The sub-systems are modeled as *components*. A component models the behavior of the sub-system faced with failure occurrences on this sub-system. In such a model, two kinds of events occur to a sub-system: exogenous failure events (Σ_{fail}) or internal events (Σ_{int}). The first kind of event is modeled as an exogenous event in the global system (event from Σ_{fail}). The second kind of event happens when failures propagate in the system. We model this propagation with internal events between components (Σ_{int}). We consider that exogenous and internal events are unobservable ($(\Sigma_{int} \cup \Sigma_{fail}) \cap \Sigma_{obs} = \emptyset$). A component reacts to exogenous events or events from other components. It can emit events outside the system (observable events from Σ_{obs}) and events which affect other components (internal events). We assume that two components of the model do not emit the same events (observable or internal events). The formalism used for modeling a component Γ_i is a communicating finite-state machine:

$$\Gamma_i = (\Sigma^i_{in}, 2^{(\Sigma^i_{out})^\star}, Q_i, E_i)$$

where

- Σ^i_{in} is the set of input events ($\Sigma^i_{in} = \Sigma^i_{fail} \cup \Sigma^i_{int}$);
- Σ^i_{out} is the set of output events ($\Sigma^i_{out} = \Sigma^i_{obs} \cup \Sigma^i_{aff}$);
- Q_i is the set of states of the component; and
- $E_i \subseteq (Q_i \times \Sigma^i_{in} \times 2^{(\Sigma^i_{out})^\star} \times Q_i)$ is the set of transitions.

Σ^i_{fail} and Σ^i_{int} are respectively the sets of exogenous and internal events that affect Γ_i ($\Sigma^i_{fail} \subseteq \Sigma_{fail}$, $\Sigma^i_{int} \subseteq \Sigma_{int}$). Σ^i_{obs} is the set of observable events potentially emitted by the component. The set $\{\Sigma^i_{obs}\}_{(1,\ldots,n)}$ is a partition of the observable events set Σ_{obs}. Σ^i_{aff} is the set of internal events which can affect the behavior of other components ($\Sigma^i_{aff} \subseteq \Sigma_{int}$). Moreover, we assume that a component cannot emit an event on itself:

$$\Sigma^i_{aff} \cap \Sigma^i_{int} = \emptyset$$

Such a decentralized model is presented in Figure 2. The presented model has three components (Γ_1, Γ_2 and Γ_3). For example, if Γ_1 is at state 1 and receives the failure event F_1, then Γ_1 emits the observable event o_{11} and the failure F_1 is propagated by the emission of the internal events i_{12} and i_{13} respectively to Γ_2 and Γ_3. The component Γ_1 goes to state 2. Then, if Γ_2 is at state 2, Γ_2 receives the i_{12} event, so it stays at state 2 without emitting anything. If Γ_3 is at state 1, it receives the i_{13} event and goes to state 2 without emitting anything.

Our goal is to compute a diagnosis of the system based on the decentralized model and avoid the building of the global model. The idea is to build a diagnosis of the system by extracting from the decentralized model the necessary information with respect to a given set of observations. This extraction consists of the computation of diagnoses of the different components (*local diagnosis*) and the building of a *global diagnosis* from the component diagnoses. In the next section, we define the notions of *global diagnosis* and *local diagnosis* and we present the architecture of the diagnosis system.

Diagnosis system

Hypotheses on the observations

We assume there is one supervisor that receives observations from each component. Each observation is labeled with the

[2] Joint project with France Telecom R&D (CNET/CNRS 93 1B 142 513 project).

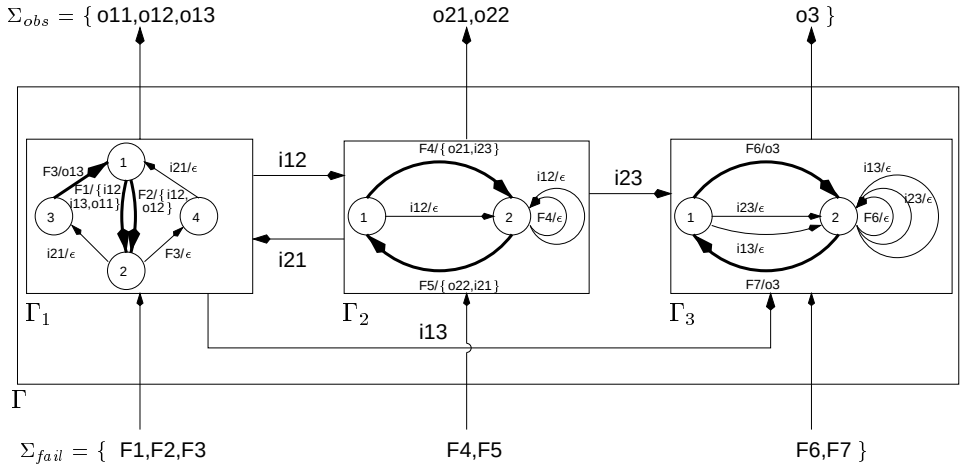


Figure 2: System model of three components. The system could be affected by 7 types of failure events : $F_1, \ldots, F_7$. It can emit 6 types of observable events (observable events are emitted by the transitions in bold). Internal events are produced when failure events propagate in the system.

date of reception by the supervisor. Thus, we do not know the order of emission of the observations because the emission order can be different from the reception order. Nevertheless, we suppose that two observed events from the same component are received by the supervisor in the order of their emission.

Therefore, we can consider that the supervisor receives a set of sequences $\sigma_1, \ldots, \sigma_n$ where σ_i is the sequence of observed events from the component Γ_i ($\sigma_i \in (\Sigma^i_{obs})^\star$). Because, we do not know the order between two observed events from two different components, the diagnosis must take all the possible orders into account. For example, in the system of Figure 2, if the supervisor observes $\{o_{11}o_{13}, o_{22}, o_3\}$, the sequence of emission of the observed events can be among others, $o_{11}o_{13}o_3o_{22}$ or $o_3o_{11}o_{13}o_{22}$ or $o_{22}o_{11}o_3o_{13}$. In the following, we denote these sets of observed event sequences as the partially ordered set $\mathcal{O} = \{\sigma_1, \ldots, \sigma_n\}$[3].

Global Diagnosis

In the supervision of systems like telecommunication networks, a diagnosis consists in providing two kinds of information. The first kind of information is the sequences of exogenous failure events that can produce the set of observations received by the supervision center. Such a sequence is a path of transitions in our global model Γ where the output events are compatible with the observations. Those sequences of failures may be infinite because some failure events can occur several times without causing observations. The second kind of information is the possible current states of the system after the emission of observations. So, a global diagnosis represents a potentially infinite set of failure sequences explaining the observations and the set of possible current states of the system after the observations. Thus, we propose to define a *global diagnosis* as a communicating finite-state machine. Each state contains a global state of the system and the subsequences of observations explained by the state. Some states are marked as final states. The final states represent the current states of the system after the observations are emitted. The transitions are labeled with exogenous failure events of the system as input and with observed events as output. Formally, according to the defined global model Γ, we define the diagnosis of the set of observations $\mathcal{O} = \{\sigma_1, \ldots, \sigma_n\}$ from a global state x of the system as:

$$\Delta(x, \mathcal{O}) = (\Sigma_{fail}, 2^{(\Sigma_{obs})^\star}, Q_\mathcal{O}, (x, \{\epsilon, \ldots, \epsilon\}), F_\mathcal{O}, E_\mathcal{O})$$

where

- $Q_\mathcal{O} \subseteq Q \times Pr(\mathcal{O})$ is the set of states. $Pr(\mathcal{O})$ is the prefix language of all the sequences represented by $\mathcal{O}$. Thus, a state of the diagnosis associates a state of the system and the partially ordered set of observed events explained at this state of the diagnosis.
- $(x, \{\epsilon, \ldots, \epsilon\})$ is the initial state of the diagnosis. The system is supposed to be at state x and the set of the explained observations at this state of diagnosis is $\{\epsilon, \ldots, \epsilon\}$.
- $F_\mathcal{O}$ is the set of final states of the diagnosis. This set contains all the states in which the system can be after the observation of $\mathcal{O}$ and under the assumption that it was at state x previously.
- $E_\mathcal{O} \subseteq (Q_\mathcal{O} \times \Sigma_{fail} \times 2^{(\Sigma_{obs})^\star} \times Q_\mathcal{O})$ is the set of failure transitions which eventually produce the observed events with respect to $\mathcal{O}$.

In Figure 3, we present the global diagnosis of the model shown in Figure 2, when we assume the initial state is (1,1,1) and the observations are $\mathcal{O} = \{o_{12}, \epsilon, o_3\}$. A diagnosed behavior is represented as a path of transitions between the

[3]If one of σ_i is equal to ϵ, it means that the supervisor does not receive any observations from the component Γ_i.

initial state and a final state. There are some loops of transitions in the machine which mean that the number of possible behaviors is infinite.

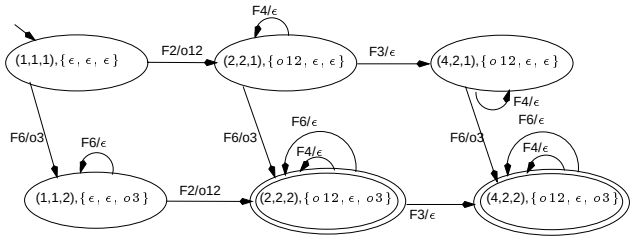


Figure 3: Global Diagnosis of the system example. The initial state of the system is (1,1,1) and the observable sequences are $\mathcal{O} = \{o_{12}, \epsilon, o_3\}$: $\Delta((1,1,1), \mathcal{O})$.

Local diagnosis

The local diagnosis is established according to the locally observed events. Given a global observation $\mathcal{O} = \{\sigma_1, \ldots, \sigma_n\}$, the local diagnosis of the component Γ_i is computed with respect to the sequence σ_i.

A local diagnosis contains the sequences of exogenous failures and propagated events (emitted and received) which explain the local sequence of observations. Formally, we define the local diagnosis of a component Γ_i from a local state x_i of the observation σ_i as the communicating finite-state machine $\Delta_i(x_i, \sigma_i)$:

$$\Delta_i(x_i, \sigma_i) = (\Sigma^i_{in}, 2^{(\Sigma^i_{out})^\star}, Q_{\sigma_i}, (x_i, \epsilon), F_{\sigma_i}, E_{\sigma_i})$$

where

- $Q_{\sigma_i} \subseteq Q_i \times Pr(\sigma_i)$ is the set of states. $Pr(\sigma_i)$ is the prefix language of σ_i. A state of the local diagnosis associates a state of the component Γ_i and a subsequence of σ_i explained at this state of the diagnosis.
- (x_i, ϵ) is the initial state of the diagnosis. The component is supposed to be at state x_i and no observed event has already been explained.
- F_{σ_i} is the set of final states of the local diagnosis. This set contains all the states in which the component can be after the observation of σ_i and under the assumption it was on state x_i previously.
- $E_{\sigma_i} \subseteq (Q_{\sigma_i} \times \Sigma^i_{in} \times 2^{(\Sigma^i_{out})^\star} \times Q_{\sigma_i})$ is the set of failures and internal event transitions. Such a transition can emit observable or/and internal events.

In Figure 4, we present the local diagnosis of Γ_1 corresponding to the example presented in Figure 3: locally, we assume the initial state is 1 and the local observations are o_{12}. According to the observations, we are sure that the component has emitted an internal event i_{12} towards Γ_2.

A local diagnosis is computed with local observations so it does not depend on the other observations. This means the local diagnosis is less constrained. The local diagnosis proposes candidate behaviors which are not compatible with

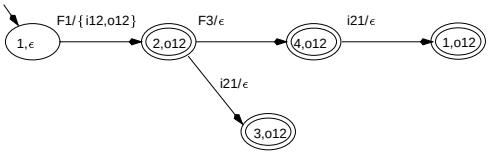


Figure 4: Local Diagnosis of component Γ_1 (at initial state 1) after observation of $\mathcal{O} = \{o_{12}, \epsilon, o_3\} : \Delta_1(1, o_{12})$

global observations. In Figure 4, the local diagnosis tells us that the local behavior can be in 4 different final states (1,2,3,4) whereas in the global diagnosis in Figure 3, we can see that only two local final states (2,4) of Γ_1 are compatible with global observations.

System Architecture

Given a set of observations, the first stage of the method consists in computing the local diagnosis of each component according to the observations. Then, those local diagnoses are merged by a coordinator to obtain a global diagnosis.

The algorithm is interactive: a human operator must ask for a diagnosis to start. We suppose that the coordinator has a set of initial states of the system. Once the process starts, the coordinator gets the set of observations $\mathcal{O} = \{\sigma_1, \ldots, \sigma_n\}$ received by the supervisor. Then, for each initial state $x = \{x_1, \ldots, x_n\}$, the coordinator asks for a local diagnosis on the component Γ_i by sending the local state x_i and the local observations σ_i to a local diagnosis machine Δ_i. The Δ_i machine is in charge of computing a local diagnosis $\Delta_i(x_i, \sigma_i)$ that is returned to the coordinator. Once the coordinator receives all the local diagnoses, coordination rules are computed. Then, the coordinator applies the rules and builds the global diagnosis $\Delta(x, \mathcal{O})$ (see Figure 5).

The following section presents the Δ_i machine. This machine is based on a data structure called *local diagnoser*. Then, we present the coordination of the local diagnoses.

Local Diagnoser

To build a local diagnosis, the algorithm parses the component model and searches for observable transitions that can be reached from a given state by a succession of unobservable transitions. This search is equivalent to a depth first search algorithm (DFS) for each observation. We propose to avoid this DFS on-line by pre-computing diagnosis information in a machine called *local diagnoser*. A local diagnoser Δ_i is a finite-state machine built off-line from a local component Γ_i. This machine is used to efficiently perform a local diagnosis on Γ_i given a state of Γ_i and a sequence of observations.

Definition

We call *observable transition* a transition of a component which emits at least an observable event[4]. We call *observable state* a state of a component which is the target of an observable transition (in Γ_1, states 1 and 2 are such observable

[4]Such transitions are in bold in Figure 2.

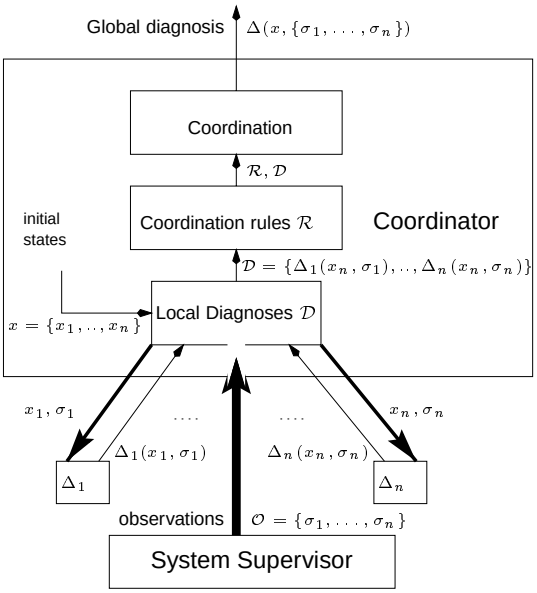


Figure 5: Architecture of diagnosis system : computation of $\Delta(x, \{\sigma_1, \ldots, \sigma_n\})$.

states). Suppose x is an observable state of the component Γ_i, we denote by $\Gamma_i^{uo}(x)$ the set of unobservable transitions of Γ_i that can be reached from state x by an unobservable path. A state of the local diagnoser Δ_i is a pair of an observable state x and $\Gamma_i^{uo}(x)$. The local diagnoser Δ_i is defined as a finite-state machine:

$$\Delta_i = (\Sigma_{in}^i, 2^{(\Sigma_{out}^i)^*}, Q_{\Delta_i}, E_{\Delta_i})$$

where

- Q_{Δ_i} is the set of pairs $(x, \Gamma_i^{uo}(x))$, where x is an observable state of Γ_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.

The diagnoser transitions of Δ_i are only observable transitions of the component Γ_i. A transition from state $x_{\Delta_i}(x_1) = (x_1, \Gamma_i^{uo}(x_1))$ to state $x_{\Delta_i}(x_2) = (x_2, \Gamma_i^{uo}(x_2))$ is defined in the diagnoser if and only if there exists an unobservable path from x_1 (path belonging to $\Gamma_i^{uo}(x_1)$) that makes it possible to reach the corresponding observable transition in the component Γ_i; the target of transition being x_2. Formally, a transition t_{Δ_i} of the diagnoser Δ_i is defined by:

$$t_{\Delta_i} = ((x_1, \Gamma_i^{uo}(x_1)), in, out, (x_2, \Gamma_i^{uo}(x_2))) \in E_{\Delta_i}$$
$$\equiv \exists t_{\Gamma_i} \in \Gamma_i | t_{\Gamma_i} = (x_1', in, out, x_2) \wedge x_1' \in \Gamma_i^{uo}(x_1).$$

In Figure 6, we present the diagnoser of the component Γ_1 (see Figure 2). The observable states of Γ_1 are 1 and 2; they are represented as the initial states of the associated set of unobservable paths ($\Gamma_1(1)$ and $\Gamma_1(2)$). This diagnoser is thus composed of two states ($x_{\Delta_1}(1)$ and $x_{\Delta_1}(2)$).

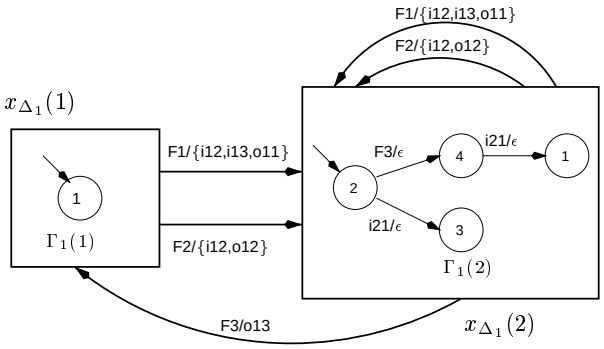


Figure 6: Local diagnoser of the component Γ_1.

Thus, a diagnoser is able to follow the observable behavior of the component. Moreover, each state of the diagnoser contains a set of unobservable transitions that the component can reach between each observable transition.

Building of local diagnoses

Each state of Δ_i contains a set of unobservable paths $\Gamma_i^{uo}(x)$ where x is an observable state of Γ_i. If we observe an event o occurring whereas the component was at state x, this means that only a subset of the unobservable paths set $\Gamma_i^{uo}(x)$ can have occurred before the emission of the event o. We denote by $\Gamma_i^{uo}(x, o)$ this subset of unobservable paths. In other words, $\Gamma_i^{uo}(x, o)$ is the set of unobservable transitions from Γ_i which can have been passed through before the emission of o whereas Γ_i was at state x.

If we concatenate to $\Gamma_i^{uo}(x, o)$ the output diagnoser transitions labeled with the observable o, we obtain the local paths of transitions that explain the observation o whereas the component was at state x. In Figure 7, the observable state of Γ_1 is 2 and the observation is o_{13}. The unobservable paths of Γ_1 from 2 are described in the diagnoser ($\Gamma_1(2)$) and are composed of three transitions. If the next observed event is o_{13}, then the possible unobservable path is $\Gamma_1(2, o_{13})$ – a part of $\Gamma_1(2)$ composed only of the transition from state 2 to 3. The explanation of o_{13} from state 2 is obtained by appending the diagnoser transition labeled with o_{13} to the unobservable path $\Gamma_1(2, o_{13})$ (see the right-hand part of Figure 7).

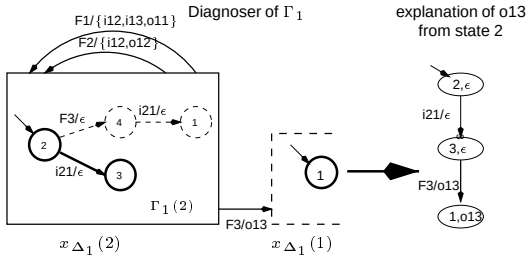


Figure 7: Construction of the explanation of the observation o_{13} from state 2 in the component Γ_1.

Thus, constructing a local diagnosis consists in parsing

the diagnoser from a given state and extracting the unobservable paths[5] according to the observations and appending them. In the case when a state x of the diagnoser is accessed at the end of the parse, this means that the component can be in any states of $\Gamma_i^{uo}(x)$; $\Gamma_i^{uo}(x)$ is then appended to the rest of the local diagnosis and states of $\Gamma_i^{uo}(x)$ are marked as final states. In Figure 8, we present the parse of the local diagnoser Δ_1 for the construction of the local diagnosis $\Delta_1(1, o_{11}o_{13})$.

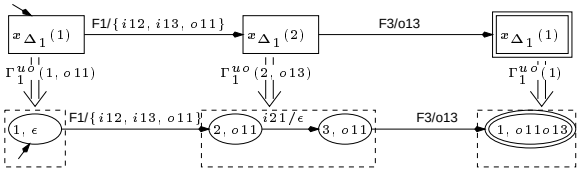


Figure 8: Parse of the diagnoser Δ_1 from state $x_{\Delta_1}(1)$ to construct the local diagnosis $\Delta_1(1, o_{11}o_{13})$.

To sum up, the on-line construction of a local diagnosis consists of a parse of the local diagnoser according to the local observation sequence. The diagnoser avoids an on-line track of unobservable and observable transitions with respect to the observations, thanks to a compilation of unobservable paths and a direct access to observable transitions.

Global diagnosis

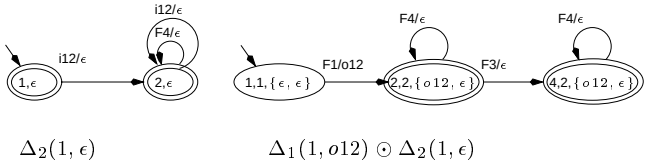
Given a partial order set $\mathcal{O} = \{\sigma_1, \ldots, \sigma_n\}$ of observable events and a global state of the system $x = (x_1, \ldots, x_n)$, each local diagnoser computes the local diagnosis $\Delta_i(x_i, \sigma_i)$. The next operation is the computation of a global diagnosis relying on these local diagnoses.

Merging of diagnoses

We have defined local diagnoses as finite-state machines which represent the possible behaviors of components according to the local observations. When a diagnosis of a component is built, we do not take the interactions with other components into account. Thus, a local diagnosis can describe a set of local behaviors that are not compatible with diagnosed behaviors of another component. The merging of two diagnoses consists in eliminating such incompatible behaviors and computing compatible shared behaviors. Thus, the merging of two diagnoses is based on a composition operation between two communicating finite-state machines. This composition is the classical (parallel) composition operation with a synchronization on the internal events (Bibas *et al.* 1996). A state of the composed diagnosis is final if it is the composition of final states of the local diagnoses. Hereafter, we denote this operation by $\odot$.

[5]The extraction of $\Gamma_i^{uo}(x, o)$ can be implemented by marking the concerned unobservable transitions during the diagnoser construction. Thus we have an efficient extraction of $\Gamma_i^{uo}(x, o)$ by just looking for unobservable transitions that have the corresponding mark.

In Figure 9 on the right, we present the composition of the local diagnoses $\Delta_1(1, o_{12})$ (see Figure 4) and $\Delta_2(1, \epsilon)$ (see Figure 9 on the left). The result $\Delta_1(1, o_{12}) \odot \Delta_2(1, \epsilon)$



$\Delta_2(1, \epsilon)$ $\Delta_1(1, o12) \odot \Delta_2(1, \epsilon)$

Figure 9: Composition of the local diagnoses $\Delta_1(1, o12)$ and $\Delta_2(1, \epsilon)$. $\Delta_1(1, o12) \odot \Delta_2(1, \epsilon)$ is the diagnosis of components Γ_1 and Γ_2 when they are at state 1 and the observation is o_{12}.

contains three states. This composed diagnosis represents the set of failure events sequences that can occur on the system (Γ_1,Γ_2) when each component is at state 1 and an event o_{12} from Γ_1 is observed. In $\Delta_1(1, o_{12})$, at the state $(1, \epsilon)$, an event i_{12} is supposed to be emitted by Γ_1 after the reception of the event F_1. In $\Delta_2(1, \epsilon)$, at the state $(1, \epsilon)$, the same event i_{12} is supposed to be received. Thus, these two local behaviors are compatible because the emission and the reception of the internal event i_{12} can be synchronized. In $\Delta_1(1, o_{12}) \odot \Delta_2(1, \epsilon)$, this behavior is described by the transition labeled with F_1/o_{12} from state $(1, 1, \{\epsilon, \epsilon\})$ to state $(2, 2, \{o_{12}, \epsilon\})$.

With this composition, we obtain a global diagnosis by applying the composition on all the local diagnoses:

$$\Delta(x, \mathcal{O}) = \bigodot_{i=1}^{n} \Delta(x_i, \sigma_i)$$

For example, the global diagnosis of Figure 3 is obtained by the operation: $\Delta((1, 1, 1), \{o_{12}, \epsilon, o_3\}) = \Delta_1(1, o_{12}) \odot \Delta_2(1, \epsilon) \odot \Delta_3(1, o_3)$.

Strategy for the coordination

The $\odot$ composition operation is commutative and associative, so we can compose local diagnoses in many different ways and also use parallel computations. Though the result is the same, some composition operations are more efficient than others. Thus, one task of the coordinator is to plan composition stages on local diagnoses to parallelize the operation of coordination and minimize the computation.

To optimize the computation, the idea is to eliminate incompatible local diagnoses in first stages of compositions. Diagnoses from two components are incompatible if their respective interactions cannot be synchronized. Therefore, if we compute the composition of local diagnoses that are directly in interaction before any other compositions, we rapidly eliminate incompatible hypotheses.

The problem is now to know which components are interacting. We can use the static information given by the decentralized model – for example, in Figure 2, we know Γ_1 has potential interactions with Γ_2 and Γ_3 (internal events i_{12} and i_{21} and i_{13}). Nevertheless, we can be more precise

and efficient if we are able to deduce the potential interactions from the local diagnoses. For example, in the local diagnosis of Figure 4, we know that Γ_1 interacts with Γ_2 but not with Γ_3. Consequently, we know that Γ_1 and Γ_3 have independent local diagnoses.

Computation of component interactions The information about the interactions of one component can be extracted from the local diagnosis by looking for interactive transitions and noticing the components which interact. This extraction can be pre-computed (off-line) in the local diagnoser. In Figure 10, we present pre-computed information about interactions of the component Γ_1. In a flag of a diagnoser transition, we list the components that are in interaction if we pass over the diagnoser transition. These interactions are computed by looking at interaction described in the diagnoser transition and in the unobservable transitions (in the diagnoser state source) that can occur before. For example, for the transition labeled F_3/o_{13} from state $x_{\Delta_1}(2)$ to $x_{\Delta_1}(1)$, there is only one possible interaction. This interaction is the reception of an i_{21} event from Γ_2 (transition from state 2 to 3). So the flag contains the fact that there is a possible interaction with Γ_2 if we pass over this diagnoser transition.

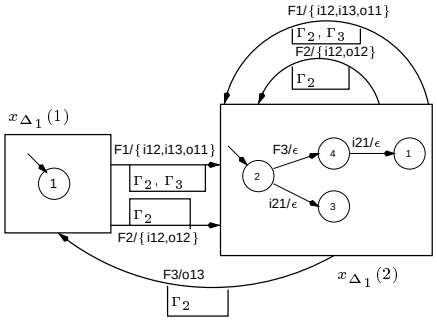


Figure 10: Local diagnoser Δ_1 of the component Γ_1 with flags denoting potential interactions on diagnoser transitions.

Computation of coordination rules The task of the coordinator is to take into account on-line the interaction hypotheses and provide rules for the coordination of the local diagnoses.

The interactions the coordinator receives are all the potential interactions of all the local diagnoses from a local point of view. A diagnoser Δ_i may claim there is an interaction between Γ_i and Γ_j whereas the diagnoser Δ_j may claim there is no such interaction. Thus, the coordinator retains information on interactions with respect to the following rule:

> If a local diagnosis $\Delta_i(x_i, \sigma_i)$ claims it interacts with another $\Delta_j(x_j, \sigma_j)$, the coordinator keeps the interaction $\{\Delta_i(x_i, \sigma_i), \Delta_j(x_j, \sigma_j)\}$ only if $\Delta_j(x_j, \sigma_j)$ claims it interacts with $\Delta_i(x_i, \sigma_i)$.

Thus, the coordinator keeps an hypothesis of interaction between two components only if their respective diagnosers agree about this hypothesis of interaction. From those interactions, the coordinator deduces rules for the coordination of the local diagnoses.

Coordination of the local diagnoses The coordination is decomposed into different stages of composition between the local diagnoses. In one stage, the coordination consists in choosing a partition of the set of diagnoses depending on the computed interactions and then applying in parallel the composition of each subset. The result of this operation is another smaller set of composed diagnoses (more global). The coordination is finished when the last composition has produced a unique diagnosis which is the global diagnosis (see algorithm 1).

Algorithm 1 Coordination of local diagnoses.

input: $\mathcal{D} = \{\Delta_1(x_1, \sigma_1), \ldots, \Delta_n(x_n, \sigma_n)\}$
input: interactions between the local diagnoses: $\mathcal{I}$
while $|\mathcal{D}| > 1$ **do**
  $\pi_{\mathcal{D}} \leftarrow OnePartitionWithInteraction(\mathcal{D}, \mathcal{I})$
  {Partition of $\mathcal{D}$ according to interactions of I.}
  $\mathcal{D} \leftarrow ComposeInParallel(\pi_{\mathcal{D}})$
  {Application of the composition on the set of the partition in parallel.}
end while
$\Delta((x_1, \ldots, x_n), \{\sigma_1, \ldots, \sigma_n\})$ is the element of $\mathcal{D}$

$OnePartitionWithInteraction(\mathcal{D}, \mathcal{I})$ chooses a partition $\pi_{\mathcal{D}}$ of $\mathcal{D}$ such that each set of the partition contains diagnoses which interact according to the interactions set $\mathcal{I}$[6]. $ComposeInParallel(\pi_{\mathcal{D}})$ produces a set of diagnoses. Each diagnosis corresponds to the composition of a subset of diagnoses in $\pi_{\mathcal{D}}$. These compositions are done in parallel to increase the efficiency of the computation.

Example

Suppose we compute the following diagnosis $\Delta((1,1,1), \{o_{12}, \epsilon, o_3\})$ of the system in Figure 2. Local diagnosers return:

- $\Delta_1(1, o_{12})$: interactions with Γ_2 (as shown in Figure 4);
- $\Delta_2(1, \epsilon)$: interactions with Γ_1 (as shown in Figure 9);
- $\Delta_3(1, o_3)$: interactions with Γ_1 and Γ_2.

Only $\Delta_1(1, o_{12})$ and $\Delta_2(1, \epsilon)$ agree with their interactions. So, the coordinator keeps only the interaction set $\mathcal{I} = \{(\Delta_1(1, o_{12}), \Delta_2(1, \epsilon)\}$. Thus, the first stage of the coordination consists in applying composition on the partition $\{\{\Delta_1(1, o_{12}), \Delta_2(1, \epsilon)\}, \{\Delta_3(1, o_3)\}\}$. The result is the set $\{\Delta_1(1, o_{12}) \odot \Delta_2(1, \epsilon), \Delta_3(1, o_3)\}$. The second stage consists in composing the two last diagnoses to obtain $\Delta((1,1,1), \{o_{12}, \epsilon, o_3\})$ from Figure 3.

[6] In the case when $\mathcal{D}$ contains diagnoses that do not interact, we can choose a partition according to other parameters for the purpose of efficiency: among others, the number of diagnoses in a set of the partition or the size of the diagnoses.

Thus, the coordination of the local diagnoses is a parallelized operation. Moreover each operation is decided thanks to possible interactions between the local diagnoses in order to optimize the computation.

Application of the method

We have already implemented a decentralized model of the network application in a software called Dyp. Each component of the model describes the behavior of one network equipment (e.g switch, technical center) according to the masking phenomenon. Therefore, the components only describe local behaviors.

The masking phenomenon is modeled with internal events between components. For example, for a *TC_break* event (breakdown of the technical center), the *TC* component emits a masking event to its *SW* components. Once a *SW* receives such an event, it does not emit observable events until it receives a demasking event from the *TC* component.

A typical diagnoser is shown for a technical center in Figure 11. In this example, we suppose the technical center has 2 switches it can mask (*SW1* and *SW2*).

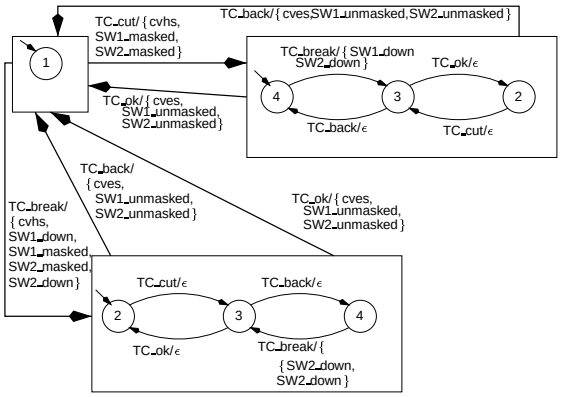


Figure 11: Local diagnoser of a technical center. The observable events of a technical center are *cvhs* and *cves*.

Related works

As noted previously, our work has been influenced by other methods for diagnosing large systems. As far as the diagnoser techniques are concerned, (Debouk, Lafortune, & Teneketzis 1998) have proposed a system of coordination of diagnosers. The diagnosers have a local point of view of the system but directly compute diagnoses of global states of the system. Thus, the coordination only consists in detecting which diagnosers have the better global diagnoses at a given time. In our application, because of the large number of global states, the size of such diagnosers is prohibitive; it is why we chose to rely on diagnosers based on local models.

As far as the simulation-based techniques are concerned, our work is influenced by (Baroni *et al.* 1999) and (Lamperti & Zanella 1999). These authors proposed simulation methods for diagnosing systems they called *active*. They presented an algorithm which is a completely on-line local diagnosis construction. We chose to partially compile this treatment by using diagnosers. Concerning the global diagnosis construction, they use a predefined *reconstruction plan*. With our strategy of coordination, we have presented an automatic reconstruction plan which relies on the local diagnoses.

Conclusion

We have presented a method for diagnosing large systems like telecommunication networks. Three points have been presented for the purpose of efficiency. The first one is the use of a decentralized model of the system (model of communicating components). The second one is the use of local diagnosers on components of the system. Those diagnosers are realistic and efficient structures in order to carry out a local diagnosis of a component. The third point is about a strategy for the coordination of the local diagnoses. The coordination is based on composition operations in parallel and a strategy which minimizes the on-line computation.

This approach is being implemented. We have already implemented a software (called Dyp) which can initialize a decentralized model of the supervised system from a file description. We are currently implementing an incremental version of the algorithm presented in this paper and we will experiment it shortly on the telecommunication network application.

References

Baroni, P.; Lamperti, G.; Pogliano, P.; and Zanella, M. 1999. Diagnosis of large active systems. *Artificial Intelligence* 110:135–183.

Bibas, S.; Cordier, M.-O.; Dague, P.; Lévy, F.; and Rozé, L. 1996. Gaspar: a model-based system for diagnosing telecommunication networks. In *IMACS-IEEE/SMC International Multiconference ofComputational Engineering in Systems Applications (CESA'96)*.

Brand, D., and Zafiropulo, P. 1983. On communicating finite-state machines. *Journal of ACM* 30(2):323–342.

Debouk, R.; Lafortune, S.; and Teneketzis, D. 1998. A coordinated decentralized protocol for failure diagnosis of discrete event systems. In *Fourth Workshop on Discrete Event Systems (WODES'98)*, 138–143.

Lamperti, G., and Zanella, M. 1999. Diagnosis of discrete-event systems integrating synchronous and asynchronous behavior. In *Proceedings of the International Workshop on Principles of Diagnosis (DX'99)*, 129–139.

Rozé, L. 1997. Supervision of telecommunication network : a diagnoser approach. In *Proceedings of the International Workshop on Principles of Diagnosis (DX'97)*, 103–111.

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