All from one, one for all, failure diagnosis of discrete event systems using representatives

Yannick Pencolé
CSL, The Australian National University, Canberra, ACT 0200, Australia
Yannick.Pencole@anu.edu.au

Abstract

Failure diagnosis in large and complex systems is a critical and challenging task. In the realm of model based diagnosis on discrete event systems, computing a failure diagnosis means computing the set of system behaviours that could explain observations. Depending on the diagnosed system, such behaviours can be numerous, so that a problem of representing them is induced. The paper discusses about this problem and presents a way of representing a diagnosis by the use of a partial order reduction technique.

1. Introduction

The problem of failure diagnosis on discrete event systems has received considerable attention in the literature of various fields including Artificial Intelligence [2, 5, 11, 10] as well as Control [12, 6, 7]. Given a model of the system representing the behaviour of the system (faulty or not) and a set of observations, the problem of failure diagnosis consists in determining the behaviours that could explain the set of observations. A behaviour is defined as a sequence of events that occur on the system and can be represented by a sequence of triggered transitions of the model.

Diagnosing complex discrete event systems implies finding a set of behaviours in a very complex state space. Therefore, the diagnosis problem is strongly linked with the well-known *state explosion problem* [9], problem which essentially comes from the fact that the system evolves in a concurrent way. Then, computing the diagnosis can be a very complex task, and the solution can be very big and cannot be easily analysed.

This paper deals with this problem by proposing to represent the diagnosis in a way that takes into account the concurrency in the diagnosed system. The proposed diagnosis representation is based on the notion of *traces* [8] which implicitly represents a set of solutions. In order to compute the diagnosis in such a representation, a partial order reduction technique is used [9].

The paper is organised as follows. The section 2 introduces the considered systems: a formal framework and an example are presented. The representation of the diagnosis is described in the section 3. Section 4 presents some results and section 5 presents a final discussion.

2. Model of system

2.1. Syntax of the model

The considered systems are reactive systems, they evolve by the occurrence of events on the system (see figure 1 for an example extracted from a real communication network). They are also based on a set of components. Each component has its own behaviour and can interact with other components (in the example, there are three kinds of components: switches, connections between switches, and stations which control the switches). To model a component, the communicating automata formalism has been chosen. Communicating automata are well suited for modelling components which communicate each other, and have been used in several previous works [2, 11, 10]. A component can emit *observable events* Σ^i_{obs}, *internal events* $\Sigma^i_{intemtd}$ (events which model propagations towards the other components). A component can receive *internal events* $\Sigma^i_{intrcvd}$ and *exogenous events* Σ^i_{exo} (events from the environment of the system, especially *failure events*).

Definition 1 *The* model of a component *is described by a communicating automaton* Γ_i *:*

$$\Gamma_i = (\Sigma^i_{trg}, \Sigma^i_{emit}, Q_i, E_i, q_{0i})$$

- Σ^i_{trg}*: triggering events (*$\Sigma^i_{trg} = \Sigma^i_{exo} \cup \Sigma^i_{intrcvd}$*);*
- Σ^i_{emit}*: emitted events (*$\Sigma^i_{emit} = \Sigma^i_{obs} \cup \Sigma^i_{intemtd}$*);*
- $\Sigma^i_{trg} \cap \Sigma^i_{emit} = \emptyset$ *;*
- Q_i*: component states;* q_{0i}*: initial state;*
- $E_i \subseteq (Q_i \times \Sigma^i_{trg} \times 2^{(\Sigma^i_{emit})} \times Q_i)$*: transitions.*

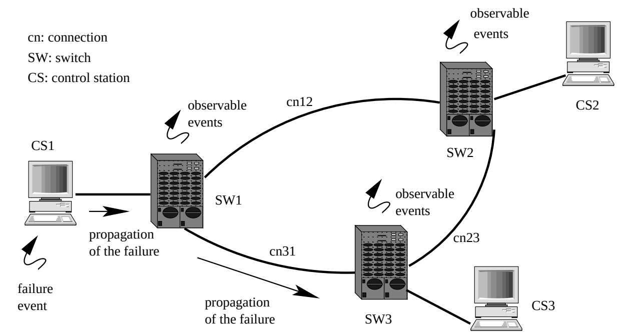

Figure 1. Communication network example.

Notation 1 *For any component transition $q \xrightarrow{t} q'$, $trg(t)$ is the trigger of the transition, $emit(t)$ the set of internal events emitted by t, and $obs(t)$ the set of observable events emitted by t.*

In figure 2, the model of the component *CS1* is presented. This control station can hang up and becomes operational again: this behaviour is modelled by the events *CS1off* and *CS1on*. *CS1* can also reboot (*CS1reboot*, *CS1endreboot*). When *CS1* is operational again, it orders to the switch to produce an observable event (emission of *CS1ok* to *SW1*). If *SW1* has a problem, *CS1* is able to detect it (reception of *SW1to_reboot*) and performs the reinitialisation of the switch by the emission of *SW1reboot* except if *CS1* is not operational.

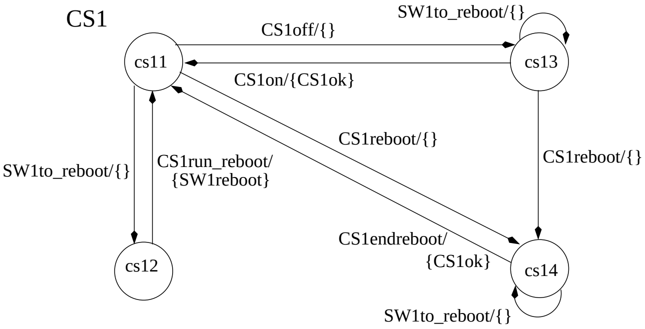

Figure 2. Model of the component *CS1*.

The model of the system is described in a modular way by the models of its components.

Definition 2 *The* model *of a system is a set of component models $\Gamma \triangleq \{\Gamma_1, \ldots, \Gamma_n\}$ such that:*

1. $\forall i, j \in \{1, \ldots, n\}, i \neq j, \Sigma^i_{trg} \cap \Sigma^j_{trg} = \emptyset$;
2. $\forall i, j \in \{1, \ldots, n\}, i \neq j, \Sigma^i_{emit} \cap \Sigma^j_{emit} = \emptyset$;
3. $\forall i, \forall e \in \Sigma^i_{intemtd}, \exists j, j \neq i | e \in \Sigma^j_{intrcvd}$;
4. $\forall i, \forall e \in \Sigma^i_{intrcvd}, \exists j, j \neq i | e \in \Sigma^j_{intemtd}$.

Condition 1 means that every event, triggering a change of state in a component (either an exogenous event or an internal one), cannot be a triggering event of another component. In particular, a failure event (exogenous event) can only occur on one component. Condition 2 means that every event, emitted by one component (either an observable event or an internal one), cannot be emitted by another one. Conditions 3 and 4 guarantee the validity of the structural links between components:

- every internal event e emitted by one component is received by another one (condition 3);
- every internal event e received by one component is emitted by another one (condition 4).

2.2. Semantic of the model

The behaviour of the system can be represented by one automaton called the *global model*. This automaton could be explicitly built by composing the automata of its components. The composition operation is based on a transition system product [1]. In order to simply express the definition of this product, some *null transitions* (noted $q \xrightarrow{\mathbf{e}|\{\}} q$, where $\mathbf{e}$ is the *null event*) are added to each state q of each component model. Such a transition means that a component can stay on a given state while states of some other components change.

Definition 3 *The* free product *of m communicating automata $T_i = (I_i, O_i, Q_i, E_i, q_{0i}), i \in \{1, \ldots, m\}$ is the communicating automaton $\langle T_1, \ldots, T_m \rangle = (I, O, Q, E, (q_{01}, \ldots, q_{0m}))$ such that :*

- $I = I_1 \times \ldots \times I_m$; $O = O_1 \times \ldots \times O_m$;
- $Q = Q_1 \times \ldots \times Q_m$ *is the set of states ;*
- $E = E_1 \times \ldots \times E_m$ *is the set of transitions*

$$(q_1, \ldots, q_m) \xrightarrow{(t_1, \ldots, t_m)} (q'_1, \ldots, q'_m) =$$

$$(q_1 \xrightarrow{t_1} q'_1, \ldots, q_m \xrightarrow{t_m} q'_m).$$

The system reacts to one exogenous event at the same time (event from the set $\Sigma_{exo} \triangleq \bigcup_{i=1}^{n} \Sigma^i_{exo}$). This reaction is represented in the model by the trigger of a set of transitions from different models of components. The transitions of this set are synchronised relying on the internal events emitted and received by those transitions. Here is the formal definition of *synchronised transitions*.

Definition 4 *A transition $q \xrightarrow{t} q' = (q_1 \xrightarrow{t_1} q'_1, \ldots, q_n \xrightarrow{t_n} q'_n)$ of the product $\langle \Gamma_1, \ldots, \Gamma_n \rangle$ is* synchronised *iff:*

1. *$\exists ! j \in \{1, \ldots, n\}$ such that $trg(t_j) \in \Sigma_{exo}$;*
2. *$\forall j \in \{1, \ldots, n\}$ such that $trg(t_j) \neq \mathbf{e}, \forall e \in emit(t_j), \exists k \in \{1, \ldots, n\}, e = trg(t_k)$.*

A synchronised transition formally defines the propagation of events between components of the model. Condition 1 allows that only one exogenous event can be triggered at a given time. In the condition 2, every internal event emitted by an elementary transition $q_j \xrightarrow{t_j} q'_j$ is also an internal event received by another elementary transition $q_k \xrightarrow{t_k} q'_k, j \neq k$. The global model can be then formally defined.

Definition 5 *The* global model $||\Gamma||$ *is the communicating automaton subpart of $\langle \Gamma_1, \ldots, \Gamma_n \rangle$ which only contains the set of synchronised transitions.*

The *global model* defines the state space of the system. A *behaviour* of the system can be represented as a transition path from the initial state $(q_{01}, \ldots, q_{0n})$ to another state of $||\Gamma||$. As far as the described example is concerned, its global model contains 8000 states and 76000 transitions.

3. Diagnosis of the system

An *observation* is the occurrence of an observable event of $\Sigma_{obs} = \bigcup_{i=1}^{n} \Sigma_{obs}^{i}$. In the following, we consider as known a sequence O of observations (totally ordered set of observations).

Thus, the problem of failure diagnosis can be defined as follows. A behaviour of the system is a path of $||\Gamma||$:

$$q_0 \overset{(t_{01}, \ldots, t_{0n})}{\longrightarrow} q_1 \ldots q_m \overset{(t_{m1}, \ldots, t_{mn})}{\longrightarrow} q_{m+1}.$$

O is explained by such a behaviour if O can be expressed as a sequence $\sigma_0 \sigma_1 \ldots \sigma_m$ where each σ_i is a sequence of observations such that: $|\sigma_i| = |\bigcup_{j=1}^{n} obs(t_{ij})| \wedge (\forall o \in \bigcup_{j=1}^{n} obs(t_{ij}), o \in \sigma_i)$. This definition can also be seen as a composition rule between the observation sequence and the model of the system, as it is defined in [5].

Definition 6 *The diagnosis of $\Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$ is the set of behaviours which explain the sequence of observations O from the initial state $(q_{01}, \ldots, q_{0n})$.*

Because of the distributed nature of the diagnosed systems, a lot of events (exogenous events) might occur in a concurrent way. From a diagnosis point of view, if we know that a failure event f_1 has occurred before a failure event f_2, and that f_1 and f_2 are independent, then we are sure that the behaviour obtained by swapping f_1 and f_2 is also an explanation. In this case, knowing that f_1 and f_2 have occurred both is sufficent, the order of their occurrence being not important. This is the reason why a *reduced* representation of the diagnosis is introduced. Using a reduced representation has two objectives:

1. to have a compact representation of the diagnosis;
2. to increase the efficiency of the diagnosis computation.

This reduction is based on a partial order reduction method [9] which is briefly presented in the following section (for more details see [8, 9, 4]).

3.1. Partial order reduction

We call an *action* a transition label from $||\Gamma||$. For example, if $q \overset{t=(t_1, \ldots, t_n)}{\longrightarrow} q'$ is a transition of $||\Gamma||$ then $t = (t_1, \ldots, t_n)$ is the action associated to the transition. The set of $||\Gamma||$ actions is noted A_Γ. We also note en_q the set of transitions that can be triggered from the state q.

Definition 7 *Two actions t_1 and t_2 from A_Γ are* independent *in $||\Gamma|| = (I, O, Q, E, q_0)$ iff $\forall q \in Q$, if $t_1, t_2 \in en_q$ then:*

1. *$t_1 \in en_{q'}$ where $q \xrightarrow{t_2} q' \in E$;*
2. *$\exists q', q'', q'''$ such that $q \xrightarrow{t_2} q' \xrightarrow{t_1} q'' \in E \wedge q \xrightarrow{t_1} q''' \xrightarrow{t_2} q'' \in E$.*

Intuitively, two actions are independent if the occurrence of one of them does not affect the occurrence of the other one (condition 1). Moreover, the order in which those actions can occur does not change the state after both occurrences (condition 2).

Definition 8 *A* dependence relation *D is a binary relation that is reflexive ($\forall t, (t, t) \in D$), symmetric ($\forall t_1, t_2, (t_1, t_2) \in D \Rightarrow (t_2, t_1) \in D$), and such that t_1 and t_2 are independent for any t_1, t_2, $(t_1, t_2) \notin D$.*

This relation allows to define an equivalent relation between sequences of actions. Given two finite sequences v, w of actions from $A_\Gamma^\star$, $v \equiv_D w$ iff there exists a set of sequences $\{u_0, \ldots, u_m\}$ such that $v = u_0$, $w = u_m$ and $\forall i \in \{0, \ldots, m-1\}, u_i = \overline{u} t_1 t_2 \widehat{u} \wedge u_{i+1} = \overline{u} t_2 t_1 \widehat{u}$ where $\overline{u}, \widehat{u} \in A_\Gamma^\star$ and $(t_1, t_2) \notin D$. This equivalence relation can be easily extended to infinite sequences. This extended relation (for the finite and infinite cases) is called the *partial order relation* $\equiv_D$.

Definition 9 *Given a dependence relation D, a* trace *is an equivalence class of sequences defined by the relation $\equiv_D$.*

Thus, a trace represents a set of sequences. Each sequence of the class can be obtained from another one by simply swapping the order of adjacent and independent actions. If s is such a sequence, we note by $[s]_D$ the corresponding trace in which s is included.

3.2. Diagnosis representation

The diagnosis must represent a set of action sequences, so the idea is to only keep one sequence of each trace that must be represented in a given diagnosis. A dependence relation D_Γ between transition labels from $||\Gamma||$ has to be defined.

Definition 10 *Given* $t_1 = (t_{11}, \ldots, t_{1n})$ *and* $t_2 = (t_{21}, \ldots, t_{2n})$ *in* A_Γ, $(t_1, t_2) \in D$ *iff one of the following conditions holds:*

1. $\exists i \in \{1, \ldots, n\}$ *such that* $t_{1i} \neq \mathbf{e}|\{\} \wedge t_{2i} \neq \mathbf{e}|\{\}$;
2. $\bigcup_{i=1}^{n} obs(t_{1i}) \neq \emptyset \wedge \bigcup_{i=1}^{n} obs(t_{2i}) \neq \emptyset \wedge \bigcup_{i=1}^{n} obs(t_{1i}) \neq \bigcup_{i=1}^{n} obs(t_{2i})$.

Intuitively, the relation D_Γ describes two criteria of dependence between two transition labels t_1 and t_2. Condition 1 says that if both t_1 and t_2 affect one component Γ_i at least ($t_{1i} \neq \mathbf{e}|\{\} \wedge t_{2i} \neq \mathbf{e}|\{\}$), they are dependent. Condition 2 says that, if t_1 and t_2 emit a different set of observable events, they are also dependent from a diagnosis point of view.

Proposition 1 *The relation* D_Γ *is a dependence relation.*

Idea of the proof: By definition, D_Γ is symmetric and reflexive. Thus, we have to prove that for any $(t_1, t_2) \notin D_\Gamma$, t_1 and t_2 are independent (see definition 7). By definition, D_Γ guarantees that two actions t_1 and t_2 with $(t_1, t_2) \notin D_\Gamma$ cannot affect the same components which is sufficient to garantuee the criteria of independence. □

Remark 1 *The relation* D_Γ *is not the unique dependence relation. There are more accurate dependence relations. Nevertheless, the advantage of* D_Γ *is the low cost for checking the dependency of two actions.*

Given the dependence relation D_Γ and the sequence of observations O, the reduced representation of the diagnosis of Γ is defined as follows.

Definition 11 *The* reduced representation *of the diagnosis of* Γ *is the communicating automaton* $\Delta_\Gamma(O) = (I, O, Q', E', X_0)$ *such that:*

- $||\Gamma|| = (I, O, Q, E, q_0)$;
- $Q' \subseteq Q \times Pr(O)$ *is the set of states, every state associating a state of* $||\Gamma||$ *with a prefix sequence of* O *explained in this state;*
- $X_0 = (q_0, \epsilon)$ *is the initial state where* ϵ *is the empty sequence;*
- E' *is the set of transitions:* $\forall (q_1, O_1) \xrightarrow{t} (q_2, O_2) \in E', \exists q_1 \xrightarrow{t} q_2 \in E$. *Every trace* $[t_1, \ldots, t_m]_{D_\Gamma}$ *of the diagnosis is represented by one path* $X_0 \xrightarrow{t_1} X_1 \ldots X_{m-1} \xrightarrow{t_m} X_m$ *with* $X_i \xrightarrow{t_{i+1}} X_{i+1} \in E'$.

In figure 3, a part of the diagnosis of the example is presented where the sequence of observations is *SW1down*, *cn12*. For the sake of simplicity, displayed labels are summarised by their exogenous event *exo* and their observable event set *obs* and noted: *exo / obs*. Each state is labelled with a global state of the system (*csij* is the state *j* of *CSi*, *swij* is the state *j* of *SWi* and *cnikj* is the state *j* of *cnik*). In this example, three traces are represented (there are three paths from the initial state (marked with an arrow) to a state which explains O (marked with a doubled box)). In each trace, *CS3reboot/*$\{\}$ occurs: *CS3reboot/*$\{\}$ is independent from the others (the reboot of the control station *CS3* is totally independent from the breakdown of *SW1* and the problem occurring on the connection between *SW1* and *SW2* according to the relation D_Γ). Therefore, each trace represents a set of sequences: the difference between each represented sequence is the moment of the *CS3reboot/*$\{\}$ occurrence. This example represents 18 possible explanations of O.

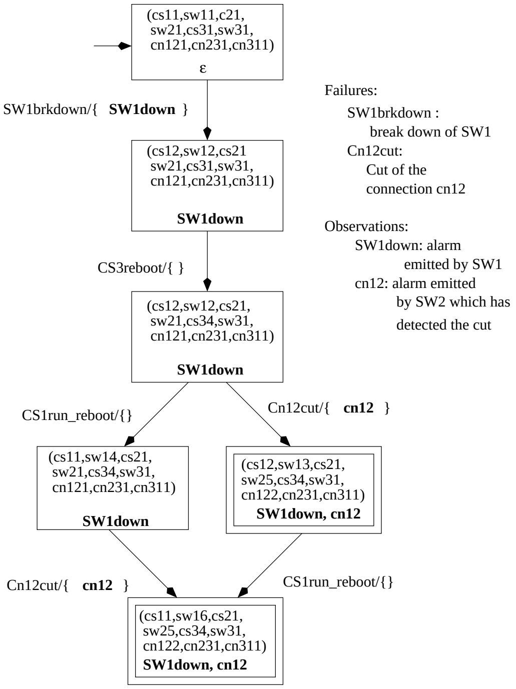

Figure 3. Part of the diagnosis explaining $O = SW1down, cn12$.

Remark 2 *There are several reduced representation of a diagnosis: it is due to the fact that any sequence of a trace is a good candidate for representing the trace.*

Scheme of the algorithm The diagnosis algorithm is based on the *sleep set* algorithm [9]. Given the set of components $\Gamma = \{\Gamma_1, \ldots, \Gamma_n\}$ and the sequence O of observations, the idea is to compute a set of paths belonging to the state space $||\Gamma||$. The search algorithm is a depth first search algorithm which manages *sleep sets*. When a state is visited, a sleep set is associated; this set contains the set of actions that are independent of actions already visited from this state. Because the actions are independent, a sequence of the same trace has already been computed if such a trace exists.

4. Results

This section presents a comparison between two different representations. The diagnosis of the described example has been computed to explain a scenario consisting of 34 observations. Diagnoses explaining prefix observation sequences of this scenario have also been computed in order to show the evolution of the number of diagnosis states relying on the number of observations (see figure4). The non reduced representation has been obtained by the use of a dependence relation where every label is dependent from the others (non independency detection, a trace represents only one sequence).

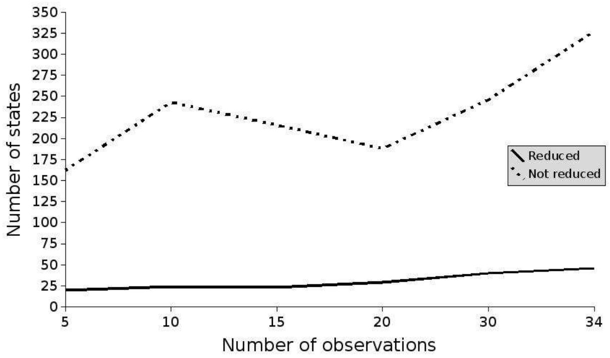

Figure 4. Comparison Reduced/Not reduced

5. Discussion

This paper presents a way to compute failure propagation diagnoses taking into account the well-known problem of *state explosion*. When dealing with large discrete event systems, we have to consider that the components might have concurrent behaviours. By the use of partial order reduction techniques, it is possible to detect such concurrencies and to compute and represent the resulting diagnosis in a more efficient way.

This approach can be updated to be used in several previous discrete event system frameworks [12, 2, 5, 10] where independency is not considered: it suffices to define a dependence relation. In particular, mixing partial order reduction approaches and decentralised approaches [2, 10] seems to give promising results.

Another way to solve the diagnosis representation problem could be the use of symbolic representation (binary decision diagrams) which are well suited for representing *belief states* and provide a good solution to the state explosion problem. Such a representation, intensively used in model checking [4] and planning [3] research areas, is a future way of investigation.

References

[1] A. Arnold. Transition systems and concurrent processes. In *Mathematical problems in Computation theory*, volume 21. Banach Center, 1987.

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

[3] P. Bertoli, A. Cimatti, J. Slaney, and S. Thiébaux. Solving power suplly restoration problems with planning via symbolic model-checking. In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02)*, pages 576–580, 2002.

[4] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

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

[6] 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, 2000.

[7] E. Fabre, A. Benveniste, and C. Jard. Distributed diagnosis for large discrete event dynamic systems. In *Proceedings of the IFAC world congress*, Barcelona, Spain, July 2002.

[8] A. Mazurkiewicz. Trace theory. In *Petri Nets: Applications and relationships to Other Models of Concurrency, Advances in Petri Nets*, number 255 in Lecture Notes in Computer Science, pages 279–324, 1986.

[9] D. Peled. All from one, one for all: on model checking using representatives. In *Proceedings of the 5th International Conference on Computer Aided Verification (CAV'93)*, number 697 in Lecture Notes in Computer Science, pages 409–423. Springer-Verlag, 1993.

[10] Y. Pencolé, M.-O. Cordier, and L. Rozé. A decentralized model-based diagnostic tool for complex systems. *International Journal on Artificial Intelligence Tools (IJAIT)*, 11(3):327–346, September 2002.

[11] L. Rozé and M.-O. Cordier. Diagnosis of discrete-event systems: Extending the diagnoser approach to deal with telecommunication networks. *Discrete Event Dynamic Sytems: Theory and Applications*, 12:43–81, 2002.

[12] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Active diagnosis of discrete-event systems. *IEEE Transactions on Automatic Control*, 43(7):908–929, 1998.