Generalizing diagnosability definition and checking for open systems: a Game structure approach

Tarek Melliti, Philippe Dague

Generalizing diagnosability definition and checking for open systems: a Game structure approach

Tarek Melliti [1], **Philippe Dague** [2]

[1] *IBISC, Univ. d'Evry Val d'Essonne, France*
(Tel: 33 1 60 87 39 36; e-mail: tmelliti@ibisc.fr)

[2] *LRI, Univ. Paris-Sud, CNRS, and INRIA Saclay-Ile de France*
(Tel: 33 1 69 72 92 59 93; e-mail: philippe.dague@lri.fr).

ABSTRACT

Diagnosability is the property of a partially observable system with a given set of possible faults, that these faults can be detected with certainty with a finite observation. Usually, the definition and the verification methods of diagnosability ignore the nature of the system events, controllable (by the system) or uncontrollable. In this paper we show the influence of controllability of events on the diagnosability definition and verification. We show that the classical diagnosability is a special case where we consider the whole system as controllable. Using Game Structure we generalize the definition of diagnosability by the mean of strategies. Then, Alternating-time Temporal Logic is used in order to model check diagnosability in the case of uncontrollable events. We show how the framework is suitable for one system and also for a set of interacting systems.

1 INTRODUCTION

Diagnosis of systems is concerned by two activities: (i) fault detection, i.e. "did a fault happen?" and (ii) fault identification i.e. "which kind of fault did happen?". In real life the operator in charge of diagnosing a failed system, can do mainly two activities in order to figure out what is wrong within a system: (*scenario 1*) she can try to make the diagnosis by only observing the current state of the system (measuring) and/or its history (passive diagnosis); (*scenario 2*) for some kind of systems, she may try some commands on the system and then she observes its reactions in order to establish her diagnosis (active diagnosis). According to these two scenarios, we can classify the systems to be diagnosed in mainly two categories: closed systems, that allow only observation and open systems that allow some interaction with the system.

An important requirement, when designing a system, is how accurate will be the diagnosis of some faults. The notion of "diagnosability" captures that requirement. The system is diagnosable if we can establish a precise diagnosis for every given possible fault from finite observation.

Model-based diagnosis aims at automating the process of diagnosis and diagnosability checking by analyzing an abstract representation of the system called the model.

In 1995, (Sampath *et al.*, 1995) proposed a formal definition of diagnosability for discrete event systems modeled using automata. The automata used have two types of events: observable and unobservable (containing fault events). This definition considers a system as nondiagnosable if its model contains two infinite executions producing the same observable trace and only one of them contains the fault. This implicitly supposes that the system has the total control on its execution making it possible to keep the ambiguity indefinitely. So, the diagnosability is defined by supposing that the diagnoser will only observe what happens in the system with no control in order to try to resolve the ambiguity (scenario 1).

But actually more and more applications are open systems where components and devices interact. Diagnosability has to deal with this kind of systems.

In this work we generalize the definition of diagnosability for any type of system (open or closed). We call this *active diagnosability*. We also propose a method to verify active diagnosability using a model checking approach.

In the sequel of the paper, after some preliminaries, we recall the classical definition of diagnosability and we present the twin plant approach as a method to check diagnosability. In section 3 we introduce the notion of open and well-controllable systems, we also define a suitable game structure for diagnosis and we give the definition of active diagnosability. In section 4 we propose the use of alternating-time temporal logic to model check active diagnosability and we give the correspondent formula. We show also how we can extend active diagnosability to a set of interacting systems. Sections 5 and 6 compare our work to the literature and conclude by some future work. To illustrate our work a toy example is used.

This work is supported by the project PERvasive Service cOmposition (PERSO) of the French National Agency for Research.

2 BACKGROUND ON CLASSICAL DIAGNOSABILITY

2.1 Preliminaries and notation

When dealing with discrete event systems diagnosis, systems are most often modeled by the way of Labeled Transition Systems (LTS).

Definition 1 (LTS) *A labeled transition system* $A = \langle Q, q_0, L, T\rangle$ *is a tuple where*

- Q *represents a set of states*
- $q_0 \in Q$ *a state considered as initial*
- L *a set of events*
- $T \subseteq Q \times L \times Q$ *is the finite branching transition relation which represents a discrete dynamics of a system. We note by* $q \xrightarrow{a} q'$ *for* $(q, a, q') \in T$.

The set of events L is partitioned into two disjoint sets L_o and L_{uo}, which state for the set of observable events and the set of non observable events. Moreover among the set L_{uo} we distinguish a non empty subset, L_f, which represents the set of failure events.

Definition 2 *Let* A *be a LTS, then*

- *A* $path$ *in* A *is a sequence* $\pi = q_0 a_0 q_1 \dots q_n$, *where* n *can be infinite, such that for all* $0 \leq i \leq n-1$ *we have* $q_i \xrightarrow{a_i} q_{i+1}$. *We denote by* $paths(q)$ *the set of all paths that start from the state* $q \in Q$ *and by* $paths(A)$ *the set of all paths in* A, *i.e.* $paths(A) = paths(q_0)$. *We write* $q \in \pi$ *(resp* $a \in \pi$*) for denoting that the state* q *(resp the action* a*) belongs to the sequence* $q_0 a_0 q_1 \dots q_n$. *Moreover, we identify the* i^{th} *state in the path* π *as* $\pi\lceil i]$ *and by* $|\pi| = n+1$ *the amount of states in* π. *We use* $\pi[0...i]$ *to denote the sub-path of* π *that ends with the state* q_i.
- *The* $trace$ σ *of a path* π, *denoted* $trace(\pi)$, *is the sequence* $\sigma = a_0 a_1 \dots a_{n-1}$ *of events in* L *occurring in* π. *We write* $traces(A) = \{trace(\pi) \mid \pi \in paths(A)\}$ *for the set of all traces in* A. *In case* σ *is finite, with* $|\sigma|$ *we denote the number of events occurring in the trace* σ, *i.e.* $|\sigma| = n$. *We use* $\sigma\lceil L'$, *for some* $L' \subseteq L$, *to represent the restriction of the trace* σ *to the set of actions in* L'.
- *We extend the transition relation to traces,* $q \xrightarrow{\sigma} q'$ *if the state* q' *can be reached from state* q *via the trace* σ, *i.e. if there is a path* $\pi \in paths(q)$ *ending at* q' *such that* $trace(\pi) = \sigma$. *We write* $q \rightarrow q'$, *if there exists a trace* σ *such that* $q \xrightarrow{\sigma} q'$ *and* $q \rightarrow$, *if there exists a state* q' *such that* $q \rightarrow q'$.
- *Given any trace* $\sigma \in traces(A)$, *we denote by* $\hat{\sigma}$ *its prefix-closure, and by* $\check{\sigma}$ *its postlanguage, i.e.* $\check{\sigma} = \{\rho \in traces(A) \mid \sigma \in \hat{\rho}\}$. *Moreover, for a given natural number* $k \in \mathbb{N}$ *we denote by* $\check{\sigma}^k$ *its postlanguage with only words with length longer than* k, *i.e.* $\check{\sigma}^k = \{\rho \in \check{\sigma} \mid |\sigma| + k \leq |\rho|\}$.
- *Given a fault event* f, *we denote by* $traces^f(A)$ *the set of traces in* A *that end with a* f *event, i.e.* $traces^f(A) = \{\sigma \in traces(A) | \sigma \in L^*.f\}$
- *Given a fault event* f *and a natural number* $k \in \mathbb{N}$ *we denote by* $traces^{f,k}(A)$, *the set of traces* σ' *such that it exists another trace* σ *that ends in* f *and* σ' *is an extension of* σ *with length longer or equal to the length of* σ *plus* k, *i.e.* $traces^{f,k}(A) = \{\sigma' \in traces(A) \mid \exists\ \sigma \in traces^f(A) \wedge \sigma' \in \check{\sigma}^k\}$.

We say that a system A is alive if for any state there exists a transition initiated in that state, and convergent if it does not have infinite traces made up of unobservable actions. In the remaining of the paper we consider only systems which are alive and convergent.

Example 1 *figure1 represents a system where* $L_o = \{a, b, c, d\}$, $L_{uo} = \{u_1, u_2, f\}$, $L_f = \{f\}$

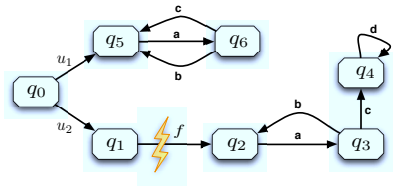

Figure 1: A system containing one fault f

2.2 Diagnosability definition and verification

The classical diagnosability is a property defined on the paths of the system. It states that each time a fault may happen, it exists a finite window of observations that allows us to decide wether this fault did happen or not (Sampath *et al.*, 1995).

Definition 3 (Diagnosability) *Let* A *be a system and* $f_i \in L_f$, *then* f_i *is diagnosable in* A *(or A is* f_i*-diagnosable) iff*
$\exists\ n_i \in \mathbb{N} : \forall\ \sigma \in traces^{f_i}(A) : \forall\ \rho \in \check{\sigma}^{n_i}$: $\forall \alpha \in traces(A)$: $\rho\lceil L_o = \alpha\lceil L_o$ *implies* $f_i \in \alpha$
Otherwise f_i *is said nondiagnosable in* A *(or A* f_i*-nondiagnosable). A system is said to be diagnosable if all its faults are so and nondiagnosable otherwise.*

If a fault is diagnosable then a diagnostic algorithm can decide of its occurrence or not with certainty based on a finite sequence of observations. Diagnosability checking methods consist in proving that the system is not nondiagnosable. This requires the search for infinite traces ρ and ρ', with $\rho\lceil L_o = \rho'\lceil L_o$ such that f appears only in one of them. The two traces ρ and ρ' are called a critical pair (Cimatti *et al.*, 2003). Many algorithms and technics are proposed to check diagnosability, we consider here the twin plant approach (Cimatti *et al.*, 2003) because it is the most appropriate to present our work. The twin plan approach consists in two steps: (i) building a diagnoser of the system; (ii) then comparing two copies of the diagnoser by a synchronous product. The diagnoser construction is inspired by the observer of a system (Sampath *et al.*, 1995), by keeping only the states of the system which are reachable by at least one observable event. These states are enriched by the set of fault events encountered during the reaching process.

Definition 4 *Let A be a system to be diagnosed,its diagnoser is also a LTS noted $\overline{A} = \langle \overline{Q}, \overline{L}, \overline{q_0}, \overline{T} \rangle$ with:*

- $\overline{Q} \subseteq Q_o \times 2^{L_f}$, *with* $Q_o = \{q_0\} \cup \{q | \exists a \in L_o, q' \in Q$ *s.t.* $(q', a, q) \in T\}$
- $\overline{L} = L_o$
- $\overline{q_0} = (q_0, \emptyset)$
- $\overline{T} \subseteq \overline{Q} \times \overline{L} \times \overline{Q}$ *is the transition set* $(q, \mathcal{F}) \xrightarrow{a} (q', \mathcal{F}')$ *s.t.* $q \xrightarrow{\sigma a} q'$ *with* $\sigma \in L_{uo}^*$, $a \in L_o$, $\mathcal{F}' = \mathcal{F} \cup \{f_i | f_i \in \sigma\}$

Example 2 *Figure 2 represents the diagnoser of the system presented in figure 1.*

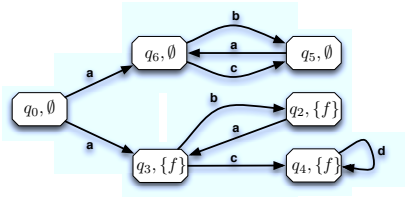

Figure 2: The diagnoser of the system in figure 1

The second step of the twin plant method is to build a machine that compares every pair of paths (ρ, ρ') in the system that have the same observable behavior. Such comparison is done by computing the synchronous product of two instances of the diagnoser $\overline{A}$. As in (Schumann and Pencolé, 2007) we denote these two instances of $\overline{A}$ respectively by *left* $(\overline{l : A})$ and *right* $(\overline{r : A})$ and we distinguish between their states by using respectively the prefixes l : and r :. The synchronous composition used for the twin plant is the classical synchronous product of $n > 1$ automata, noted $(A_1 ... \| ... A_n) \backslash \Sigma$. The states of the resulted automaton form a subset of the cartesian product $\underset{i=1...n}{\times} Q_i$. The transitions of the product are constructed by allowing only simultaneous transitions for events in Σ and individual evolutions otherwise.

Proposition 1 (f-nondiagnosable state, system)
Let A be a system and $\overline{l : A}$, $\overline{r : A}$ two copies of its diagnoser.

f-nondiagnosable state *A state $(l : (q_i, \mathcal{F}_i), r : (q_j, \mathcal{F}_j))$ in the synchronous product $(\overline{l : A} \| \overline{r : A}) \backslash L_o$ is called* f-nondiagnosable *iff $f \in (\mathcal{F}_i \cup \mathcal{F}_j) \setminus (\mathcal{F}_i \cap \mathcal{F}_j)$. Otherwise the state is* f-diagnosable.

f-nondiagnosable system *The system A is* f-nondiagnosable *iff it exists in $(\overline{l : A} \| \overline{r : A}) \backslash L_o$ a cycle composed only by* f-nondiagnosable *states. Otherwise the system is* f-diagnosable.

Example 3 *Figure 3 represents the twin plant product of the two instances of the diagnoser of the figure 2. According to the proposition 1 we can see that the fault f is nondiagnosable because we observe a cycle $[(l : q_6, \emptyset), (r : q_3, \{f\})] \xrightarrow{b} [(l : q_5, \emptyset), (r : q_2, \{f\})] \xrightarrow{a} [(l : q_6, \emptyset), (r : q_3, \{f\})]$... of f-nondiagnosable states. This proves the existence of two infinite paths that have the same observable trace, $(ab)^\infty$, where only one of them contains the fault f.*

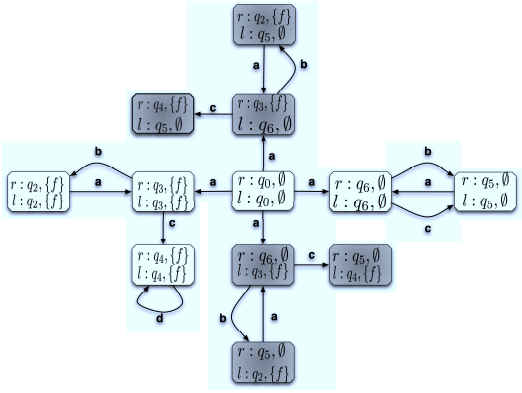

Figure 3: The twin plant of the diagnoser of figure 2

Note that in the example 3 the synchronous product is finite while the systems are alive. This, as proved in (Cimatti *et al.*, 2003), does not influence the decision. In the same paper the authors propose an idea to deal with these blocking states. We show in this paper that we can handle this problem in a simple and elegant way and also by benefiting from the blocking states.

3 DIAGNOSABILITY OF OPEN SYSTEMS

Let A be a system to be diagnosed. We split its actions L into two disjoint subsets: the controllable actions L^c and the uncontrollable actions L^{uc} where $L_{uo} \subseteq L^c$. The notion of controllable here is viewed from the point of view of the system. A system controls an action if it decides of its occurrences; at the opposite an action is uncontrollable if the system undergoes its occurrences. A system with $L^{uc} = \emptyset$ is called closed and open otherwise. We extend the controllability to the states as follows $Q^c = \{q \in Q | \exists a \in L^c$ s.t. $q \xrightarrow{a}\}$ and $Q^{uc} = \{q \in Q | \exists a \in L^{uc}, q \xrightarrow{a}\}$. Note that the two sets are not necessarily disjoint.

Definition 5 (Open and well-controllable Systems)
Let A be a system and L^{uc}, L^c be respectively the set of uncontrollable and controllable actions. We say that the system is open, respectively well-controllable, iff $L^{uc} \neq \emptyset$, resp. $Q^c \cap Q^{uc} = \emptyset$

Example 4 *The system of the figure 1 is nondiagnosable because the environment does not control any action. This allows the system to stay in the $(ab)^*$ trace with uncertainty about the fault f. By considering $L^{uc} = \{b, c\}$, we can see that the system becomes diagnosable. This is because at each time we can take the system out from its $(ab)^*$ trace by enforcing it to execute the action c. One can note that after action c the system will converge, within a finite set of actions, in a situation where the fault or the non fault f is certain. This demonstrates that diagnosability as proposed in definition 3 does not stand for open systems.*

To cover open systems as well as closed systems (which can be considered as a degenerated case of open ones) we propose a generalization of the diagnosability definition. For this purpose we use game structure to formalize that generalization and an adapted temporal logic, Alternating-Time Temporal Logic, in order to check it.

3.1 Game structure for active diagnosability

Let P be a set of propositions. Conceptually, we are dealing with a set Θ of n players, where each player is represented by an alive and well-controllable LTS $\theta_i = \langle Q_i, q_{i0}, L_i, T_i, \mu_i : Q_i \rightarrow 2^P \rangle$. We suppose that for any player θ_i, all its uncontrolled (thus observable) actions are controlled by at least another player, $L_i^{uc} \subset \bigcup_{j \neq i} L_j^c$.

Definition 6 (Round-based game structure) *A game structure between a set of players Θ is a tuple $G^\Theta = \langle \mathcal{C}, c_0, P, M, \delta, \mu \rangle$ where:*

- *$\mathcal{C}$ is a non empty set of configurations with $\mathcal{C} \subset \times_{i=1...n} Q_i$. We range over it using c_i*
- *$c_0 \in \mathcal{C}$ is the initial configuration, $c_0 = \langle q_{10}, ..., q_{n0} \rangle$*
- *P is a non empty set of propositions*
- *$M \subset \times_{i=1...n} (L_i \cup \{\epsilon\})$ where ϵ stands for the non-event. We range over it using m_i (for moves)*
- *$\delta \subseteq \mathcal{C} \times M \times \mathcal{C}$*
- *$\mu : \mathcal{C} \rightarrow 2^P$*

δ encodes the rules of the game. There are many dynamics for game structures which differ by the definition of δ (Alur *et al.*, 1997). We define here a game structure suitable for open and well-controllable systems. In the initial configuration all players are in their initial state. The game consists in a sequence of rounds where each round is played in three steps. In [Step 1] all the players that are in a controlled state (active ones) can choose one among all their possible transitions from their current states. In [Step 2] all the players which are in an uncontrollable state (passive ones) determine their reactions by choosing one of their uncontrolled actions which have been chosen by at least one of the active players in the [Step 1][1]. [Step 3] consists in computing the next configuration according to the following rules: (i) an active player, which action was chosen by at least one passive player, moves to one of the possible states reachable by that action, otherwise it remains in the same state by executing ϵ; (ii) if none of the possible actions of a passive player was chosen in [Step 1] then it remains in the same state by executing ϵ. Let c be a configuration, the game can move in each round to one of the possible next configurations allowed by local choices of each player. For some player θ_i we note by $Out(q_i) = \{(a_i, q_i') \in L_i \times Q_i : q_i \xrightarrow{a_i} q_i' \in T_i\}$ the set of the successor states of q_i. We note also by $Out(q_i, a_i)$ the restriction of $Out(q_i)$ to transitions labeled with a_i. We have a transition from a source configuration c_s to a target configuration c_t by a labeled move m_l i.e. $c_s = \langle q_{1s}, \ldots, q_{ns} \rangle \xrightarrow{m_l = \langle a_{1l}, \ldots, a_{nl} \rangle} c_t = \langle q_{1t}', \ldots, q_{nt}' \rangle \in \delta$ iff:

- $\forall i = 1 \ldots n$, s.t. $q_{is} \in Q_i^c$ we have:
  - $q_{is} = q_{it}' \wedge a_{il} = \epsilon$ if $\nexists a_{jl}$, s.t. $q_{js} \in Q^{uc} \wedge (q_{js}, a_{jl}, q_{jt}') \in T_j \wedge a_{il} = a_{jl}$
  - $(a_{il}, q_{it}') \in Out(q_{is})$ otherwise

  We note by ℓ_t^s the set of actions chosen by the players in a controllable state to move from a source configuration c_s to a target c_t.
- $\forall i = 1 \ldots n$, s.t. $q_{is} \in Q_i^{uc}$ we have:
  - $q_{is} = q_{it}' \wedge a_{il} = \epsilon$ if $\forall a \in \ell_t^s, Out(q_{is}, a) = \emptyset$
  - $(a_{il}, q_{it}') \in Out(q_i)$ for some $a_{il} \in \ell_t^s$

Definition 7 (Player strategy) *Let θ_i be one of the players in a game structure G^Θ. A strategy of the player is composed by two functions:*

- *$\int_{\theta_i}^c : paths(G^\Theta) \rightarrow L_i \times Q_i$ s.t. $\int_{\theta_i}^c (c_0 m_0 \ldots m_{k-1} c_k) = (a_{ik}, q_{ik+1})$ where $(a_{ik}, q_{ik+1}) \in Out(q_{ik})$ if $q_{ik} \in Q_i^c$ and undefined otherwise.*
- *$\int_{\theta_i}^{uc} : paths(G^\Theta) \rightarrow L_i \times Q_i$ s.t. $\int_{\theta_i}^{uc} (c_0 m_0 \ldots m_{k-1} c_k) = (a_{ik}, q_{ik+1})$ where $a_{ik} \in \ell_{k+1}^k \wedge (a_{ik}, q_{ik+1}) \in Out(q_{ik})$ and (ϵ, q_{ik}) otherwise, if $q_{ik} \in Q_i^{uc}$; undefined if $q_{ik} \in Q_i^c$.*

We denote by $\int_{\theta_i}$ the function defined on $paths(G^\Theta)$ resulting from these two functions.

A strategy $\int_{\theta_i}$ of a player θ_i is a function that, given an execution of the game, decides about the next move of the player (either freely chosen when controllable or constrained as reaction when uncontrollable). A computation of a game structure G^Θ from a configuration $c \in \mathcal{C}$ under the strategy $\int_{\theta_i}$ is a set of valid paths according to the strategy function, i.e. $Comp(c, \int_{\theta_i}) = \{\pi \in paths(c) | \forall k, 0 \leq k \leq |\pi| - 1$ we have $\int_{\theta_i}(\pi[0 \ldots k]) = (a_{ik}, q_{ik+1})\}$. Given $C \subset \Theta$ and a set of strategies, $\int_C$, one for each $\theta \in C$, $Comp(c, \int_C) = \bigcap_{\int_\theta \in \int_C} Comp(c, \int_\theta)$. It is easy to see that $Comp(c, \int_\Theta)$ is a unique path in G^Θ.

3.2 Active diagnosability definition

Let A be an open and well-controllable system to be diagnosed and let us consider a system $A_\mathcal{E} = \langle Q_\mathcal{E}, L_\mathcal{E}, q_{0\mathcal{E}}, T_\mathcal{E} \rangle$ such that :

- $Q_\mathcal{E}$ is a set of states.
- $L_\mathcal{E} = L_o$ with $L_\mathcal{E}^c = L^{uc}$ and $L_\mathcal{E}^{uc} = L^c \cap L_o$.
- $q_{0\mathcal{E}}$ is the initial state.
- $T_\mathcal{E} \subseteq Q_\mathcal{E} \times L_\mathcal{E} \times Q_\mathcal{E}$ is a transition relation.

[1] Note here that the passive players can also have more than one choice (in some way they are thus active too)

The transition relation $T_{\mathcal{E}}$ must be defined in such a way that the resulted game structure from the two systems $A_{\mathcal{E}}$ and A, $G^{A_{\mathcal{E}},A}$, must respect the following conditions:

- $\forall c, c \xrightarrow{m}$ with $m \neq \langle \epsilon, \epsilon \rangle$
- $\forall c_1 = \langle q_{1\mathcal{E}}, q_1 \rangle \xrightarrow{m=\langle a_{1\mathcal{E}}, a_1 \rangle} c_2 = \langle q'_{1\mathcal{E}}, q'_1 \rangle \in \delta$ we have $a_1 \neq \epsilon$
- $c_1 = \langle q_{1\mathcal{E}}, q_1 \rangle \xrightarrow{m=\langle \epsilon, a_1 \rangle} c_2 = \langle q'_{1\mathcal{E}}, q'_1 \rangle \in \delta$ iff $a_1 \in L_{uo}$

The system $A_{\mathcal{E}}$ represents a perfect environment of the system A: (i) the game is never blocked (ii) the environment is always able to observe the observable reactions of the system and always produces at least one of the commands waited by the system (iii) the environment never reacts when an unobservable event is executed by the system.

For a game structure $G^{A_{\mathcal{E}},A}$ we naturally extend, for some fault f, the definitions of traces: $traces^f(G^{A_{\mathcal{E}},A}) = \{\sigma \in traces(c_0) | \sigma$ ends with $\langle \epsilon, f \rangle\}$ and $traces^{f,k}(G^{A_{\mathcal{E}},A}) = \{\sigma' \in traces(G^{A_{\mathcal{E}},A}) \mid \exists\, \sigma \in traces^f(G^{A_{\mathcal{E}},A}) \wedge \sigma' \in \breve{\sigma}^k\}$. We can now give the generalization of diagnosability definition for open and well-controllable systems.

Definition 8 (Active Diagnosability) *Let A be a system to be diagnosed, $A_{\mathcal{E}}$ its environment and $G^{A_{\mathcal{E}},A}$ the game structure involving both of them. The fault $f_i \in L_f$ is actively diagnosable in A iff*

$$\exists\, n_i \in \mathbb{N} : \forall\, \sigma \in traces^{fi}(G^{A_{\mathcal{E}},A}) \text{ s.t. } c_0 \xrightarrow{\sigma} c_f : \exists \int_{A_{\mathcal{E}}} : \forall \rho \in Comp(c_f, \int_{A_{\mathcal{E}}}) \text{ s.t. } \sigma\rho \in \breve{\sigma}^{n_i} : \forall \alpha \in traces(G^{A_{\mathcal{E}},A}) : \sigma\rho \lceil (L_{\mathcal{E}} \times L_o) = \alpha \lceil (L_{\mathcal{E}} \times L_o) \text{ implies } \langle \epsilon, f_i \rangle \in \alpha$$

The definition states the following: for each trace in the game that ends with a fault event, as a move of the system, then the environment has a strategy in such a way that, for any infinite continuation according to that strategy, if there is another execution of the game that produces the same observable moves, this execution should contain the fault. It is easy to verify that if $L^{uc} = \emptyset$ then $Comp(c_f, \int_{A_{\mathcal{E}}}) = traces(c_f)$. By renaming each move $\langle a, b \rangle$ by b we fit exactly the definition 3.

The next section presents a method and a tool in order to model check active diagnosability.

4 ACTIVE DIAGNOSABILITY VERIFICATION USING ATL

In this section we use the Alternating-time Temporal Logic (ATL) in order to check diagnosability of an open and well-wontrollable system. First we recall the Alternating-time Temporal Logic and then we give a logic formula for checking active diagnosability.

4.1 ATL

Alternating-time Temporal Logic ATL was designed to formulate correctness properties for open systems, which have to be proved correct with respect to an arbitrary environment. The environment can be either one or more interacting discrete event systems. As we will show, this problem is very close to the problem of checking active diagnosability. ATL can be seen as an extension of the Computational Tree Logic (CTL) where the universal ($\mathcal{A}$) and existential ($\mathcal{E}$) path quantifiers are parameterized by cooperation modalities between a set of agents in the system. The syntax of an ATL formula is defined recursively over a set P of propositions and a set Θ of players as follows:

$$\psi ::= \top | p | (\psi \wedge \psi) | \neg\psi | \langle\langle C \rangle\rangle \mathrm{X}\psi | \langle\langle C \rangle\rangle \mathrm{G}\psi | \langle\langle C \rangle\rangle (\psi \mathrm{U} \psi)$$
where $C \subset \Theta$

$\top$ stands for True while neXt, Globally, Until are the path temporal operators of CTL. Unlike CTL, these operators are parameterized by a set $\langle\langle C \rangle\rangle$ of players, called a coalition, which means that the players in C can cooperate in such a way that the resulted computation verifies the property considered. The semantic of an ATL formula is provided based on a game structure, and the truth of a formula ψ in a configuration c of a game structure G^{Θ} is defined via the standard clauses of the Boolean connectors and the following clauses for the strategized temporal operators:

- $(G^{\Theta}, c) \vDash \top$
- $(G^{\Theta}, c) \vDash p \Leftrightarrow p \in \mu(c)$ for $p \in P$
- $(G^{\Theta}, c) \vDash \neg\psi \Leftrightarrow (G^{\Theta}, c) \nvDash \psi$
- $(G^{\Theta}, c) \vDash (\psi_1 \wedge \psi_2) \Leftrightarrow (G^{\Theta}, c) \vDash \psi_1 \wedge (G^{\Theta}, c) \vDash \psi_2$
- $(G^{\Theta}, c) \vDash \langle\langle C \rangle\rangle \mathrm{X}\psi \Leftrightarrow \exists \int_C$ s.t. $\forall \pi \in Comp(c, \int_C)$ we have $(G^{\Theta}, \pi[1]) \vDash \psi$
- $(G^{\Theta}, c) \vDash \langle\langle C \rangle\rangle \mathrm{G}\psi \Leftrightarrow \exists \int_C$ s.t. $\forall \pi \in Comp(c, \int_C)$ we have $\forall i, (G^{\Theta}, \pi[i]) \vDash \psi$
- $(G^{\Theta}, c) \vDash \langle\langle C \rangle\rangle (\psi_1 \mathrm{U}\ \psi_2) \Leftrightarrow \exists \int_C$ s.t. $\forall \pi \in Comp(c, \int_C)$, $\exists i \geq 0$ s.t. $\forall j < i$ we have $(G^{\Theta}, \pi[j]) \vDash \psi_1 \wedge (G^{\Theta}, \pi[i]) \vDash \psi_2$

The CTL duality of temporal operators is still valid in ATL; $\langle\langle C \rangle\rangle \mathrm{F}\psi$ stands for $\langle\langle C \rangle\rangle \top \mathrm{U}\ \psi$; we can also express the classical CTL path quantifiers Always and Eventually as follows: $\mathrm{AX}\psi$, $\mathrm{AG}\psi$, $\mathrm{A}(\psi_1 \mathrm{U}\ \psi_2)$ respectively by $\langle\langle \emptyset \rangle\rangle \mathrm{X}\psi$, $\langle\langle \emptyset \rangle\rangle \mathrm{G}\psi$, $\langle\langle \emptyset \rangle\rangle (\psi_1 \mathrm{U} \psi_2)$, and $\mathrm{EX}\psi$, $\mathrm{EG}\psi$, $\mathrm{E}(\psi_1 \mathrm{U}\ \psi_2)$ respectively by $\langle\langle \Theta \rangle\rangle \mathrm{X}\psi$, $\langle\langle \Theta \rangle\rangle \mathrm{G}\psi$, $\langle\langle \Theta \rangle\rangle (\psi_1 \mathrm{U}\ \psi_2)$. We can also introduce the parameterized universal path quantifier, by writing $[[C]]\mathcal{X}\psi$ and $[[C]]\mathrm{G}\psi$ respectively for $\neg\langle\langle C \rangle\rangle \mathrm{X}\neg\psi$ and $\neg\langle\langle C \rangle\rangle \mathrm{F}\neg\psi$. $[[C]]$ expresses the fact that the agent in C cannot avoid paths that verify a given path formula. This implies that the agent in $\Theta \setminus C$ has a strategy to produce only paths validating the path formula (e.g. $[[C]]\mathrm{G}\psi \Leftrightarrow \neg\langle\langle C \rangle\rangle \mathrm{F}\neg\psi$).

4.2 Diagnosability checking for one system

Let A be a system to be diagnosed and let $\overline{A}$ be its diagnoser according to definition 4. When interacting with the system, the environment cannot estimate exactly its real state due to the nondeterminism of $\overline{A}$. It can only have an idea about the minimal set of its actual possible states. Let us consider an environment of $\overline{A}$, noted $A_{\mathcal{E}}$, as exactly the mirror of the diagnoser except that $L^c_{\mathcal{E}} = L^{uc}_o$ and $L^{uc}_{\mathcal{E}} = L^c_o$. As pointed in

(Cimatti *et al.*, 2003), the game structure, $G^{A_\mathcal{E},\overline{A}}$, resulting from definition 6, may have finite moves. So a perfect environment of the system is a system that can avoid such cases and that can also benefit from the information of the dead configurations to have an idea about the real state of the system. We propose here, as we made in (Melliti *et al.*, 2008), to synthesize a perfect environment by extending the transition relation $T_\mathcal{E}$ of $A_\mathcal{E}$ as follows:

$T_\mathcal{E} = T_\mathcal{E} \cup \{(q_{i\mathcal{E}}, a, q'_{i\mathcal{E}}) | q_{i\mathcal{E}} \overset{a}{\nrightarrow} \wedge \exists w \in L^*_\mathcal{E}, q_{j\mathcal{E}} \in Q_\mathcal{E}$ s.t. $i \neq j \wedge q_{0\mathcal{E}} \overset{w}{\longrightarrow} q_{i\mathcal{E}} \wedge q_{0\mathcal{E}} \overset{w}{\longrightarrow} q_{j\mathcal{E}} \wedge (q_{j\mathcal{E}}, a, q'_{i\mathcal{E}}) \in T_\mathcal{E}\}$

The perfect environment enriches its states by transitions of all the states that are reachable from the initial states by the same trace. The meaning of this extension can be interpreted as follows:

- In the case where the system is in a controllable state, the environment will observe its reaction among all possible reactions. This reaction of the system may help the environment to better precise the estimation about the real state of the system.
- In the case where the system is in an uncontrolled state (waiting for a command) the environment will send a command. If the command is not expected by the system, then the environment remains in the same state and tries another command until it works. Note that according to our extension the environment has the minimum set of commands expected by the system.

We denote by $A^+_\mathcal{E}$ the extension of $A_\mathcal{E}$.

Example 5 *Figure 4 represents the perfect environment of $\overline{A}$. The extensions are represented using dashed arcs.*

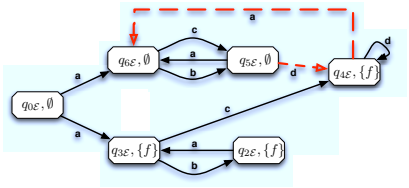

Figure 4: The extended perfect environment $A^+_\mathcal{E}$

To each fault $f_i \in L_f$ we associate the propositions p^{f_i}, $\neg p^{f_i}$ and $\Diamond p^{f_i}$. They respectively mean a fault f_i did happen, did not happen and did possibly happen. The set of fault propositions of a LTS is noted Δ^A.

Consider the game structure $G^{A^+_\mathcal{E},\overline{A}}$. We recall here that the states of $\overline{A}$, respectively of $A^+_\mathcal{E}$, are of the form $(q_i, \mathcal{F}_j)$ (to differentiate them we note them by $(\overline{q_i}, \mathcal{F}_j)$, resp. $(q_{i\mathcal{E}}, \mathcal{F}_j)$, also $\overline{q_i}$, resp. $q_{i\mathcal{E}}$, when $\mathcal{F}_j$ is not relevant).

$G^{A^+_\mathcal{E},\overline{A}}$ can be interpreted as a game where in each configuration the environment has an hypothesis about the states of the system, i.e. the configuration $c = \langle (q_{i\mathcal{E}}, \mathcal{F}), (\overline{q_j}, \mathcal{F}') \rangle$ means that the environment thinks that the system is in the state $(q_i, \mathcal{F})$ while the system is in the state $(q_j, \mathcal{F}')$. The diagnosis problem can be then reduced to the game where the environment wins

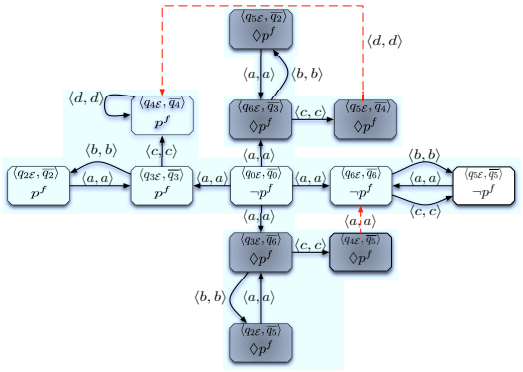

Figure 5: The game structure $G^{A^+_\mathcal{E},\overline{A}}$ between $\overline{A}$ and $A^+_\mathcal{E}$

by determining the state of the system within a finite set of moves. This means that active diagnosability analysis can be reduced to the existence for the environment of a universal strategy to determine, within a finite set of moves, the real state of the system.

We annotate the set $\mathcal{C}$ of configurations of the game $G^{A^+_\mathcal{E},\overline{A}}$ using the set of propositions Δ^A with $\mu : \mathcal{C} \to 2^{\Delta^A}$ s.t. $\mu(\langle (q_{i\mathcal{E}}, \mathcal{F}), (\overline{q_j}, \mathcal{F}') \rangle) = \Delta_1 \cup \Delta_2 \cup \Delta_3$ with:

- $\Delta_1 = \bigcup_{f \in (\mathcal{F} \cap \mathcal{F}')} \{p^f\}$
- $\Delta_2 = \bigcup_{f \in [(\mathcal{F} \cup \mathcal{F}') \setminus (\mathcal{F} \cap \mathcal{F}')]} \{\Diamond p^f\}$
- $\Delta_3 = \bigcup_{f \in [L_f \setminus (\mathcal{F} \cup \mathcal{F}')]} \{\neg p^f\}$

Example 6 *Figure 5 represents $G^{A^+_\mathcal{E},\overline{A}}$ annotated with μ function.*

Based on the definition of the environment $A^+_\mathcal{E}$ and the game structure dynamics we can state the following propositions.

Proposition 2 *Let $G^{A^+_\mathcal{E},\overline{A}}$ resulting from $\overline{A}$ and its perfect environment. The binary relation $R = \{(\langle q_{i\mathcal{E}}, \overline{q_j} \rangle, \overline{q_j})\}$ is a bisimulation (Milner, 1989) modulo renaming function $\mathcal{N} : M \to \overline{L}$ with $\mathcal{N}(\langle a, b \rangle) = b$, i.e. $G^{A^+_\mathcal{E},\overline{A}}[\mathcal{N}] \sim \overline{A}$.*

Proof 1 *Let $c = \langle q_{i\mathcal{E}}, \overline{q_j} \rangle$ be a configuration and $\overline{q_j} \in \overline{Q}$.*

1 *According to the definition of $A^+_\mathcal{E}$ we have $Out(\overline{q_i}) \subseteq Out(q_{i\mathcal{E}})$ i.e. $\forall a \in \overline{L}, \overline{q_i} \overset{a}{\longrightarrow}$ implies $\langle -, \overline{q_i} \rangle \overset{\langle a,a \rangle}{\longrightarrow}$, so implies $\langle -, \overline{q_i} \rangle \overset{a}{\longrightarrow}$ in $G^{A^+_\mathcal{E},\overline{A}}[\mathcal{N}]$.*

2 *A move m, s.t. $\mathcal{N}(m) \neq \epsilon$, is possible from a configuration $c = \langle q_{i\mathcal{E}}, \overline{q_j} \rangle$ iff it is possible from the two states that compose it, i.e. $\forall (c, \mathcal{N}(\langle a, b \rangle) = b, c') \in \delta$ then $\exists (\overline{q_j}, b, \overline{q'_j}) \in \overline{T}$. In addition the reached configuration c' is of the form $c' = \langle -, \overline{q'_j} \rangle$ and as we have $(\langle -, \overline{q'_j} \rangle, \overline{q'_j}) \in R$ we conclude their bisimulation.*

Proposition 3 *Let θ_1, θ_2, θ_3 be three players where $\theta_1 \sim \theta_2$, then we have $G^{\theta_1,\theta_3} \sim G^{\theta_2,\theta_3}$ i.e. the bisimulation is a congruence for the game structure rules operator.*

Proof 2 *The proof is obvious from the definition of the game rules.*

Each configuration c such that $\Diamond f \in \mu(c)$ means that there is an uncertainty about the fault occurrence (same as *f-nondiagnosable* state in the twin plant). The system is diagnosable in that configuration iff the environment can enforce the system to reveal its truth about the occurrence of the fault by leading the game to a configuration c' with $p^f \in \mu(c')$ or $\neg p^f \in \mu(c')$. This can be expressed for a given fault f using ATL formula $Diag^f$ as follows:

$$\boxed{Diag^f \stackrel{def}{\equiv} \mathrm{AG}[\Diamond p^f \Rightarrow \langle\langle A^+_{\mathcal{E}}\rangle\rangle \mathrm{F}(p^f \vee \neg p^f)]}$$

The formula says that each time we reach a configuration where we have a doubt about the occurrence of the fault f, then from that configuration the environment can establish a strategy of commands on the system to enforce it to reveal the truth about the occurrence of f. We can express the same requirement without the need of the environment as follows:

$$\boxed{\ddot{Diag}^f \stackrel{def}{\equiv} \mathrm{AG}[\Diamond p^f \Rightarrow [[\overline{A}]] \mathrm{F}(p^f \vee \neg p^f)]}$$

The system is diagnosable for a fault f iff each time a doubt about f appears, then the system does not have any strategy to keep that doubt infinitely.

Theorem 1 *An open and well-controllable system A is actively diagnosable according to definition 8 iff $\forall f_i \in L_f, G^{A^+_{\mathcal{E}},\overline{A}} \models Diag^{f_i}$.*

This means that a system is actively diagnosable iff its perfect environment has a strategy to prove it.

Proof 3 *Let A be a system and let $G^{A^+_{\mathcal{E}},\overline{A}}$ be the game structure of the system and its perfect environment.*

(if) *Let us suppose that $G^{A^+_{\mathcal{E}},\overline{A}} \models Diag^f$ for some fault f and the system is f-nondiagnosable according to definition 8. This means that:*

- **(1)** $\exists(\overline{q_i},\mathcal{F}_i),(\overline{q_j},\mathcal{F}_j) \in \overline{Q}, \exists\sigma \in L^*_o$ *with* $(\overline{q_0},\emptyset) \xrightarrow{\sigma} (\overline{q_i},\mathcal{F}_i)$ *and* $(\overline{q_0},\emptyset) \xrightarrow{\sigma} (\overline{q_j},\mathcal{F}_j)$ *s.t.* $f \in (\mathcal{F}_i \cup \mathcal{F}_j) \setminus (\mathcal{F}_i \cap \mathcal{F}_j)$. *Let us say $f \in \mathcal{F}_i$ and $f \notin \mathcal{F}_j$.*
- **(2)** $traces((\overline{q_i},\mathcal{F}_i)) = traces((\overline{q_j},\mathcal{F}_j))$ *and* $\forall\pi \in paths((\overline{q_j},\mathcal{F}_j)), \nexists(\overline{q},\mathcal{F}) \in \pi$ *with* $f \in \mathcal{F}$.

Following the game structure of $G^{A^+_{\mathcal{E}},\overline{A}}$, it exists a configuration $c = \langle q_{i\mathcal{E}}, \overline{q_j}\rangle$ s.t. $\Diamond p^f \in \mu(c)$. As c is bisimilar to $\overline{q_j}$, according to the proposition 2, then all reachable configurations c' from c, $c \to c'$, will be of the form $c' = \langle q'_{i\mathcal{E}}, \overline{q'_j}\rangle$ with $q_{i\mathcal{E}} \to q'_{i\mathcal{E}}$ in $A^+_{\mathcal{E}}$ and $\overline{q_j} \to \overline{q'_j}$ in $\overline{A}$, which means, according to (1) and (2), that $\Diamond p^f \in \mu(c')$. We can conclude that: $(G^{A^+_{\mathcal{E}},\overline{A}}, c) \models \mathrm{AG}\Diamond p^f$. Equivalently we have $(G^{A^+_{\mathcal{E}},\overline{A}}, c) \models \langle\langle\emptyset\rangle\rangle \mathrm{G}\Diamond p^f$, i.e. $(G^{A^+_{\mathcal{E}},\overline{A}}, c) \models \neg\langle\langle A^+_{\mathcal{E}}, \overline{A}\rangle\rangle \mathrm{F}(p^f \vee \neg p^f)$. This implies $\neg\langle\langle A^+_{\mathcal{E}}\rangle\rangle \mathrm{F}(p^f \vee \neg p^f)$ ***which is a contradiction****.*

(only if) *This direction of the proof becomes an obvious opposite running of (if).*

Example 7 *In the example of the figure 5 we can see that for each configuration $\langle q'_{\mathcal{E}}, \overline{q}\rangle$ that is not issued from a same state $q \in Q$, s.t. $\langle q_{0\mathcal{E}}, \overline{q_0}\rangle \xrightarrow{(\langle a,a\rangle(\langle b,b\rangle\langle a,a\rangle))^*)} \langle q'_{\mathcal{E}}, \overline{q}\rangle$ we have $\mu(\langle q'_{\mathcal{E}}, \overline{q}\rangle) = \Diamond p^f$. We can also read that for any post-language $\sigma \in \breve{\rho}^n$ with $\rho = (\langle a,a\rangle(\langle b,b\rangle\langle a,a\rangle)^*\langle c,c\rangle)$ and $n \geq 1$ s.t. $\langle q_{0\mathcal{E}}, \overline{q_0}\rangle \xrightarrow{\sigma} \langle q'_{\mathcal{E}}, \overline{q}\rangle$ we have $\mu(\langle q'_{\mathcal{E}}, \overline{q}\rangle) = \{p^f\}$ or $\mu(\langle q'_{\mathcal{E}}, \overline{q}\rangle) = \{\neg p^f\}$.*

4.3 Active Diagnosability of a set of distributed systems

In the previous section we considered active diagnosability of a system within its environment. The environment as defined represents a maximal use of the system. Implicitly we supposed that, each time the system can accept a command, the environment can provide it. It is interesting to extend the notion of active diagnosability to any environment composed by a set of interacting systems. In this context, when a system fails, the diagnosis is performed by its environment composed by the other partners. Let us consider a set of interacting systems $\mathcal{A} = \{A_i\}_{i=1\ldots n}$. We suppose that the k^{th} system holds a fault f and the others do not. We call context of A_k the set of the other systems, $Cont_k = \mathcal{A} \setminus \{A_k\}$. The interaction of the system A_k with the other subsystems can be seen as a game between the system and a coalition composed by its context. Let us note $G_k = G^{A^+_{k\mathcal{E}},\overline{A_k}}$ the game between the system A_k and its perfect environment. According to propositions 2 and 3 we have $G^{\{A_i\}_{i=1\ldots n}} \sim G^{Cont_k \cup \{G_k[\mathcal{N}]\}}$. Let $G = G^{Cont_k \cup \{G_k[\mathcal{N}]\}}$, we annotate its configurations by the function $\mu : \mathcal{C} \to 2^{\Delta^{A_k}}$ with $\mu(\langle q_1, \ldots, c_k, \ldots q_n\rangle) = \mu_k(c_k)$ with $c_k \in \mathcal{C}_k$. The environment resulting from $Cont_k$ will at the best behave as the perfect environment. This makes interesting the question of diagnosability in a given context, i.e. "can the context of a faulty system actively diagnose it?".

Definition 9 *Let $\mathcal{A} = \{A_i\}_{i=1\ldots n}$ be a set of interacting systems. The system A_k is diagnosable in the context of $\mathcal{A}$ iff $\forall f \in L_{fk}, G \models \mathrm{AG}[\Diamond p^f \Rightarrow \langle\langle C\rangle\rangle \mathrm{F}(p^f \vee \neg p^f)]$ with $C \subseteq Cont_k$.*

This means that a system is diagnosable in a context if there is a subset of systems, in its context, that can form a coalition in order to diagnose any of its faults.

Example 8 *Let $\mathcal{A} = \{A, A1, A2\}$ with A the system of figure 1 with $L^{uc} = \{b, c\}$ and $A1$, $A2$ the two systems represented in figure 6. In this context the system $A2$ decides about the diagnosability of A by activating the command e that produces the command c. We have here $G^{\{A1,A2\} \cup \{G^{A^+_{\mathcal{E}},\overline{A_k}}[\mathcal{N}]\}} \models \mathrm{AG}[\Diamond p^f \Rightarrow \langle\langle A2\rangle\rangle \mathrm{F}(p^f \vee \neg p^f)]$, i.e. the system $A2$ has a strategy to actively diagnose the system A. After receiving the*

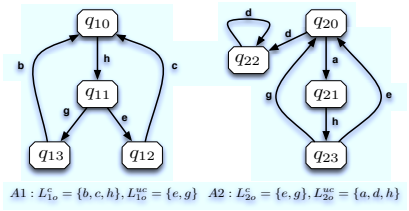

Figure 6: An active diagnosable context for A

first a, followed by h, the system $A2$ can send the command e to $A1$ to force it to send c to A. The system $A2$ will then receive the diagnosis result: a means correct, d means the fault happened in A.

5 RELATED WORK

The paper proposes a generalization of the diagnosability definition presented in (Sampath *et al.*, 1995). The aim is to take into account the semantics of the observable events (actions or reactions). The idea of gathering control theory and diagnosis is promising as claimed in (Kelly *et al.*, 2009). At the best of our knowledge (Wang, 2009), and then in (Wang *et al.*, 2009), was the first that linked diagnosis concerns and control. These two works focus on the use of controllability and observability for fault avoidance purpose. They suppose a safe sublanguage of the system and also produce, if it is possible, a controller that can ensure that the system will be safe. In this work we generalize diagnosability without avoiding fault. We do not influence the behavior of system before the diagnosis process is engaged. Our approach to check active diagnosability is related to works that use symbolic model checking technics in order to verify diagnosability (such as in (Jiang and Kumar, 2001) (Cordier and Largout, 2001) (Cimatti *et al.*, 2003)). We used to run the toy example here the MOCHA tool (Alur *et al.*, 2001). MOCHA offers a language to specify a set of systems and to compose a game structure between a set of players. The tool has also a model checker for ATL formulas. We extended our method to a set of systems, but the checking process is still performed in a centralized manner. It will be interesting to adapt the fault propagation method proposed in (Schumann and Pencolé, 2007) in order to study a purely distributed checking method (Distributed strategy computing).

6 CONCLUSION AND FUTURE WORK

In this paper we propose a generalization of diagnosability definition and verification in the context of open systems. Our active diagnosability definition can detect systems as diagnosable while the classical definition states the opposite. Also it can improve the diagnosis process by producing automatically a scenario (strategy of commands) of interaction with the system in order to get accurate diagnosis. The method can be used for a single system or for a system placed in a context. In the last case the active diagnosability and its checking method proposed here can be used for a number of applications like pervasive systems and self-healing systems. The natural extension of this work is the use of the game metaphor for a distributed checking of active diagnosability. Unfortunately, the MOCHA tool does not compute strategies neither as counter example nor as illustration: we used it only for diagnosis purpose.

REFERENCES

(Alur *et al.*, 1997) R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Journal of the ACM*, pages 100–109. IEEE Computer Society Press, 1997.

(Alur *et al.*, 2001) R. Alur, L. Alfaro, R. Grosu, T. A. Henzinger, M. Kang, C. M. Kirsch, R. Majumdar, F. Mang, and B. Y. Wang. jmocha: A model checking tool that exploits design structure. In *In Proceedings of the 23rd international conference on Software engineering*, pages 835–836, 2001.

(Cimatti *et al.*, 2003) A. Cimatti, C. Pecheur, and R. Cavada. Formal verification of diagnosability via symbolic model checking. In *In Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI03*, pages 363–369, 2003.

(Cordier and Largout, 2001) M.-O. Cordier and C. Largout. Using model-checking techniques for diagnosing discrete-event systems. In *Proceedings of the Twelve International Workshop on Principles of diagnosis (DX'01)*, pages 39–46, Mars 2001.

(Jiang and Kumar, 2001) S. Jiang and R. Kumar. Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. In *IEEE Transactions on Automatic Control*, pages 128–133, 2001.

(Kelly *et al.*, 2009) T. Kelly, Y. Wang, S. Lafortune, and M. Welsh. A formal foundation for failure avoidance and diagnosis. technical Repport HPL-2009-203, HP labs, August 2009.

(Melliti *et al.*, 2008) T Melliti, P Poizat, and S Ben Mokhtar. Distributed behavioural adaptation for the automatic composition of semantic services. *Fundamental Approaches to Software Engineering, FASE'08*, pages 146–162, 2008.

(Milner, 1989) R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

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

(Schumann and Pencolé, 2007) A. Schumann and Y. Pencolé. Scalable diagnosability checking of event-driven systems. In *IJCAI*, pages 575–580, 2007.

(Wang *et al.*, 2009) Y. Wang, S. Lafortune, T. Kelly, M. Kudlur, and S. A. Mahlke. The theory of deadlock avoidance via discrete control. In *POPL*, pages 252–263, 2009.

(Wang, 2009) Y. Wang. *Software Failure Avoidance Using Discrete Control Theory*. PhD thesis, University of Michgan, 2009.