ECAI 2008
M. Ghallab et al. (Eds.)
IOS Press, 2008

doi:10.3233/978-1-58603-891-5-189

Model-Based Diagnosis of Discrete Event Systems with an Incomplete System Model

Xiangfu Zhao and **Dantong Ouyang** [1]

Abstract. Model-based diagnosis of discrete event systems (DESs) is more and more active in artificial intelligence. However, there has been always a very restrictive assumption in the previous works that the model of a given DES is complete, including all nominal behaviors and all possible failure behaviors of the system. In order to relax this so restrictive assumption, in this paper, model-based diagnosis of a DES with an incomplete system model is investigated. A new concept of "*P-synchronization product*" of finite state automata is proposed, by which the P-diagnosis of the DES with an incomplete system model is easily put forward. It is also shown that the traditional synchronization product of finite state automata can be seen as a special situation of P-synchronization product. In addition, an ideal heuristic way from theoretical view to improve the P-synchronization product is discussed as well.

1 Introduction

Model-based diagnosis is one of the active branches of artificial intelligence. Since a formalization of model-based diagnosis with first-order logic given by R. Reiter [26], it has been widely studied. Earlier, static systems were studied by researchers (e.g. [13, 12, 33, 25], etc.), and then researches on dynamic systems have begun since the last decade (see [6, 16, 22, 31], etc.). Especially, model-based diagnosis of DESs has arisen increasing interests, as DESs cover continuous-variable systems which, after quantization, are represented as discrete systems [21] for the purpose of diagnosis at a higher level of abstraction, as well as "discrete by nature" systems.

This domain is more and more active since the seminal work of [29, 30], which has been the basis not only for subsequent contributions in the control engineering field [14], but also for further research in artificial intelligence [27]. A number of model-based approaches for diagnosing DESs have been proposed in both fields literature. And they have been widely applied, particularly in large scale telecommunication networks in [9, 24, 28] and power transmission networks in [1, 4, 17, 18, 19, 20].

Model-based diagnosis of DESs consists in finding what happened to the system from existing observations. A diagnosis is defined as the set of trajectories consistent with the observations. There have been different terminologies used as *histories* [1], *narratives* [2], *consistent paths* [5], *trajectories* [10] or *scenarios* [11]. In this paper, we mainly concern the diagnosis of DESs [3] where the system behavior is modeled by automata. Then a usual formal way of representing the diagnosis problem is to express it as the synchronized product of the system model automaton and an observation automaton.

[1] School of Computer Science and Technology, Jilin University, Changchun 130012 China, email: xiangfuzhao@gmail.com, ouyangdantong@163.com

However, there has been always an assumption in the previous works that the model of the given DES is complete, including all nominal behaviors and all possible failure behaviors. Generally speaking, the assumption is very restrictive, as it is difficult to be assured that the model is complete practically. Also inspired from the paper [7], which copes with an incomplete system model for static diagnosis, in this paper, we mainly concern model-based diagnosis of a DES with an incomplete system model similarly. A novel concept of "*P-synchronization product*" of finite state automata is proposed, by which the diagnosis of the DES with an incomplete system model is easily presented. It is also shown that the traditional synchronization product can be seen as a special situation of P-synchronization product.

This paper is organized as follows: Some preliminary knowledge about model-based diagnosis of DESs is introduced in the second section. The concepts of P-synchronization product and P-diagnosis are proposed in section three. Section four presents a heuristic way to refine P-diagnosis. Related works are compared in section five. And in the last section, we give a conclusion.

2 Preliminaries

2.1 Automata and synchronization

An *automaton* is represented as a tuple (Q, E, T, I, F) where Q is the set of states, E the set of events, T the set of transitions (q, l, q') with $l \subseteq E$, I the set of initial states, and F the set of final states. For each state $q \in Q$, generally we suppose $(q, \phi, q) \in T$.

A *trajectory* denotes a path in the automaton joining an initial state to a final state. And we use $Traj(A)$ to denote the set of trajectories of an automaton A correspondingly. Moreover, in the following, we consider *trim* automata only, where the *trim* operation transforms an automaton by removing the states that do not belong to any trajectory.

The synchronization operation on any two automata A_1 and A_2 can build the *trim* automaton, where all the trajectories of both automata which cannot be synchronized according to the synchronization events (i.e. $E_1 \cap E_2$) will be removed. Formally, suppose given two automata $A_1 = (Q_1, E_1, T_1, I_1, F_1)$ and $A_2 = (Q_2, E_2, T_2, I_2, F_2)$, the synchronization of A_1 and A_2, denoted by $A_1 \otimes A_2$, is the *trim* automaton $A = Trim(A')$ with $A' = (Q_1 \times Q_2, E_1 \cup E_2, T', I_1 \times I_2, F_1 \times F_2)$ such that: $T' = \{((q_1, q_2), l, (q_1', q_2')) \mid \exists\, l_1, l_2: (q_1, l_1, q_1') \in T_1 \wedge (q_2, l_2, q_2') \in T_2 \wedge (l_1 \cap (E_1 \cap E_2) = l_2 \cap (E_1 \cap E_2)) \wedge l = l_1 \cup l_2\}$.

2.2 Diagnosis

Thanks to the definition of the synchronization operation, the definitions used in the domain of DES diagnosis where the model of the system is represented by an automaton can be given directly. Let t_0 be the starting time and t_n be the ending time of diagnosis in the following. And more details about the synchronization of automata can be found for instance in [9, 24].

Definition 1 (Model). The model of the system, denoted by Mod, is an automaton, in which the behaviors of the system are described and the trajectories of Mod represent the evolutions of the system.

The set of initial states I^{Mod} is the set of possible states of the system at t_0. As all the states of the system may be final, we suppose as usual that $F^{Mod} = Q^{Mod}$. And the set of observable events is denoted by E_o^{Mod}, a subset of E^{Mod}, and the other part of E^{Mod} is the set of all the unobservable events E_{uo}^{Mod}.

Observations can be uncertain [17] and can be represented by an automaton, where the transition labels are observable events of E_o^{Mod} in the complete system model Mod.

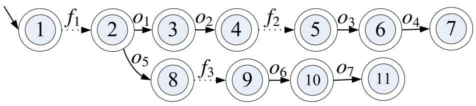
Definition 2 (Observation automaton). The observation automaton, denoted by Obs, is an automaton describing all possible observation sequences emitted by the system during the period $[t_0, t_n]$.

The diagnosis of a DES therefore can be represented as the set of *all the trajectories* of the model that are consistent with the observation sequences practically emitted by the system during the period $[t_0, t_n]$. The automaton obtained by the synchronization of the model and the observations denotes all these trajectories.

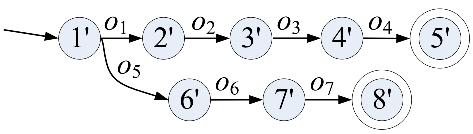
Definition 3 (Diagnosis). The diagnosis of a DES model Mod and the obtained observations Obs, denoted by Δ, is a *trim* automaton such that: $\Delta = Mod \otimes Obs$.

For a simple example, consider the system model Mod_1 and the observation automaton Obs_1 shown in Figure 1.(a) and Figure 1.(b) respectively. Then we can get the diagnostic results in Figure 1.(c) by the synchronization of Mod_1 and Obs_1.

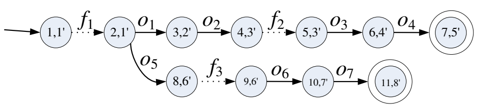
In the previous related works, generally speaking, the system model Mod is assumed to be complete, including all the nominal behaviors and all possible faulty behaviors. However, the assumption is rather restrictive, and that is only an ideal situation. It is difficult to be assured that the system model is complete practically. Therefore, some diagnostic solutions would be missed by the synchronization operation $\otimes$ when the system model is incomplete, even any diagnostic result can not be found. For instance, given the incomplete system model Mod'_1 and the observation automaton Obs_1 shown in Figure 2 and Figure 1.(b) respectively. If we still use the synchronization operation $\otimes$ to obtain the diagnoses, clearly, the result will be the null automaton. As a result, even some of the approximate diagnostic solutions to explain the observations would have been missed.



(a) The complete system model Mod_1.



(b) The real observation Obs_1.



(c) The synchronization of Mod_1 and Obs_1.

Figure 1. The Model, observation, and their synchronization.

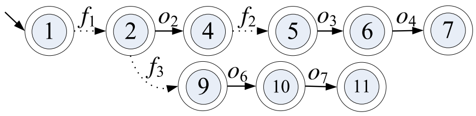


Figure 2. The incomplete system model Mod'_1.

3 P-synchronization Product and Diagnosis

3.1 Related concepts

In order to obtain the approximate diagnostic results under an incomplete model of the system and the real complete emitted observation sequences, a new concept of "*P-synchronization product*" of finite state automata is proposed in the following.

Definition 4 (P-synchronization product). Given two finite state automata A_1 and A_2, where $A_1 = (Q_1, E_1, T_1, I_1, F_1)$ and $A_2 = (Q_2, E_2, T_2, I_2, F_2)$, such that: $E_1 = \Sigma_{o1} \cup \Sigma_{uo1}$, $\Sigma_{o1} \cap \Sigma_{uo1} = \phi$, $Q_1 = F_1$, and $E_2 = \Sigma_{o2}$. The P-synchronization product of A_1 and A_2 is the automaton $A_1 \otimes_p A_2 = Trimp(A', p)$, in which " $\otimes_p$ " denotes the P-synchronization operation. And $A' = (Q_1 \times Q_2, E_1 \cup E_2, T', I_1 \times I_2, F_1 \times F_2)$ such that: $T' = \{((q_1, q_2), l, (q'_1, q'_2)) \mid \exists\, l_1, l_2$: $(q_1, l_1, q'_1) \in T_1 \wedge (q_2, l_2, q'_2) \in T_2 \wedge Constraint(l, l_1, l_2)\}$. Moreover, $Constraint(l, l_1, l_2)$ is defined as follows:

$$l = \begin{cases} l_1 & : \quad l_{1obs} = l_2 \\ l_2 & : \quad l_1 = \phi \end{cases}$$

Where l_{1obs} denotes the set: $\{\ e \mid e \in l_1 \wedge e \in \Sigma_{o1}\}$, i.e. the set of all the observable events in l_1 when the system model is considered to be as A_1 in the following. And l_{1obs} can be ϕ when $l_1 \subseteq \Sigma_{uo1}$.

In addition, the $trimp$ is a more restrictive *trim* operation, and it is based on the original $trim$ operation and a new concept of the degree of the synchronization, which will be described in the following:

Definition 5 (Synchronization degree). The synchronization degree of a trajectory $traj$ in $Trim(A')$ is denoted by $Syn\_degree(traj)$, and defined as: $Syn\_degree\ (traj) = \mid E_{syn} \mid / \mid E_{obs} \mid$, where $E_{syn} = \{l \mid l_{obs} \neq \phi \wedge l_{1obs} = l_2 \wedge l_{obs} \subseteq \Sigma_{o1} \wedge l \in traj\ \}$, $E_{obs} = \{l \mid l_{obs} \neq \phi \wedge l_{obs} \subseteq \Sigma_{o2} \wedge l \in traj\ \}$.

Note: here $l \in traj$ means that l is one of the transition labels on $traj$; l_{1obs} and l_2 are the corresponding ones in Definition 4.

With the definition of synchronization degree above, $Trimp(A', p)$ can be described as follows: $\forall$ trajectory $traj \in Trimp(A', p)$, such that: $traj \in Trim(A')$ and $Syn\_degree(traj) \geq p$. In a word, the $Trimp(A', p)$ operator deletes all the trajectories from $Trim(A')$ whose synchronization degree is less than p.

With the *P-synchronization product*, given the synchronization degree p, we can present a definition of diagnosis of a DES, when the model of the DES is incomplete.

Definition 6 (P-diagnosis). Let Mod be an automaton, which represents the incomplete system model, Obs be another automaton, which denotes all the real observation sequences. Obs is supposed to be complete. Then the P-diagnosis of the DES, denoted by Δ_p, a *trim* automaton, can be defined as follows:

$\Delta_p = Mod \otimes_p Obs$.

Note: in Mod, the corresponding Σ_{o1} is the set of all the observable transition events E_o^{Mod}, and Σ_{uo1} is the set of all the rest unobservable transition events E_{uo}^{Mod}, in which failure events are included. Whereas in Obs, the corresponding Σ_{o2} is the set of all the transition events, for each transition event is observable in Obs.

In addition, as to the synchronization operation "$\otimes$" and the P-synchronization operation "$\otimes_p$", we have the following proposition:

Proposition 1. Let A_1, A_2 be any two finite state automata, then we have $A_1 \otimes A_2 = A_1 \otimes_p A_2$ with the synchronization degree $p = 1$.

The proposition can be simply explained as follows: When $p = 1$, i.e. it is required in the *P-synchronization product* that each observation event in the automaton Obs must be synchronized with the set of all the observable events of the automaton Mod, thus the *P-synchronization product* is the same as the synchronization product.

From proposition 1 and the definitions above, it is clear that the $\otimes_p$ operator, used for the P-synchronization product, can be seen as a generalization of $\otimes$ operator, used for the synchronization product (i.e. the P-synchronization product when $p = 1$).

Moreover, when $p = 1$, the $Constraint(l, l_1, l_2)$ will become the following constraints:

$l = l_1 : l_{1obs} = l_2$.

Obviously, the value of l is more restrictive than before.

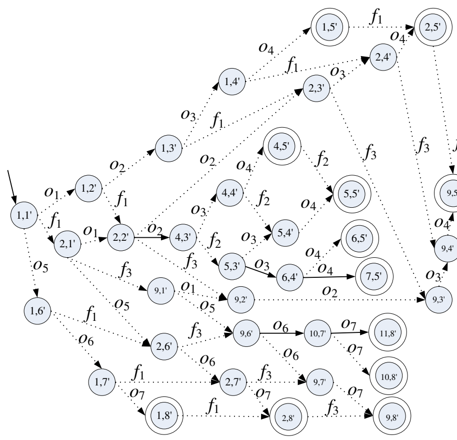
3.2 The impact of the synchronization degree p

Generally, P-diagnosis is affected by the value of synchronization degree p ($0 \leq p \leq 1$). On the one hand, if p is bigger, some diagnostic results would be missed, while the obtained results might be more reduced. On the other hand, if p is set to be smaller, more diagnostic results may be produced, while more of them would be spurious.

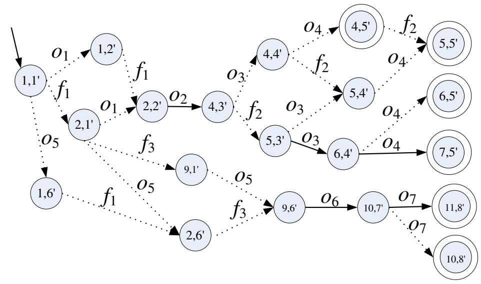
Let us give an example to show the impact on diagnostic results when the synchronization degree p is different in the following.

Example 1: Given an automaton of a DES model and an automaton of the real observation sequences shown in Figure 2 and Figure 1.(b), respectively, then we can obtain the P-diagnosis of the system shown in Figure 3 according to different values of p, where all the trajectories, each of whose synchronization degree is less than p, have been cut off, and only the events on the solid line transitions in any trajectory are the synchronized events by the observation and the incomplete system model (i.e. $l_{1obs} = l_2$).

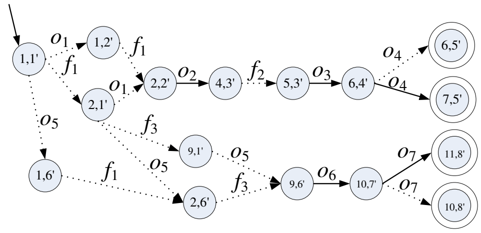
From example 1, by comparing each of the sub-figures in Figure 3 with Figure 1.(c) (the synchronization product when the model is complete), we can see clearly that when $p = 2/3$, i.e. Figure 3.(e) is the best synchronization in Figure 3. And when $p < 2/3$ in Figure 3(a),



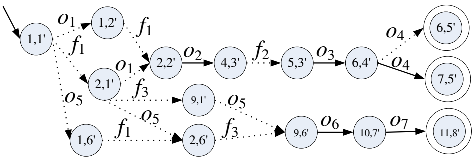
(a) $p = 0$.



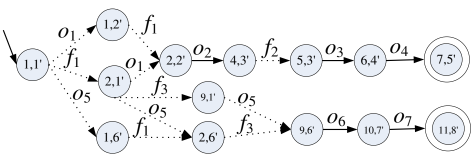
(b) $p = 1/4$.



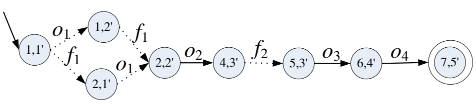
(c) $p = 1/3$.



(d) $p = 1/2$.



(e) $p = 2/3$.



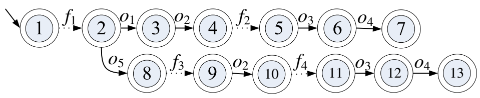
(f) $p = 3/4$.

Figure 3. The P-synchronization of incomplete system model Mod'_1 and the real observation Obs_1.

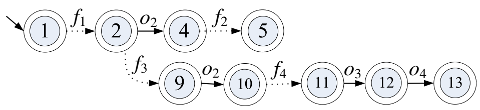
Figure 3.(b), Figure 3.(c) and Figure 3.(d), respectively, though all the possible trajectories are included in each of them, many spurious results are produced, too. While in Figure 3.(f), a practical possible trajectory is missed as a result of the high value of p.

3.3 An ideal heuristic way to refine P-diagnosis

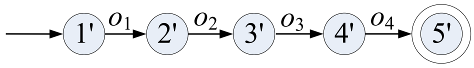
It is clearly seen that it is a difficult problem how to set the value of the synchronization degree p for better diagnosis. In fact, how to set a better value of the synchronization degree p still depends on how complete the system model is. Therefore, an ideal heuristic way from theoretical view to refine P-diagnosis is proposed as follows.



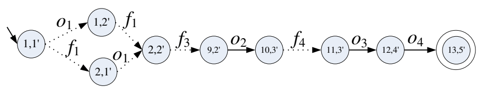
(a) A complete system model.



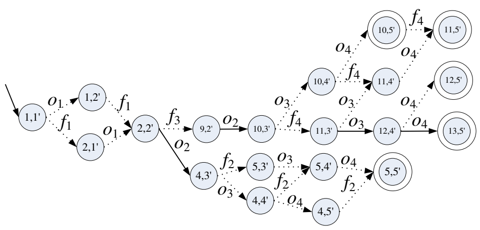
(b) An incomplete system model.



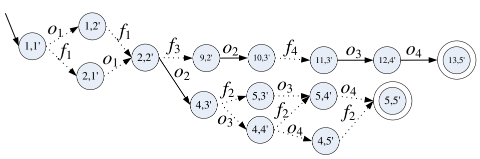
(c) An observation sequence.



(d) p = 3/4 (only the synchronization degree considered).



(e) p = 1/4 (only the synchronization degree considered).



(f) p = 1/4 (with the complete degree further considered).

Figure 4. The P-synchronization of incomplete system model and the real observation, with the synchronization degree and the complete degree considered.

Definition 7 (Completeness degree). Let the automaton M be an ideal complete system model, M' be an incomplete system model, $traj'$ be any trajectory in M', and $trajset$ be the set of corresponding complete trajectories in M. The completeness degree of the trajectory $traj'$ in M' is denoted by $compl\_degree(traj')$, and defined as: $compl\_degree(traj') = \min(\{|traj'_{obs}|/|traj_{obs}|\})$, where $traj'_{obs}$ is the set: $\{l \mid l_{obs} \neq \phi \wedge l_{obs} \subseteq E_o^M \wedge l \in traj'\}$, l_{obs} is the set: $\{e \mid e \in l \wedge e \in E_o^M\}$, and for any $traj \in trajset$, $traj_{obs}$ is the set: $\{l \mid l_{obs} \neq \phi \wedge l_{obs} \subseteq E_o^M \wedge l \in traj\}$.

Note: like before, $l \in traj$ means that l is one of the transition labels of $traj$. Here we use the minimum value of all the ratios between the incomplete trajectory and all the corresponding complete trajectories, to represent the completeness degree to further refine diagnosis, with keeping as many approximate trajectories as possible.

Suppose that we can give the completeness degree of each trajectory (such as approximately by experience, etc.) in the given incomplete system model. If the *synchronization degree* of a trajectory is not less than the *completeness degree* of the corresponding trajectory in the system model projected by the synchronized trajectory, then the synchronized trajectory can be still kept as a candidate diagnostic result. Or else, it will be removed.

Example 2: Given the complete system model shown in Figure 4.(a), and an incomplete system model and the real observation sequences are shown in Figure 4.(b) and Figure 4.(c), respectively.

If we only use the synchronization degree to obtain the approximate diagnostic results, from Figure 4.(d), we can see that some of the real possible synchronization trajectories have been missed when p = 3/4. While if p = 1/4, all the possible synchronization trajectories have been kept in Figure 4.(e). However, there are more spurious trajectories in Figure 4.(e). In order to reduce the spurious trajectories in Figure 4.(e) and at the same time keep all the real possible trajectories, the completeness degree of the trajectory is introduced, and the diagnostic results will be shown in Figure 4.(f), which is obtained as follows: We suppose here the completeness degree of the trajectory <1, $\{f_1\}$, 2, $\{o_2\}$, 4, $\{f_2\}$, 5> is 1/4, and the completeness degree of the trajectory <1, $\{f_1\}$, 2, $\{f_3\}$, 9, $\{o_2\}$, 10, $\{f_4\}$, 11, $\{o_3\}$, 12, $\{o_4\}$, 13> is 3/4. Then we add the constraint of the completeness degree to Figure 4.(e), and the more precise synchronization results are obtained shown in Figure 4.(f).

In a word, we can use the completeness degree of a trajectory as a heuristic way to refine the P-diagnosis results.

4 Related Works and Comparisons

One of the classical approaches in monitoring dynamic systems is knowledge-based techniques that directly associate a diagnosis to a set of symptoms, such as expert systems [23], or chronicle recognition systems [8, 15]. However, the main weakness of the approach is the lack of generality: once the system changes (new components, new connections, new technologies, etc.), a new expertise has to be acquired. Instead, model-based techniques used in this paper rely on a behavioral model of the system, which are known to be better suited to diagnosing DESs than expertise-based approaches.

In [32], stochastic automaton is used to represent the model of a DES, where probabilistic information is added into each transition. The main purpose of [32] is to extend the logic finite-state machines to stochastic automata to represent uncertainty of transitions. Whereas we mainly concern the incompleteness of a DES model in this paper, as usual we suppose the transitions are certain in the system model. In addition, the computation is more complex by the introduction of probability information in [32].

As to diagnosis of static system earlier (e.g. [26]), usually each diagnostic result is represented by a set of failure components. Whereas we use all the *trajectories* consistent with observation sequences to explain the evolutions of DESs. As a result of the incompleteness of the DES model, maybe there would be not much more failure information in the produced trajectories. However, suppose when all the failure behaviors are complete in the incomplete DES model, the failure information provided by the trajectories will be more precise.

5 Conclusion and Future Work

An attempt is made on diagnosis of a DES with an incomplete system model in this paper. A new concept of P-synchronization product of finite state automata is firstly proposed, which can be seen as a generalization of the traditional synchronization product, and by which the P-diagnosis of the DES is put forward as well. It is also shown that the P-diagnosis of a DES can approximately represent the diagnostic results based on the given synchronization degree p. In addition, the completeness degree of a trajectory used to improve the P-diagnosis as an ideal heuristic way is discussed as well.

In this paper, the given DES model is supposed to be global, how to process the P-diagnosis of the DES, considering the decentralized model of the system is worth doing some research in future, and the main problem may be the design of the synchronization degree of each subsystem and the corresponding sub-observations.

Once we can obtain more and more observations, the incomplete model may be complemented to be more complete by learning to discover the missed system transitions and the corresponding states. This direction can be seen as another interesting future work.

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments which helped improve this paper.

This work was supported by NSFC Grant No. 607730097; NSFC Major Research Program under Grant No. 60496321, Basic Theory and Core Techniques of Non Canonical Knowledge; Program for New Century Excellent Talents in University; Jilin Province Science and Technology Development Plan under Grant No. 20060532; and European Commission under Grant No. TH/Asia Link/010 (111084).

REFERENCES

[1] P. Baroni, G. Lamperti, P. Poglianob and M. Zanella, Diagnosis of large active systems, *Artificial Intelligence*, **110**(1) (1999), 135-183.

[2] C. Barral, S. McIlraith, and T. Son, Formulating diagnostic problem solving using an action language with narratives and sensing, in: *7th International Conference on Knowledge Representation and Reasoning - KR'00*, Breckenridge, CO, 2000, pp. 311-322.

[3] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, vol. 11 of *The Kluwer International Series in Discrete Event Dynamic Systems*, Kluwer Academic Publisher, Boston, MA, 1999.

[4] S. Cerutti, G. Lamperti, M. Scaroni, M. Zanella and D. Zanni, A diagnostic environment for automaton networks, *Software - Practice & Experience*, **37**(4) (2007), 365-415.

[5] L. Console, C. Picardi and M. Ribaudo, Diagnosis and diagnosability analysis using PEPA, in: *14th European Conference on Artificial Intelligence - ECAI'00*, Berlin, Germany, 2000, pp. 131-135.

[6] L. Console, C. Picardi, and D. Theseider Duprè, Temporal decision trees: model-based diagnosis of dynamic systems on-board, *Journal of Artificial Intelligence Research*, **19** (2003), 469-512.

[7] L. Console, D. Theseider Duprè, abd P. Torasso, A theory of diagnosis for incomplete causal models, in: *11th International Joint Conference on Artificial Intelligence - IJCAI'89*, Detroit, USA, 1989, pp. 1311-1317.

[8] M.-O. Cordier and C. Dousson, Alarm driven monitoring based on chronicles, in: *4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes - SAFEPROCESS'2000*, Budapest, Hungary, 2000, pp. 286-291.

[9] M.-O. Cordier and A. Grastien, Exploiting independence in a decentralised and incremental approach of diagnosis, in: *20th International Joint Conference on Artificial Intelligence - IJCAI'07*, Hyderabad, India, 2007, pp. 292-297.

[10] M.-O. Cordier and C. Largouët, Using model-checking techniques for diagnosing discrete-event systems, in: *12th International Workshop on Principles of Diagnosis - DX'01*, San Sicario, Italy, 2001, pp. 39-46.

[11] M.-O. Cordier and S. Thiébaux, Event-based diagnosis for evolutive systems, in: *5th International Workshop on Principles of Diagnosis - DX'94*, New Paltz, USA, 1994, pp. 64-69.

[12] P. Dague, Model-Based Diagnosis of Analog Electronic Circuits, *Annals of Mathematics and Artificial Intelligence*, **11**(1-4) (1994), 439-492.

[13] J. de Kleer and B. C. Williams, Diagnosing multiple faults, *Artificial Intelligence*, **32**(1) (1987), 97-130.

[14] R. Debouk, S. Lafortune and D. Teneketzis, Coordinated decentralized protocols for failure diagnosis of discrete-event systems, *Discrete Event Dynamical Systems*, **10**(1-2) (2000), 33-86.

[15] C. Dousson and P. L. Maigat, Chronicle recognition improvement using temporal focusing and hierarchization, in: *20th International Joint Conference on Artificial Intelligence - IJCAI'07*, Hyderabad, India, 2007, pp. 324-329.

[16] J. Gamper and W. Nejdl, Abstract temporal diagnosis in medical domains, *Artificial Intelligence in Medicine*, **10**(3) (1997), 209-234.

[17] G. Lamperti and M. Zanella, Diagnosis of discrete-event systems from uncertain temporal observations, *Artificial Intelligence*, **137**(1-2) (2002), 91-163.

[18] G. Lamperti and M. Zanella, *Diagnosis of Active Systems: Principles and Techniques*, vol. 741 of *The Kluwer International Series in Engineering and Computer Science*, Kluwer Academic Publishers, Dordrecht, NL, 2003.

[19] G. Lamperti and M. Zanella, Flexible diagnosis of discrete-event systems by similarity-based reasoning techniques, *Artificial Intelligence*, **170**(3) 2006, 232-297.

[20] G. Lamperti, M. Zanella, and D. Zanni, Incremental processing of temporal observations in model-based reasoning, *AI Communications*, **20**(1) (2007), 27-37.

[21] J. Lunze, Discrete-event modelling and diagnosis of quantized dynamical systems, in: *10th International Workshop on Principles of Diagnosis - DX'99*, Loch Awe, United Kingdom, 1999, pp. 147-154.

[22] P. J. Mosterman, *Hybrid Dynamic Systems: a Hybrid Bond Graph Modeling Paradigm and Its Application in Diagnosis*, Ph.D. dissertation, Vanderbilt University, 1997.

[23] D. Niebur, Expert Systems for Power System Control in Western Europe, in: *5th IEEE Symposium on Intelligent Control*, Philadelphia, USA, 1990, pp. 1112-1119.

[24] Y. Pencolé and M.-O. Cordier, A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks, *Artificial Intelligence*, **164**(1-2) (2005), 171-208.

[25] L. Portinale, D. Magro, and P. Torasso, Multimodal diagnosis combining case-based and model-based reasoning: a formal and experimental analysis, *Artificial Intelligence*, **158**(2) (2004), 109-153.

[26] R. Reiter, A theory of diagnosis from first principles, *Artificial Intelligence*, **32**(1) (1987), 57-96.

[27] L. Rozé, Supervision of telecommunication network: a *diagnoser* approach, in: *8th International Workshop on Principles of Diagnosis - DX'97*, Mont-Saint-Michel, France, 1997, pp. 103-111.

[28] L. Rozé and M.-O. Cordier, Diagnosing discrete-event systems: extending the "diagnoser approach" to deal with telecommunication networks, *Discrete Event Dynamic Systems*, **12**(1) (2002), 43-81.

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

[30] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, Failure diagnosis using discrete-event models, *IEEE Transactions on Control Systems Technology*, **4**(2) (1996), 105-124.

[31] P. Struss, Fundamentals of model-based diagnosis of dynamic systems, in: *15th International Joint Conference on Artificial Intelligence - IJCAI'97*, Nagoya, Japan, 1997, pp. 480-485.

[32] D. Thorsley and D. Teneketzis, Failure diagnosis of stochastic automata, in: *14th International Workshop on Principles of Diagnosis - DX'03*, Washington, DC, USA, 2003.

[33] F. Wotawa, On the relationship between model-based debugging and program slicing, *Artificial Intelligence*, **135**(1-2) (2002), 125-143.