INFORMATICA, 2005, Vol. 16, No. 3, 431–448

Fault Diagnosis of Distributed Discrete Event Systems Using OBDD *

Fei XUE, Da-zhong ZHENG
Department of Automation, Tsinghua University
Beijing 100084, China
e-mail: xuefei00@mails.tsinghua.edu.cn

Lu YAN
Turku Centre for Computer Science (TUCS) and Department of Computer Science
Åbo Akademi University, Turku 20520, Finland

Received: April 2004

Abstract. In this paper, we study the fault diagnosis problem for distributed discrete event systems. The model assumes that the system is composed of distributed components that are modeled in labeled Petri nets and interact with each other via sets of common resources (places). Further, a component's own access to a common resource is an observable event. Based on the *diagnoser* approach proposed by Sampath et al, a distributed fault diagnosis algorithm with communication is presented. The distributed algorithm assumes that the local diagnosis process can exchange messages upon the occurrence of observable events. We prove the distribute diagnosis algorithm is correct in the sense that it recovers the same diagnostic information as the centralized diagnosis algorithm. And then, the OBDD (Ordered Binary Decision Diagrams) is introduced to manage the state explosion problem in state estimation of the system.

Key words: fault diagnosis, discrete event system, Petri net, OBDD.

1. Introduction

Fault detection and isolation is an important task in the automatic control of large complex systems, due to its importance in terms of safety and efficiency of operation. A variety of complementary approaches have been proposed, based on the level of detail chosen for the model of the system and the kinds of faults that need to be diagnosed. In this paper, we consider the technological systems that can be modeled at some level of abstraction as discrete event dynamic systems. We follow an event-based approach proposed by Sampath in (Sampath *et al.*, 1995; Sampath *et al.*, 1994), namely *diagnoser approach*, and extend it to deal with distributed discrete event systems.

*This work was supported by NSFC of China under Grant 60074012 and by the National Fundamental Research Funds of China under Grant G1998020310.

In *diagnoser approach*, faults are modeled as unobservable events, namely, events whose execution can not directly detected by the sensors. A failure is said to have occurred in system if these special events, fault events, execute. Fault diagnostic is to detect the execution of fault events and identify their type or origin. The *diagnoser* is constructed based on the system model to infer the execution of fault events from the system model and future observations of the evolution of the system (Sampath *et al.*, 1995; Sampath *et al.*, 1994). However, for the large-scale distributed complex discrete event systems, such as communication networks and power systems (Benveniste *et al.*, 2003; Pencole *et al.*, 2002; Genc and Lafortune, 2003), the diagnosis is often made more complex by the need to construct a global model of system behavior and estimate global state of system on-line.

The discrete event systems considered in this paper are assumed to be composed of distributed components that are modeled in labeled Petri nets. One aspect of these systems is the presence of common resource. The different components of systems interact with each other via sets of common resources, i.e., the tokens in common places for the Petri net model. Each component's overall behavior is expressed in terms of its own behavior and its interactions with other components (Benveniste *et al.*, 2003; Ricker and Fabre, 2000). Our purpose is to propose an effectively distributed fault diagnosis algorithm for these systems.

Previously, the diagnoser approach is extended to Petri net model and a distributed diagnosis algorithm is proposed. However, the communication protocol in (Genc and Lafortune, 2003) is complex and the length of message exchanged between different local diagnosis processes is unbounded. In this paper, we improve the algorithm proposed in (Genc and Lafortune, 2003) and present a new algorithm of distributed diagnosis with communication. The communication protocol of our algorithm is much simpler and the length of exchange message is bounded. We also prove our algorithm is correct in the sense that it recovers the same diagnostic information as the centralized diagnosis algorithm.

Furthermore, diagnosing complex discrete event systems implies finding a set of behaviors that could explain the observation of system in a very complex state space. Therefore, the diagnosis problem is strongly linked with the well-known state explosion 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 can not be easily analyzed. In order to manage the state-explosion problem, the Ordered Binary Decision Diagram (OBDD) is introduced in this paper to apply for fault diagnosis. OBDDs have the capability of representing large sets of encoded data with small date structures and enable the efficient manipulation of those sets. OBDDs provide a symbolic representation for Boolean functions in the form of directed acyclic graphs (Bryant, 1986). Bryant described a set of algorithms that implement operations on Boolean functions as graph algorithms on OBDDs. Taking advantage of the efficient symbolic manipulations, researchers have solved a wide range of problems in hardware verifications, testing, and real-time systems. It is also used to analysis the properties of bounded Petri nets (Pastor *et al.*, 2001). In this paper, we firstly present the model and dynamic behavior of system using OBDD. And then, distributed fault diagnosis algorithm

based OBDD is proposed to manage the state-explosion problem for state estimation in diagnosis process.

2. System Model and Centralized Diagnosis Algorithm

In this section, we define the system model for distributed discrete event systems and briefly present the centralized diagnosis algorithm by extending the *diagnoser* approach.

2.1. *System Model*

A Petri net graph is a weighted bipartite graph, defined as a 4-tuple (Murata, 1989; Cassandras and Lafortune, 1999; Genc and Lafortune, 2003)

$$N = (P, T, I, O),$$

where P and T are finite sets respectively for places and transitions; $P \cap T = \phi$ and $P \cup T \neq \phi$; I: $P \times T \rightarrow Z$ and O: $T \times P \rightarrow Z$ are the input and output maps respectively; Z is the set of non-negative integers.

The state of Petri net graph is a mapping x: $P \rightarrow Z$. A state is represented by $x = [x(p_1), x(p_2), \ldots, x(p_n)]$, where $p_1, p_2, \ldots, p_n$ is an arbitrary fixed enumeration of P and n is the number of elements in P. A Petri net is a pair (N, x_0), where N is Petri net graph and x_0 is the initial state. The state space of (N, x_0) is given by $X = Z^n$ and $x_0 \in X$. We denote the state transition function as f: $X \times T \rightarrow X$. The state transition function is defined for state x and transition $t \in T$ if $(\forall p \in P)[x(p) \geqslant I(p,t)]$, where $I(p,t)$ is the input map from p to t. If $f(x,t)$ is defined, then we set $x' = f(x,t)$, where

$$x'(p) = x(p) - I(p,t) + O(t,p), \quad \text{for all } p \in P.$$

Here, $O(t,p)$ is the output map from t to p. Extend the state transition function f from domain $X \times T$ to domain $X \times T^*$:

$$f(x, \varepsilon) := x,$$
$$f(x, st) := f(f(x,s),t) \quad \text{for } s \in T^* \text{ and } t \in T,$$

where ε is to be interpreted as the absence of transition firing and T^* denotes the Kleene-closure of T. The set of reachable state for Petri net (N, x_0) is denoted by $R(N, x_0)$.

A labeled Petri net is defined as

$$G = (N, \Sigma, l, x_0),$$

where $N = (P, T, I, O)$ is the Petri net graph and x_0 is the initial state. Σ is the finite set (alphabet) of events and $\lambda \in \Sigma$ is the empty word. l: $T \rightarrow \Sigma$ is the labeling function that

assigns an event to each transition and can be extended to mapping $T^* \to \Sigma^*$ in the usual way, where Σ^* is the Kleene-closure of Σ.

The discrete event systems considered in this paper are modeled in labeled net $G = (N, \Sigma, l, x_0)$, where $N = (P, T, I, O)$ is the Petri net graph of the system and x_0 is the initial state of the system. We assume the discrete event system to be composed of k distributed components that interact with each other via sets of common resources. Each component $i = 1, 2, \ldots, k$ is formally modeled as a labeled Petri net $G_i = (N_i, \Sigma_i, l_i, x_{0i})$, where $N_i = (P_i, T_i, I_i, O_i)$ is the Petri net graph for component i and $x_{0i} = x_0(P_i)$ is the initial state. It is assumed that the following conditions are satisfied:

1. $T = \bigcup\limits_i T_i,\ \ \Sigma = \bigcup\limits_i \Sigma_i$ and for any $i \neq j,\ T_i \cap T_j = \phi$ and $\Sigma_i \cap \Sigma_j = \{\lambda\}$;
2. $P_i = \cup_{t \in T_i}({}^*t \cup t^*)$ for any component $i = 1, 2, \ldots, k$;

where ${}^*t := \{p \in P\colon\ I(p,t) > 0\}$ and $t^* := \{p \in P\colon\ O(t,p) > 0\}$ are the predecessors set and successor set of transition t respectively. I_i, O_i and l_i are the restrictions of I, O and l to $P_i \times T_i$, $T_i \times P_i$ and T_i, respectively. x_{0i} is the initial state of component i and it is hold that for any $p \in P_i \cap P_j$, $x_{0i}(p) = x_{0j}(p)$, $\forall i, j = 1, 2, \ldots, k$.

The corresponding Petri net graph N_i and N_j have disjoint sets of transitions. However, the sets of places are not disjoint, i.e., there are common resources (places) for different components of the system. For the common places, each component has the same initial state. Furthermore, for each component of the system, the transition set is partitioned as $T_i = T_{oi} \cup T_{uoi}$ and $T_{oi} \cap T_{uoi} = \phi$, where T_{oi} and T_{uoi} denote the set of observable transitions and the set of unobservable transitions of component i respectively. Let $T_{fi} \subseteq T_i$ be the set of failure transitions of component i, which should be diagnosed. Without loss of generality, we assume that $T_{fi} \subseteq T_{uoi}$. A transition t is labeled with λ if and only if $t \in T_{uoi}$. Similarly to (Genc and Lafortune, 2003), we assume that for any component i of the system, it satisfies the following condition:

Assumption C. $\forall t \in T_i$, if $\exists j = 1, 2, \ldots, k$ and $i \neq j$, $({}^*t \cup t^*) \cap (P_i \cap P_j) \neq \varnothing$, then the transition is observable, i.e., $t \in T_{oi}$.

The condition says that transitions putting tokens in or removing tokens from common places are observable, i.e., any transition generating or consuming common resources is observable.

Fig. 1 illustrates a simple example of distributed discrete event system (Benveniste *et al.*, 2003). The distributed system is composed by two components, component 1 and component 2. The components are modeled in labeled Petri nets $G_i = (N_i, \Sigma_i, l_i, x_{0i})$, $i = 1, 2$, and interact with each other via the common places, place p_3 and place p_7. Here, $N_i = (P_i, T_i, I_i, O_i)$ is the Petri net model for components and $T_1 = \{t_1, t_2, t_3\}$, $T_2 = \{t_4, t_5, t_6\}$, $P_1 = \{p_1, p_2, p_3, p_7\}$, $P_2 = \{p_4, p_5, p_6, p_3, p_7\}$ are the sets for transitions and places of each component respectively. For component 1, the set of events $\Sigma_1 = \{\beta, \lambda\}$ and $l_1(t_1) = l_1(t_2) = \beta,\ \ l_1(t_3) = \lambda$, i.e., transition t_1, t_2 is observable and transition t_3 is unobservable. Similarly, $\Sigma_2 = \{\alpha, \lambda\}$ and $l_2(t_4) = l_2(t_5) = \alpha$, $l_2(t_6) = \lambda$. Transition t_3 and t_6 are the failure transitions for each component to be diagnosed.

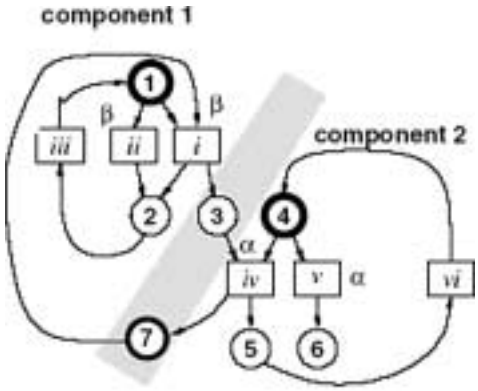

Fig. 1. Example of distributed discrete event system.

2.2. *Centralized Diagnosis Algorithm*

We briefly present the centralized diagnosis algorithm for discrete event systems modeled in labeled Petri nets. The detail for the algorithm is referred to (Genc and Lafortune, 2003).

Firstly, we define the state x_d of fault diagnosis by labeled each state of system with diagnostic information as follows

$$x_d = (x, l_f),$$

where $x = [x(p_1), x(p_2), \ldots, x(p_n)] \in X$ is the state of system modeled in labeled Petri net and fault label $l_f \in \Delta = \{0, 1\}^m$ is the diagnostic information labeled with each state of system. The fault label is a vector of length m (the number of failure transitions) which has entries of "0" or "1". When the fault label is the "zero" vector, we say that the fault label is "normal". The initial state has the "normal" fault label by definition. Let $x \in X$, $l_f \in \Delta$ and $s \in T^*$. Then, the propagation function of fault label L: $X \times \Delta \times T^* \to \Delta$ is defined as

$$L(x, l_f, s) = l_f + \left(\sum_{i=1}^{m} b_i^s \right),$$

where $b_i^s \in \Delta$ and

$$b_i^s = \begin{cases} [0, \ldots, 0, 1, 0, \ldots\ldots\ldots, 0] & \text{if } s \text{ contains } f_i \in T_f, \\ \qquad\qquad \uparrow i^{th} column & \\ [0, \ldots, 0, 0, 0 \ldots\ldots\ldots, 0] & \text{otherwise.} \end{cases}$$

Now we present the centralized diagnosis algorithm as follows (Genc and Lafortune, 2003).

Centralized Diagnosis Algorithm. Given the observable event sequence $s = \sigma_0\sigma_1\sigma_2 \ldots \sigma_n$, where $|s| = n + 1$ and $\sigma_i \in \Sigma - \{\lambda\}$,

1. Initialize the algorithm $i := 0$, $(x_0, l_{0f} = 0^m)$. Estimate the initial possible state of system as follows

$$e^0 = \big\{(x, l_f) | \exists s \in T^* \wedge l(s) = \lambda,\ x = f(x_0, s),\ l_f = L(x_0, l_{0f}, s)\big\}. \quad (1)$$

   Here s can be empty.
2. Upon observation of σ_i, do
   2.1. Compute the estimation as follows

$$\hat{\mathrm{e}}^i = \big\{x'_d = (x', l_f) \mid \exists t \in T \wedge l(t) = \sigma_i, \exists (x, l_f) \in e^i \wedge x' = f(x, t) \text{ is defined}\big\}. \quad (2)$$

   2.2. Compute the possible state of system as follows

$$e^{i+1} = \big\{x_d = (x', l'_f) | \exists s \in T^* \wedge l(s) = \lambda, \exists (x, l_f) \in \hat{e},\ x' = f(x, s),\ l'_f = L(x, l_f, s)\big\}. \quad (3)$$

   Here s can be empty. Failure transition $f_i \in T_f$ is said that has occurred if and only if $l_f(i) = 1$ for any $x_d \in e^{i+1}$.
   2.3. Increment i.

3. Distributed Diagnosis with Communication

In this section, we study the problem of distributed fault diagnosis for discrete event systems. We define the distributed diagnosis algorithm with communication between each component of the system. We assume that the message is transferred between the components correctly and without delaying and prove the distribute diagnosis algorithm is correct in the sense that it recovers the same diagnostic information as the centralized diagnosis algorithm. A simple example is given in this section to illustrate how our algorithm running. The example is also used to demonstrate the improvements over the previous algorithm in (Genc and Lafortune, 2003).

3.1. *Algorithm of Distributed Diagnosis with Communication*

For any component $G_i = (N_i, \Sigma_i, l_i, x_{0i})$ of the system, we can define the local diagnosis process using the centralized diagnosis algorithm defined in the previous section. However, if the local diagnosis processes work in isolation, the individual estimates of local process cannot provide enough information for diagnosis because the common resources (places) of any component G_i are influenced not only by its own transition, but also by the transitions of other components. Therefore, we overcome this problem by defining a communication protocol between local diagnosis processes. This protocol recovers the centralized diagnosis information by allowing the local diagnosis processed to send each other the change of tokens in the common places.

Here we consider the case when $k = 2$, i.e., the system has two component G_1 and G_2. Assuming that the two components have common places $P_c = P_1 \cap P_2$ and $|P_c| = r$, we define the state of local diagnosis process by extending the state of diagnosis, which defined in the previous section, with the message exchange with the other local diagnosis processes. For a given component $G_i = (N_i, \Sigma_i, l_i, x_{0i})$, $i = 1, 2$, the state x_d^i of local diagnosis process is defined as follows:

$$x_d^i = \big(x_i, l_f^i, (l_m^i)_p, (l_m^i)_n, l_m^j\big), \quad j \neq i$$

where x_i is the local state of the component and l_f^i is the fault label. Message label $(l_m^i)_p$, $(l_m^i)_n$ and $l_m^j \in Z^r$ represent the influence of common places between two components. The fault label and message label of the initial state are defined to be "zero" vectors. And then we define the algorithm of distributed diagnosis with communication as follows.

Distributed Diagnosis with Communication. Given the observable event sequence $s = \sigma_0\sigma_1\sigma_2 \ldots \sigma_n$, where $|s| = n + 1$ and $\sigma_i \in \Sigma - \{\lambda\}$,

1. Initialize the algorithm $i := 0$. Estimate the initial possible state of each component as follows:

$$\begin{aligned} e_1^0 = \Big\{ &\big(x_1, l_f^1, (l_{0m}^1)_p, (l_{0m}^1)_n, l_{0m}^2\big) \,|\, \exists s \in T_1^* \wedge l_1(s) = \lambda, \\ &x_1 = f(x_{01}, s),\ l_f^1 = L(x_{01}, l_{0f}^1, s)\Big\}, \end{aligned} \tag{4}$$

$$\begin{aligned} e_2^0 = \Big\{ &\big(x_2, l_f^2, (l_{0m}^2)_p, (l_{0m}^2)_n, l_{0m}^1\big) \,|\, \exists s \in T_2^* \wedge l_2(s) = \lambda, \\ &x_2 = f(x_{02}, s),\ l_f^2 = L(x_{02}, l_{0f}^2, s)\Big\}. \end{aligned} \tag{5}$$

   Here s can be empty.
2. Upon observation of σ_i, do {if $\sigma_i \in \Sigma_1$, then go to 3, else go to 4}.
3. {Master is the diagnosis process of component 1}.

   3.1. Compute the estimation of component 1 as follows

$$\begin{aligned} \hat{e}_1^i = \Big\{ &\big(x_1', l_f^1, (l_m^1)_p', (l_m^1)_n', l_m^2\big) \,|\, \exists t \in T_1 \wedge l_1(t) = \sigma_i, \\ &\exists\big(x_1, l_f^1, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in e^i \wedge x_1' = f(x_1, t) \text{ is defined.} \\ &(l_m^1)_p' = (l_m^1)_n, (l_m^1)_n' = (l_m^1)_n - I(P_c, t) + O(t, P_c)\Big\}. \end{aligned} \tag{6}$$

   3.2. Compute the possible state of component 1 as follows

$$\begin{aligned} e_1^{i+1} = \Big\{ &\big(x_1', (l_f^1)', (l_m^1)_p, (l_m^1)_n, l_m^2\big) \,|\, \exists s \in T_1^* \wedge l_1(s) = \lambda, \\ &\exists\big(x_1, l_f^1, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in \hat{e}_1,\ x_1' = f(x_1, s), \\ &(l_f^1)' = L(x_1, l_f^1, s)\Big\}. \end{aligned} \tag{7}$$

Here s can be empty. Failure transition $f_i \in T_f^1$ is said that has occurred if and only if $l_f^1(i) = 1$ for any $x_d^1 \in e_1^{i+1}$.

3.3. Send a *message* to the diagnosis process of component 2:

$$message := \Big\{ \big((l_m^1)_p, (l_m^1)_n, l_m^2\big) | \exists \big(x_1, l_f^1, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in e_1^{i+1} \Big\}. \quad (8)$$

3.4. Upon reception of this message, set of possible state of component 2 update as follows:

$$\begin{aligned} e_2^{i+1} = \Big\{ & \big(x_2', l_f^2, (l_m^2)_p, (l_m^2)_n, (l_m^1)'\big) \,|\, \exists \big((l_m^1)_p, (l_m^1)_n, l_m^2\big) \in message, \\ & \exists \big(x_2, l_f^2, (l_m^2)_p, (l_m^2)_n = l_m^2, l_m^1 = (l_m^1)_p\big) \in e_2^i,\ (l_m^1)' = (l_m^1)_n, \\ & x_2'(P_2 - P_c) = x_2(P_2 - P_c),\ x_2'(P_c) = x_2(P_c) - (l_m^1)_p + (l_m^1)_n \Big\}. \end{aligned} \quad (9)$$

3.5. Increment i.

4. {Master is the diagnosis process of component 2} Same as 3 but change 1 and 2 in every expression.

REMARK. For the previous distributed algorithm in (Genc and Lafortune, 2003), message label l_m of state $x_d^i = (x_i, l_f^i, l_m)$ is defined as the change sequence on common places, i.e., if $t \in T$ is defined at x_i, the message label propagates as follows:

$$x_i' = f(x_i, t) \quad (l_f^i)' = L(x_i, l_f^i, t) \quad \text{and } (l_m)' = \big[l_m, O(t, P_c) - I(P_c, t)\big].$$

Hence, the size of *message* label l_m will grow up unbounded along with observable events sequence. We will compare it with our algorithm detailed in Subection 3.3 using a simple example.

3.2. *Recovering the Diagnostic Information of Centralized Algorithm*

We show how the distributed diagnosis process with communication represented in the previous section can recover the state of centralized diagnosis process and prove the correctness of the algorithm of distributed diagnosis with communication by showing that it reconstructs the state of centralized diagnosis process after each observable event in the given observed sequence.

At the end of completion of the distributed diagnosis process for an observable event in sequence, let e_1^i and e_2^i be the possible states sets in the diagnosis process of component 1 and 2, respectively. We define the set $Merge(e_1^i, e_2^i)$ as follows:

$$\begin{aligned} Merge(e_1^i, e_2^i) = \Big\{ & \big(x_1(P_1)x_2(P_2 - P_c), l_f^1 l_f^2\big) \,|\, \exists \big(x_1, l_f^1, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in e_1^i, \\ & \big(x_2, l_f^2, (l_m^2)_p, (l_m^2)_n, l_m^1\big) \in e_2^i \text{ and } l_m^1 = (l_m^1)_n, l_m^2 = (l_m^2)_n \Big\}. \end{aligned}$$

Lemma 1. *For algorithm of distributed diagnosis with communication, if there exist* $(x_1, l_f^1, (l_m^1)_p, (l_m^1)_n, l_m^2) \in e_1^i$, $(x_2, l_f^2, (l_m^2)_p, (l_m^2)_n, l_m^1) \in e_2^i$ *that it is satisfied that* $l_m^1 = (l_m^1)_n$, $l_m^2 = (l_m^2)_n$, *then it is hold that* $x_1(P_c) = x_2(P_c)$.

Proof. It is easy to prove by induction using Eqs. 7 and 9. Thus, the detailed proof is omitted here.

Hence, the set $Merge(e_1^i, e_2^i)$ can equivalently be defined as follows:

$$Merge(e_1^i, e_2^i) = \Big\{\big(x_1(P_1 - P_c)x_2(P_2), l_f^1 l_f^2\big) \,|\, \exists\big(x_1, l_f^1, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in e_1^i, \big(x_2, l_f^2, (l_m^2)_p, (l_m^2)_n, l_m^1\big) \in e_2^i \quad \text{and } l_m^1 = (l_m^1)_n, l_m^2 = (l_m^2)_n\Big\}. \tag{10}$$

We prove the possible states set of centralized diagnosis process e^i can be recovered by the set $Merge(e_1^i, e_2^i)$.

Theorem 1. *Given the distributed system* G *and* G_1, G_2. *Given an observable sequence* $s = \sigma_0\sigma_1\sigma_2\ldots\sigma_n$ *and the possible states* e^i, e_1^i *and* e_2^i *are defined as above. Then,* $e^i = Merge(e_1^i, e_2^i)$.

Proof. We prove the theorem by induction.

Induction Base: $e^0 = Merge(e_1^0, e_2^0)$.

Proof (*of Induction Base*): The initial state of centralized diagnosis process is $(x_0, l_{0f} = 0^m)$. From Eq. 1, we know that $(x, l_f) \in e^0$ and

$$x = f(x_0, s) = \big\{x_0 + W(s), l_f = L(x_0, l_{0f}, s)\big\}. \tag{11}$$

$s = t_1 t_2 \ldots t_{|s|} \in T_{uo}^*$ and $f(x_0, s)$ is defined. $W(s) = \sum_{i=1}^{|s|} O(t_i, P) - I(P, t_i)$. Since addition is component-wise, Eq. 11 can be separated into two equations as

$$x(P_1) = x_0(P_1) + W_{P_1}(s), \quad x(P_2) = x_0(P_2) + W_{P_2}(s). \tag{12}$$

Based on the definition of distributed systems, $x_{01} = x_0(P_1)$ and $x_{02} = x_0(P_2)$.

Thus, from the definition of distributed systems and assumption $\boldsymbol{C}$ that unobservable transition dose not influence the common place, we have

$$\begin{aligned} &\big(x_1 = x(P_1), l_f^1 = (l_f)_{T_{f1}}, (l_{0m}^1)_p, (l_{0m}^1)_n, l_{0m}^2\big) \in e_1^0 \quad \text{and} \\ &\big(x_2 = x(P_2), l_f^2 = (l_f)_{T_{f2}}, (l_{0m}^2)_p, (l_{0m}^2)_p, l_{0m}^1)\big) \in e_2^0. \end{aligned} \tag{13}$$

Here, $(l_{0m}^1)_n = l_{0m}^1 = (l_{0m}^2)_n = l_{0m}^2 = 0^r$, $r = |P_c|$. Conversely, if $x_d^1 \in e_1^0$ and $x_d^2 \in e_2^0$, then we have that $Merge(x_d^1, x_d^2) \in e^0$.

It should be noted that $x_1(P_1)x_2(P_2 - P_c) = x_1(P_1 - P_c)x_2(P_2)$, since unobservable transition dose not influence the common places.

Induction Hypothesis: $e^i = Merge(e_1^i, e_2^i)$.
Induction Step: $e^{i+1} = Merge(e_1^{i+1}, e_2^{i+1})$.

Proof (*of Induction Step*): Without loss of generality, we assume that $\sigma_i \in \Sigma_1 - \{\lambda\}$. The proof will be done by showing inclusion in both directions for these two sets.

($\Leftarrow$) If $(x, l_f) \in e^{i+1}$, then from Eq. 2 and 3 we know that there exist $(x', l'_f) \in e^i, t \in T_1 \subseteq T, l_1(t) = l(t) = \sigma_i$ and $s = t_1 t_2 \dots t_{|s|} \in T_{uo}^*$, it holds that

$$x = f(x', ts) = x' + W(ts) \text{ is defined and } l_f = L\big(f(x', t), l'_f, s\big). \tag{14}$$

Here s can be empty. Since addition is component-wise, Eq. 14 can be separated into two equations as

$$x(P_1) = x'(P_1) + W_{P_1}(ts), \quad x(P_2) = x'(P_2) + W_{P_2}(ts). \tag{15}$$

From the induction hypothesis, we have

$$\begin{aligned} &\big(x'_1 = x'(P_1), (l_f^1)' = (l'_f)_{T_{f1}}, (l_m^1)'_p, (l_m^1)'_n, (l_m^2)'\big) \in e_1^i \text{ and} \\ &\big(x'_2 = x'(P_2), (l_f^2)' = (l'_f)_{T_{f2}}, (l_m^2)'_p, (l_m^2)'_n, (l_m^1)'\big) \in e_2^i. \end{aligned} \tag{16}$$

Here, $(l_m^1)'_n = (l_m^1)'$ and $(l_m^2)'_n = (l_m^2)' \in Z^r, r = |P_c|$. Based on Eqs. 5, 6 and assumption $\boldsymbol{C}$ that unobservable transitions dose not influence the common places, we have

$$\big(x_1, l_f^1, (l_m^1)_p = (l_m^1)'_n, (l_m^1)_n = (l_m^1)'_n - I(P_c, t) + O(t, P_c), (l_m^2)'\big) \in e_1^{i+1}. \tag{17}$$

Here, $x_1 = x(P_1)$ and $l_f^1 = (l_f)_{T_{f1}}$.

By removing all transition $t_i \in T_1$ from the unobservable sequence $s = t_1 t_2 \dots t_{|s|} \in T_{uo}^*$, we have a transition sequence $s' = t_1^2, t_2^2 \dots t_{|s'|}^2 \in T_{uo2}^*$. With assumption $\boldsymbol{C}$ that unobservable transitions do not influence the common places, $f(x'_2, s')$ is defined. From Eqs. 16 and 7, we have

$$\big(x''_2, (l_f^2)'', (l_m^2)'_p, (l_m^2)'_n, (l_m^1)'\big) \in e_2^i. \tag{18}$$

Here, $x''_2 = f(x'_2, s')$ and $(l_f^2)'' = L(x'_2, (l_f^2)', s') = (l_f)_{T_{f2}}$.

When the message is sent from the diagnosis process of component 1 to the diagnosis process of component 2, $((l_m^1)_p, (l_m^1)_n, (l_m^2)' \in message$ from Eqs. 17 and 8. Since $(l_m^1)_p = (l_m^1)'_n = (l_m^1)'$ and $(l_m^2)' = (l_m^2)'_n$, from Eq. 9, we have

$$\big(x_2, (l_f)_{T_{f2}}, (l_m^2)'_p, (l_m^2)'_n, (l_m^1)_n\big) \in e_2^{i+1}. \tag{19}$$

Here, $x_2 = x''_2 - (l_m^1)_p + (l_m^1)_n = x''_2 - I(P_c, t) + O(t, P_c) = x(P_2)$.

($\Rightarrow$) It is similar to the proof of the converse statement proved in detail above, but the steps are followed in reverse order. Thus, the details of the proof are omitted.

3.3. *A Simple Example*

Consider the distributed discrete event system given in Fig. 1. In this section, we will use it to illustrate how the algorithm in above section running. We will also compare the algorithm given in Subsection 3.2 with the previous distributed fault diagnosis algorithm in (Genc and Lafortune, 2003).

Assuming the initial state for system is

$$x_0(p_1, p_2, p_3, p_4, p_5, p_6, p_7) = (1, 0, 0, 1, 0, 0, 1).$$

Assume the sequence of observable events is $\beta\alpha\beta$. t_3, t_6 are the failure transitions to be diagnosed. Using the centralized diagnosis algorithm, the estimation set of possible states is given as following:

$$\begin{aligned}
e^0 &= \{(1,0,01,0,0,1\vdots 0,0)\},\\
e^1 &= \{(0,1,1,1,0,0,0\vdots 0,0);(1,0,1,1,0,0,0\vdots 1,0);\\
&\quad (0,1,0,1,0,0,1\vdots 0,0);(1,0,0,1,0,0,1\vdots 1,0)\},\\
e^2 &= \{(0,1,0,0,1,0,1\vdots 0,0);(1,0,0,0,1,0,1\vdots 1,0);\\
&\quad (0,1,0,1,0,0,1\vdots 0,1);(1,0,0,1,0,0,1\vdots 1,1);\\
&\quad (0,1,1,0,0,1,0\vdots 0,0);(1,0,1,0,0,1,0\vdots 1,0);\\
&\quad (0,1,0,0,0,1,1\vdots 0,0);(1,0,0,0,0,1,1\vdots 1,0)\},\\
e^3 &= \{(1,0,0,0,1,0,1\vdots 1,0);(0,1,1,0,1,0,0\vdots 1,0);\\
&\quad (1,0,1,0,1,0,0\vdots 1,0);(0,1,0,0,1,0,1\vdots 1,0);\\
&\quad (1,0,0,0,1,0,1\vdots 1,0);(1,0,0,1,0,0,1\vdots 1,1);\\
&\quad (0,1,1,1,0,0,0\vdots 1,1);(1,0,1,1,0,0,0\vdots 1,1);\\
&\quad (0,1,0,1,0,0,1\vdots 1,1);(1,0,0,1,0,0,1\vdots 1,1);\\
&\quad (1,0,1,0,0,1,0\vdots 1,0);(0,1,1,0,0,1,0\vdots 1,0);\\
&\quad (1,0,0,0,0,1,1\vdots 1,0);(0,1,0,0,0,1,1\vdots 1,0)\}.
\end{aligned}$$

Using the distributed diagnosis algorithm, the initial states for each component are

$$x_{01}(p_1, p_2, p_3, p_7) = (1, 0, 0, 1) \text{ and } x_{02}(p_4, p_5, p_6, p_3, p_7) = (1, 0, 0, 0, 1).$$

The estimation set of possible states is given as following:

$$e_1^0 = \{(1,0,0,1,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 0,0)\} \text{ and}$$

$$e_2^0 = \{(1,0,0,0,1,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 0,0)\}.$$

While observe the event β,

$$e_1^1 = \{(0,1,1,0,\vdots 0,\vdots 0,0,\vdots 1,-1,\vdots 0,0);(1,0,1,0,\vdots 1,\vdots 0,0,\vdots 1,-1,\vdots 0,0);$$
$$(0,1,0,1,\vdots 0,\vdots 0,0,\vdots 0,0\vdots 0,0);(1,0,0,1,\vdots 1,\vdots 0,0,\vdots 0,0\vdots 0,0)\}.$$

Send a *message* to the diagnosis process of component 2,

$$message(1) = \{(0,0,\vdots 0,0,\vdots 0,0);(0,0,\vdots 1,-1,\vdots 0,0)\}.$$

The estimation set of possible states of component 2 is updated as following,

$$e_2^1 = \{(1,0,0,0,1,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 0,0);(1,0,0,1,0,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 1,-1)\}.$$

Similarly, the second observed event is α, hence

$$e_2^2 = \{(0,0,1,0,1,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 0,0);(0,0,1,1,0,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 1,-1);$$
$$(0,1,0,0,1,\vdots 0,\vdots 0,0,\vdots -1,1,\vdots 1,-1);(1,0,0,0,1,\vdots 1,\vdots 0,0,\vdots -1,1,\vdots 1,-1)\},$$
$$message(2) = \{(0,0,\vdots 0,0,\vdots 0,0);(0,0,\vdots 0,0\vdots 1,-1,);(0,0,\vdots -1,1,\vdots 1,-1)\},$$
$$e_1^2 = \{(0,1,1,0,\vdots 0,\vdots 0,0,\vdots 1,-1,\vdots 0,0);(1,0,1,0,\vdots 1,\vdots 0,0,\vdots 1,-1,\vdots 0,0);$$
$$(0,1,0,1,\vdots 0,\vdots 0,0,\vdots 1,-1,\vdots -1,1);(1,0,0,1,\vdots 1,\vdots 0,0,\vdots 1,-1,\vdots -1,1);$$
$$(0,1,0,1,\vdots 0,\vdots 0,0,\vdots 0,0\vdots 0,0);(1,0,0,1,\vdots 1,\vdots 0,0,\vdots 0,0\vdots 0,0)\}.$$

The third observed event is β, hence

$$e_1^3 = \{(0,1,1,0,\vdots 1,\vdots 1,-1,\vdots 1,-1,\vdots 0,0);(1,0,1,0,\vdots 1,\vdots 1,-1,\vdots 1,-1,\vdots 0,0);$$
$$(0,1,1,0,\vdots 1,\vdots 1,-1,\vdots 2,-2,\vdots -1,1);(1,0,1,0,\vdots 1,\vdots 1,-1,\vdots 2,-2,\vdots -1,1);$$
$$(0,1,0,1,\vdots 1,\vdots 1,-1,\vdots 1,-1,\vdots -1,1);(1,0,0,1,\vdots 1,\vdots 1,-1,\vdots 1,-1,\vdots -1,1);$$
$$(0,1,1,0,\vdots 1,\vdots 0,0,\vdots 1,-1\vdots 0,0);(1,0,1,0,\vdots 1,\vdots 0,0,\vdots 1,-1\vdots 0,0);$$
$$(0,1,0,1,\vdots 1,\vdots 0,0,\vdots 0,0\vdots 0,0);(1,0,0,1,\vdots 1,\vdots 0,0,\vdots 0,0\vdots 0,0)\},$$

$$
\begin{aligned}
message(3) = \big\{&(0,0,\vdots 0,0,\vdots 0,0);(0,0,\vdots 1,-1\vdots 0,0,);(1,-1,\vdots 1,-1,\vdots 0,0);\\
&(1,-1,\vdots 1,-1,\vdots -1,1);(1,-1,\vdots 2,-2,\vdots -1,1)\big\},\\
e_2^3 = \big\{&(0,0,1,0,1,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 0,0);(0,0,1,1,0,\vdots 0,\vdots 0,0,\vdots 0,0,\vdots 1,-1);\\
&(0,1,0,0,1,\vdots 0,\vdots 0,0,\vdots -1,1,\vdots 1,-1);(1,0,0,0,1,\vdots 1,\vdots 0,0,\vdots -1,1,\vdots 1,-1)\\
&(0,1,0,1,0,\vdots 0,\vdots 0,0,\vdots -1,1,\vdots 2,-2);(1,0,0,1,0,\vdots 1,\vdots 0,0,\vdots -1,1,\vdots 2,-2)\big\}.
\end{aligned}
$$

It can be determined that the failure transition t_3 has occurred, because of $l_f^1(3) = 1$ for any $x_d^1 \in e_1^3$. Meanwhile, the occurrence of failure transition t_6 is undetermined. Comparing the above diagnosis results of centralized algorithm and distributed algorithm, it can be verified that $e^i = Merge(e_1^i, e_2^i)$, $i = 0, 1, 2, 3$, i.e., the distributed algorithm recovers the same diagnostic information as the centralized diagnosis algorithm.

For the previous distributed diagnosis algorithm proposed in (Genc and Lafortune, 2003), the *message* label for state x_d^i of each local diagnosis process is defined as the sequence of change for common places, i.e., the *messages* for above example should be

$$
\begin{aligned}
message(1) &= \big\{(0,0);(1,-1)\big\},\\
message(2) &= \big\{(0,0,\vdots 0,0);(1,-1,\vdots 0,0);(1,-1,\vdots -1,1)\big\},\\
message(3) &= \big\{(0,0,\vdots 0,0,\vdots 0,0);(1,-1,\vdots 0,0,\vdots 0,0);\\
&\qquad (1,-1,\vdots -1,1,\vdots 1,-1);(1,-1,\vdots -1,1,\vdots 0,0)\big\}.
\end{aligned}
$$

Hence, for any *message* label in $message(k)$ and state x_d^i, its size will grow up unbounded along with observable events sequence. Although it is claimed in (Genc and Lafortune, 2003) that the *message* labels can be truncated and the upper bounds on the size of *message* labels can be determined based on the structure of the Petri net, the upper bound on the size of message label is not given and the truncation process is very complex.

Different with the algorithm in (Genc and Lafortune, 2003), we defined the *message* label as Eq. 8. For local diagnosis process of component i, $(l_m^i)_p$ is the totally influence for component i on common places before the observable event and $(l_m^i)_n$ is the totally influence for component i on common places after the observable event. l_m^j is the total influence for component j and $i \neq j$. Our results show that the state estimation of local diagnosis process is independent of the sequence of change for common places, but only depend on the totally influence for each component on the common places. The size of *message* label and states x_d^i are bounded and will not grow up along with the observable events sequence. Consequently, the complex truncation process also needn't in the on-line fault diagnosis processes.

4. Distributed Fault Diagnosis Using OBDD

As shown in above, no matter the centralized or the distributed algorithm, it is required to estimating the possible state of system on-line. In large scale and complex systems, this will result bring forth the state explosion problem. In this section, we introduce the OBDD method to manage the state explosion problem in diagnosis process.

For a given weighted and bounded Petri net, the set of system states can be encoded as Boolean functions and be represented using OBDD. Then, the dynamic behaviors of system can be represented as the operation of Boolean functions using OBDD. A Petri net $(N = (P, T, I, O), x_0)$ is said to be a safe net if I: $P \times T \to \{1, 0\}$, O: $T \times P \to \{1, 0\}$, and it is holds that $x(p) \leqslant 1$ for any reachable state x of the system$\forall p \in P$. Here, we consider case that the Petri net model of the system is safe and briefly represent the symbolic analysis of Petri net using OBDD. For the bounded Petri net, it is similar to the safe case studied here. More details for of symbolic analysis of bounded Petri net using OBDD are referred to (Pastor *et al.*, 2001).

Encode State: For a given safe Petri net $(N = (P, T, I, O), x_0)$, the state space is $X = \{0, 1\}^n$. And the fault label with system state is $l_f \in \Delta = \{0, 1\}^m$. Hence, the state of the diagnosis process $x_d = (x, l_f)$ can be encoded as a Boolean function

$$F(x_d) = \prod_{i=1}^{m} p_i \prod_{j=1}^{|P|-m} \bar{p}_j \prod_{k=1}^{l} l_k \prod_{r=1}^{|T_f|-l} \bar{l}_r, \tag{20}$$

where $x(p_i) = 1$, $x(p_j) = 0$, $l_f(k) = 1$ and $l_f(r) = 0$. A set of states M can be encoded as $F_M = \bigvee_{x_d \in M} F_{x_d}$.

Encode Transition: Given a transition$t \in T$, the enable function is defined as follows

$$E_t = \prod_{p \in {}^*t} p. \tag{21}$$

And the transition function $\delta^t = (\delta^t_1, \ldots \delta^t_{|P|} \delta^t_{|P|+1}, \ldots \delta^t_{|P|+|T_f|})$ defines how the content of each place is transformed as a result of firing at marking in which it is enabled. The function is defined as follows

$$\delta^t_i(p_1 \ldots p_{|P|} l_1 \ldots l_{|T_f|}) = \begin{cases} 1 & \text{if } (p_i \in t^*), \\ 0 & \text{if } (p_i \in {}^*t) \wedge (p_i \notin t^*), \\ p_i & \text{otherwise,} \end{cases} \quad i = 1, \ldots, |P|. \tag{22}$$

And

$$\delta^t_i(p_1 \ldots p_{|P|} l_1 \ldots l_{|T_f|}) = \begin{cases} 1 & \text{if } t = f_i \in T_f, \\ l_i & \text{otherwise,} \end{cases} \quad i = |P| + 1, \ldots, |P| + |T_f|. \tag{23}$$

Dynamic Behavior of System: For a given state set M and a given transition t, the set resulted after the firing of t is

$$\delta^t(F_M \bullet E_t). \tag{24}$$

By iteratively using Eq. 24 in bounded steps, we can compute the reachable states set for the entire system, denoted as $\delta_T(F_M)$.

An ordered binary decision diagram (OBDD) (Pastor *et al.*, 2001; Bryant, 1986) is a directed acyclic graph (DAG) representation of a Boolean expression. Generally, it is exponentially more compact than its corresponding truth table representation. There are many efficient algorithms to perform all kinds of logic operations on OBDD's. It is well-known that the problem *satisfiability of Boolean expressions* is *NP-complete*, but for the OBDD of a Boolean function$f(x_1, x_2, \ldots, x_n)$, denoted $D(f)$, the time complexity of checking its *satisfiability* is $O(n)$, where n is the number of variables. So once the OBDD of a Boolean function is built, its *satisfiability* will be verified in polynomial time. In addition, the choice of variable ordering of $x_1, x_2, \ldots, x_n$ can have a significant impact on the size of its OBDD. A person with some understanding of the problem domain can generally choose an appropriate variable ordering without difficulty to build an OBDD in acceptable size (generally, in polynomial size). Furthermore, if binary encoding is applied, an arbitrary integer variable can be expressed by an OBDD vector, whose each OBDD element represents one binary bit of that integer variable. Consequently, any algebraic expression only including integers and integer variables can be represented by OBDD's.

Based on the OBDD representation of system dynamic behavior, we rewrite the Distributed Algorithm using OBDD as follows.

Distributed Diagnosis with Communication (Based on OBDD). Given that the observable event sequence $s = \sigma_0\sigma_1\sigma_2\ldots\sigma_n$ where $|s| = n+1$ and $\sigma_i \in \Sigma - \{\lambda\}$,

1. Initialize the algorithm $i := 0$. Estimate the initial possible state of each component as follows

$$e_1^0 = \left\{\left(\delta_{T_{uo1}}\left[F(x_{01}, l_{0f}^1)\right], (l_{0m}^1)_p, (l_{0m}^1)_n, l_{0m}^2\right)\right\}, \tag{25}$$

$$e_2^0 = \left\{\left(\delta_{T_{uo2}}\left[F(x_{02}, l_{0f}^2)\right], l_{0m}^1, (l_{0m}^2)_p, (l_{0m}^2)_n\right)\right\}. \tag{26}$$

2. Upon observation of σ_i, do {if $\sigma_i \in \Sigma_1$, then go to 3, else go to 4}.
3. {Master is the diagnosis process of component 1}.

   3.1. Compute the estimation of component 1 as follows:

$$\begin{aligned}\hat{e}_1^i \;=\; \cup_{t\in T_1 \wedge l_1(t)=\sigma_i} \Big\{ & \delta^t(F_M \bullet E_t), (l_m^1)_n, (l_m^1)_n - I(P_c, t) \\ & + O(t, P_c), l_m^2 \,|\, \exists\big(F_M, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in e_1^i \Big\}.\end{aligned} \tag{27}$$

   3.2. Compute the possible state of component 1 as follows:

$$e_1^{i+1} = \left\{\delta_{T_{uo1}}(F_M), (l_m^1)_p, (l_m^1)_n, l_m^2 \,|\, \exists\big(F_M, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in \hat{e}_1^i\right\}. \tag{28}$$

   3.3. Send a *message* to the diagnosis process of component 2:

$$message := \left\{\big((l_m^1)_p, (l_m^1)_n, l_m^2\big) \,|\, \exists\big(F_M, (l_m^1)_p, (l_m^1)_n, l_m^2\big) \in e_1^{i+1}\right\}. \tag{29}$$

3.4. Upon reception of this message, update the set of possible state of component 2 update as follows:

$$e_2^{i+1} = \Big\{\delta^m(F_M), (l_m^1)', (l_m^2)_p, (l_m^2)_n \,|\, \exists\big((l_m^1)_p, (l_m^1)_n, l_m^2\big) \in message, \\ \exists\big(F_M, l_m^1 = (l_m^1)_p, (l_m^2)_p, (l_m^2)_n = l_m^2\big) \in e_2^i, (l_m^1)' = (l_m^1)_n\Big\}. \quad (30)$$

Here $\delta^m = (\delta_1^m, \ldots \delta_{|P_2|}^m \delta_{|P_2|+1}^m, \ldots \delta_{|P_2|+|T_{f2}|}^m)$ is the transition according to message defined as follows

$$\delta_i^m(p_1 \ldots p_{|P_2|} l_1 \ldots l_{|T_{f2}|}) = \begin{cases} 1 & \text{if } (l_m^1)_n^{p_1} - (l_m^1)_p^{p_1} = 1, \\ 0 & (l_m^1)_p^{p_1} - f(l_m^1)_n^{p_1} = 1, \\ p_i & \text{otherwise}, \ \ i = 1, \ldots, |P_2|, \end{cases} \quad (31)$$

and

$$\delta_i^m(p_1 \ldots p_{|P_2|} l_1 \ldots l_{|T_{f2}|}) = l_i, \quad \forall i = |P| + 1, \ldots, |P| + |T_f|. \quad (32)$$

3.5. Increment i.

4. {Master is the diagnosis process of component 2}. Same as 3 but change 1 and 2 in every expression.

It should be noted that here the elements of e_1^i and e_2^i are state sets with message label. The transition according to message is defined at Eqs. 31 and 32 to modify the state corresponding to message.

5. Conclusion

In this paper, we improve the algorithm of distributed diagnosis with communication defined in (Genc and Lafortune, 2003) and apply it to the distributed systems. The new algorithm proposed here has a simpler communication protocol between diagnosis processes of different components. The size on state x_d^i and *message* exchanged between different local diagnosis processes is bounded. Moreover, the OBDD is introduced to manage the state explosion problem in state estimation of system. In this paper, we assume that the diagnosis process of different components can communicate with each other correctly and without delaying. A necessary future extension of the algorithm is to tackle with communication delay and message losing.

References

Aghasaryan, A., E. Fabre, A. Benveniste, R. Boubour, C. Jard (1997). A Petri net approach to fault detection and diagnosis in distributed systems. Part I: application to telecommunication networks, motivations, and modeling. In *Proceedings of 36th Conference on Decision&Control*.

Aghasaryan, A., A. Benveniste, R. Boubour, C. Jard (1998). Fault detection and diagnosis in distributed systems: an approach by partially stochastic Petri nets. *Discrete Event Dynamic Systems*: *Theory and Application*, **8**(2).

Baroni, P., G. Lamperti, P. Pogliano, M. Zanella (1999). Diagnosis of large active systems. *Artificial Intelligence*, **110**.

Baroni, P., G. Lamperti, P. Pogliano, M. Zanella (2000). Diagnosis of a class of distributed discrete event systems. *IEEE Trans. on Systems, Man and Cybernetics*, Part A.

Benveniste, A., E. Fabre, S. Haar, C. Jard (2003). Diagnosis of asynchronous discrete-event systems: a net unfolding approach. *IEEE Tran. on Automatic Control*, **48**(5).

Boubour, R., C. Jard, A. Aghasaryan, E. Fabre, A. Benveniste (1997). A Petri net approach to fault detection and diagnosis in distributed systems. Part II: extending Viterbi algorithm and HMM techniques to Petri nets. In *Proceedings of 36th Conference on Decision&Control*.

Bryant, R.E. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Tran. on Computers*, **C-35**(8).

Cassandras, C.G., S. Lafortune (1999). *Introduction to Discrete Event Systems*. Kluwer Academic Publishers.

Debouk, R., S. Lafortune, D. Teneketzis (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems*: *Theory and Application*, **10**(1/2).

Debouk, R., S. Lafortune, D. Teneketzis (2003). On the effect of communication delays in failure diagnosis of decentralized discrete event systems. *Discrete Event Dynamic Systems*: *Theory and Application*, **13**(3).

Jiang, S., R. Kumar, H.E. Garcia (2003). Diagnosis of repeated/intermittent failures in discrete event systems. *IEEE Tran. on Robotics and Automation*, **19**(2).

Genc, S., S. Lafortune (2003). Distributed diagnosis of discrete-event systems using Petri nets. In *Proceedings of the Applications and Theory of Petri Nets*, *Lecture Notes in Computer Science*, **2679**.

Giua, A., C. Seatzu (2002). Observability of place/transition nets. *IEEE Tran. on Automatic Control*, **47**(9).

Murata, T. (1989). Petri nets: properties, analysis and applications. In *Proceedings of IEEE*, Vol. 77(4). pp. 541–579.

Pastor, E., J. Cortadella, O. Roig (2001). Symbolic analysis of bounded Petri nets. *IEEE Tran. on Computers*, **50**(5).

Pencole, Y., M.-O. Cordier, L. Roze (2002). Incremental decentralized diagnosis approach for the supervision of a telecommunication network. In *Proceedings of the 41st IEEE Conference on Decision and Control*.

Ricker, S.L., E. Fabre (2000). On the construction of modular observers and diagnosers for discrete event systems. In *Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia*.

Roze, L., M.-O. Cordier (2002). Diagnosing discrete-event systems: extending the 'Diagnoser approach' to deal with telecommunication networks. *Discrete Event Dynamic Systems*: *Theory and Applications*, **12**(1).

Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen, D. Teneketzis (1994). Failure diagnosis using discrete-event models. *IEEE Tran. on Control System Tech.*, **4**(2), 105–123.

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

Sinnamohideen, K. (2001). Discrete-event diagnostics of heating, ventilation, and air-conditioning systems. In *Proceedings of the American Control Conference*.

Ushio, T., I. Onishi, K. Okuda (1998). Fault detection on Petri net models with faulty behaviors. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1. pp. 113–118.

F. Xue was born in Henan, China, in 1975. He received the BS degree in automatic control from Beijing Institute of Technology, Beijing, China, in 2000. He is currently pursuing the PhD degree in the Department of Automation at Tsinghua University. His research interests include discrete event dynamic systems, hybrid systems, and power systems.

L. Yan is a research fellow at the Distributed Systems Design Lab, Turku Centre for Computer Science (TUCS) and a PhD candidate at the Department of Computer Science, ÅAbo Akademi University in Finland. He received BS degree in computer science from Beijing University, China in 2000 and MS degree in computer science from ÅAbo Akademi University, Finland in 2002. He is a visiting professor of ESIGELEC (École Supérieure d'Ingénieurs généralistes) and ESC Rouen (École Supérieure de Commerce de Rouen), France in 2004. He is a member of IEEE, BCS and FME. He has been a program committee member for many international conferences and served in the editorial board in *Parallel and Distributed Computing Practices*, NOVA Science Publishers and as a book reviewer for Springer-Verlag Berlin. His current research interests are on pervasive and global computing.

D.-Z. Zheng received the diploma in automatic control from Tsinghua University, Beijing, China, in 1959. Since 1959, he has been with the Department of Automatic Control at Tsinghua University, where he is a professor in control theory and engineering. He is also a vice-chairman of Control Theory Technical Committee for Chinese Association of Automation (CAA), a deputy editor-in-chief of *Acta Automatica Sinica*, Beijing, China, and an editor of *Asian Journal of Control* (AJC). He was a visiting scholar in Department of Electrical Engineering, State University of New York at Stony Brook, from 1981 to 1983 and in 1993. His research interests include linear systems, discrete event dynamic systems, and power systems. Mr. Zheng has published many journal papers and five books.

Gedimų diagnozavimas išsklaidytoje diskrečių įvykių sistemoje naudojant OBDD

Fei XUE, Lu YAN, Da-zhong ZHENG

Straipsnyje nagrinėjama gedimų diagnozavimo išsklaidytoje diskrečių įvykių sistemoje problema. Modeliuojant daroma prielaida, kad sistema susideda iš išsklaidytų komponenčių, modeliuojamų žymėtais Petri tinklais ir tarpusavyje sąveikaujančių naudojantis bendrais resursais. Komponentės pasinaudojimas bendrais resursais yra stebimas įvykis. Straipsnyje pateikiamas išsklaidytos klaidos diagnozavimo algoritmas, naudojantis pranešimus. Išsklaidytame algoritme skaitoma, kad lokaliame diagnozavimo procese yra galimybė apsikeisti pranešimais apie tai, kad įvyko stebimas įvykis. Įrodoma, kad išsklaidytas diagnozavimo algoritmas teikia tą pačią diagnostinę informaciją, kaip ir centralizuotas diagnozavimo algoritmas. Supažindinama su OBDD (sutvarkytomis dvejetainių sprendimų diagramomis), skirtomis įvertinti sistemos būklę.