A COORDINATED DECENTRALIZED PROTOCOL FOR FAILURE DIAGNOSIS OF DISCRETE EVENT SYSTEMS

Rami Debouk, Stéphane Lafortune and Demosthenis Tenekezis
Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109-2122, USA
{ridebouk,stephane,teneke}@eecs.umich.edu; www.eecs.umich.edu/umdes

Keywords
Failure Diagnosis, Decentralized Information, Diagnostic Protocols.

Abstract

We address the problem of failure diagnosis in discrete event systems with decentralized information. We propose a coordinated decentralized architecture consisting of two local sites communicating with a coordinator that is responsible for diagnosing the failures occurring in the system. We extend the notion of diagnosability, originally introduced in [1] for centralized systems, to the proposed coordinated decentralized architecture. We specify one protocol that realizes the proposed architecture. We analyze the diagnostic properties of this protocol. The key feature of the proposed protocol is that it achieves the same diagnostic performance as the centralized diagnoser.

1 Introduction

Failure detection and isolation is an important task in the automatic control of large complex systems, and consequently, the problem of failure diagnosis has received considerable attention in the literature. Many schemes ranging from fault-tree and analytical redundancy methods to discrete event system (DES) approaches [1, 2, 3, 4, 5], model based reasoning and expert systems methods, have been proposed to approach this problem. For a brief description of these methods, the interested reader is referred to [6] and the introduction of [1] and the references therein.

Almost all of the abovementioned approaches have been developed for systems where the information used for fault diagnosis is centralized. Only the method of template monitoring [3] is cited to have the advantage of being easily implemented in distributed control architectures. Many systems are decentralized in nature, for instance, the majority of technological complex systems (computer and communication networks, manufacturing and power systems, etc.) are informationally decentralized. In this paper, we investigate failure diagnosis problems in DES under decentralized information. We extend the notion of diagnosability, introduced in [1] for centralized systems, to a coordinated decentralized architecture consisting of two local sites communicating with a coordinator that is responsible for diagnosing the failures occurring in the system. We present a specific protocol that realizes the architecture under consideration. This protocol specifies the diagnostic information generated at the local sites, the communication rules used by the local sites, and the decision rule employed by the coordinator. We present and discuss the diagnostic properties of the suggested protocol. The on-line diagnostic process is carried out by a variation of the diagnosers introduced in [1]. The key feature of the coordinated decentralized protocol presented in this paper is that it achieves the same diagnostic performance as the centralized diagnoser.

2 Preliminaries

2.1 The system model

The system to be diagnosed is modeled as a deterministic FSM

$$G = (X, \Sigma, \delta, x_0)$$  \hspace{1cm} (1)

where $X$ is the state space, $\Sigma$ is the set of events, $\delta$ is the partial transition function, and $x_0$ is the initial state of the system. The model $G$ accounts for the normal and failed behavior of the system. The behavior of the system is described by the prefix-closed language $L(G)$ generated by $G$. $L(G)$ is a subset of $\Sigma^*$, where $\Sigma^*$ denotes the Kleene closure of the set $\Sigma$. In this paper we will use the language $L(G)$, or simply $L$, and the system interchangeably.

The event set $\Sigma$ is partitioned as $\Sigma = \Sigma_o \cup \Sigma_{uo}$ where $\Sigma_o$ represents the set of observable events and $\Sigma_{uo}$ the set of unobservable events.

Let $\Sigma_f \subseteq \Sigma$ denote the set of failure events which are to be diagnosed. We assume, without loss of generality, that $\Sigma_f \subseteq \Sigma_{uo}$. Our objective is to identify
the occurrence, if any, of failure events, given that, in
the traces generated by the system, only the events
in $\Sigma_o$ are observed. In this regard, we partition the
set of failure events into disjoint sets corresponding
to different failure types

$$\Sigma_f = \Sigma_{f1} \cup \Sigma_{f2} \cup \ldots \cup \Sigma_{fj}. \quad (2)$$

Let $\Pi_f$ denote this partition. Hereafter, when we
write that a failure of type $F_i$ has occurred, we mean
that some event of the set $\Sigma_{fj}$ has occurred. We will
write $s \in \Psi(\Sigma_{fj})$ to denote the fact that the last
event of $s$ is a failure event of type $F_i$.

2.2 Diagnosability and diagnosers

A language is said to be diagnosable with respect
to a set of observable events and a failure partition
if within a finite delay, the occurrence of any failure
can be detected using the history of observable events
we refer the reader to [1] for the formal definition).

The diagnoser is a deterministic FSM built from
the system model $G$. This machine is at the core of
the diagnostic methodology of [1, 7, 8]. It is used to
analyze the diagnosability properties of $G$ and to
perform diagnostic when it observes, on-line, the behav-
or of the system. We define first the set of failure
labels $\Delta_f = \{F_1, F_2, \ldots, F_j\}$ where $|\Pi_f| = j$, and
the complete set of possible labels, $\Delta = \{N\} \cup 2^{\Delta_f}$. 
Here $N$ is to be interpreted as meaning normal, while
$F_i, i \in \{1, 2, \ldots, j\}$ as meaning that a failure of type
$F_i$ has occurred. Define $Q_o = 2^{X_o + D}$, where

$$X_o = \{x_0\} \cup \{x \in X : x\ has\ an\ observable\ 
event\ into\ it\}.$$

The centralized diagnoser for $G$ is the FSM

$$G_d = (Q_d, \Sigma_o, \delta_d, q_0) \quad (3)$$

where $Q_d, \Sigma_o, \delta_d$, and $q_0$ have the usual inter-
pretation of state space, event set, transition function
and initial state. The initial state of the diagnoser
is defined to be $\{(x_0, \{N\}\}$. The transition func-
tion $\delta_d$ of the diagnoser is constructed in a similar
manner to the transition function of an observer of
$G$, with an additional aspect that includes attaching
failure labels to the states and propagating these labels
from state to state. For more information about
the construction of the diagnoser, the reader is re-
tered to [1]. The state space $Q_d$ is the resulting
subset of $Q_o$ composed of the states of the diagnos-
er that are reachable from $q_0$ under $\delta_d$. A state $q_d$ of
$G_d$ is of the form $q_d = \{(x_1, l_1), \ldots, (x_n, l_n)\}$, where
$x_i \in X_o$ and $l_i \in \Delta$.

We say that the diagnoser $G_d$ has an $F_i$-
determinate cycle if there exist two traces $s_1$ and
$s_2$ of arbitrarily long length in $L(G)$, such that they
both have the same observable projection, and $s_1$
contains a failure event from the set $\Sigma_{fj}$, while $s_2$
does not.

3 General specification of the

3.1 A Coordinated decentralized

architecture

In this paper, we restrict attention to a coordinated
decentralized architecture with two local sites com-
muunicating with a coordinator. This architecture is
depicted in Figure 1. We will present a protocol that
realizes this architecture in Section 4.

![Figure 1: Coordinated decentralized architecture](image)

In Figure 1, the top block represents the complete
system model, or $G$ in the notation of Section 2.1.
Each site is composed of two modules: an observation
module and a diagnostic module. The site $i, i \in \{1, 2\}$, locally observes the system based on its
available sensing capabilities. Therefore, a projec-
tion $P_i$ is associated with site $i$, where $P_i$ is defined
on the set of observable events $\Sigma_{oi}$ (note here that
$\Sigma_{oi}$ and $\Sigma_{oi}$ need not be disjoint although sites 1
and 2 may be physically apart). The union of $\Sigma_{oi}$
and $\Sigma_{oi}$ is the set of observable events $\Sigma_o$. Site $i$
locally processes its own observations and generates
its diagnostic information. Both sites communicate
some form of their diagnostic information to the co-
ordinator. The task of the coordinator is to process,
according to a prescribed decision rule, the messages
received from both sites to infer occurrences of fail-
ures. If a failure is detected by the coordinator, it is
broadcasted to the failure recovery module.

We investigate the diagnosability properties of
the above architecture under the following assump-
tions.
1. $L(G)$ is live, i.e., $\forall s \in L(G), \exists \sigma \in 
\Sigma$ such that $\sigma \sigma \in L(G)$.
2. $G$ has no cycles of unobservable events with re-
spect to either $\Sigma_{oi}$ or $\Sigma_{oi}$.
3. $L(G)$ is not diagnosable with respect to $P_i$ (re-
spectively $P_2$) and $\Pi_f$ on $\Sigma_f$.
4. There is reliable communication between the local
sites and the coordinator.
5. Messages communicated between the local sites
and the coordinator are received in the order they
are sent (globally).
6. Each site knows the events that are observable by the other site.
7. The two sites are allowed to report to the coordinator only some processed version of their raw data.
8. The coordinator does not have a model of the system. It has a simple structure; specifically, it has limited memory and limited processing capabilities.

3.2 Definition of diagnosability

The definition of diagnosability in [1] assumes centralization of the available information; hence it is not directly applicable to coordinated decentralized systems. Moreover, the coordinated decentralized architecture in Figure 1 represents a class of realizations of the same architecture differentiated by the choice of the communication rules and the coordinator's decision rule. Therefore, to define diagnosability for coordinated decentralized systems, we need to account for the rules used to generate local diagnostic information together with the associated communication rules and the coordinator's decision rule. In the proposed coordinated architecture the local agents do not interact with one another; they only communicate with the coordinator that is assigned the task of detecting and isolating failures. Let $C$ denote the coordinator's diagnostic information. Based on this discussion we introduce the following definitions.

Definition 1 The coordinator's diagnostic information $C$ is said to be $F_i$-certain if based on $C$, the coordinator is certain that a failure of type $F_i$ has occurred.

Definition 2 Within the context of the coordinated decentralized architecture described in Section 3.1 and depicted in Figure 1, a protocol is defined by the diagnostic information generated at the local sites, the rules used by the local sites to communicate to the coordinator, and the decision rule used at the coordinator site.

Definition 3 A prefix-closed and live language $L$ is said to be diagnosable under a protocol, a set of projections $P_1$, $P_2$ and a failure partition $\Pi_f$ on $\Sigma_f$ if the following holds

$$(\forall i \in \Pi_f)(\exists n_i \in N)(\forall s \in \Psi(\Sigma_{f_i}))$$

$$(\forall t \in L/s)(||t|| \geq n_i \Rightarrow C = F_i - certain).$$

Thus diagnosability, requires that the detection of any failure should be achieved by the coordinator within a finite delay of the occurrence of that failure ($L/s$ denotes the post-language of $L$ after $s$ and $||t||$ denotes the length of $t$).

4 Protocol 1: a coordinated decentralized protocol

In this section, we present a protocol, called Protocol 1, for the preceding coordinated decentralized architecture that is capable of diagnosing the same types of failures as the ones diagnosed using the centralized diagnoser.

4.1 Diagnostic information at local sites

The diagnostic information at the local sites is generated by the extended diagnoser defined below. The extended diagnoser for $G$ was first introduced in [9], and it is the FSM

$$G_{e} = (Q_{e}, \Sigma_{o}, \delta_{e}, q_{0}) \tag{4}$$

where $Q_{e}$, $\Sigma_{o}$, $\delta_{e}$, and $q_{0}$ have the usual interpretation of state space, event set, transition function, and initial state. The initial state of the extended diagnoser is defined to be $\{(x_0, N), (x_0, N)\}$. A state $q \in Q_{e}$ is of the form

$$q = \{(x_1, l_1), (x_1', l_1'), \ldots, (x_n, l_n), (x_n', l_n')\}$$

where each $(x, l)$ pair is in $Q_o$, i.e., $x \in X_o$ and $l \in \Delta$. A tuple of $(x, l)$ pairs, say $\{(x_1, l_1), (x_1', l_1')\}$, has the following meaning: $x_1'$ is a component of a system state estimate after the occurrence of an observable event and $l_1'$ is its failure label, while $x_1$ is the immediate predecessor state of $x_1'$ in $G^o$ and $l_1$ is its corresponding failure label. The transition function $\delta_{e}$ of the extended diagnoser is constructed in a manner similar to the transition function of the diagnoser $G_d$, with the additional aspect that every state of $G$ that appears in a state component of $G_e$ is associated with its immediate predecessor state in $G^o$ (along the subtrace of events under consideration) and both states carry their labels; these labels are attained following the same label propagation rules as in [1]. The state space $Q_{e}$ is the resulting subset of $Q_o \times Q_o$ composed of the states of the extended diagnoser that are reachable from $q_0$ under $\delta_{e}$. By construction, $L(G_{e}) = L(G_d)$. We illustrate the construction of extended diagnosers in the following example.

Example 1 Consider the system shown in Figure 2 with $\Sigma = \{a,b,c,d,e,f\}$, $\Sigma_{uo} = \{\sigma\}$, $\Sigma_{f1} = \{\sigma\}$, $\Sigma_{o3} = \{a,c,d,e\}$, and $\Sigma_{o4} = \{b,d,e\}$. The extended diagnosers $G_{e1}$ and $G_{e2}$ for this system are shown in Figure 3. Consider the state $q = \{(2N,6N), (5N,7N)\}$ in $G_{e1}$; $q$ is read as follows: the system is either in state 6 with a normal label, or it is in state 7, also with a normal label; state 6 has been reached (by an observable event, possibly preceded by unobservable events) from state 2, while state 7 has been
reached (by an observable event, possibly preceded by unobservable events) from state 5. Now the next observable event is \( d \): if the system is at state 6, then it transitions into state 8, and since there are no failure events along the path from state 6 to state 8 the resulting component of the new state estimate is \((6N,8N)\); if the system is at state 7, it transitions into state 10 following the occurrence of the sequence \( \sigma d \), i.e., a failure of type \( F_1 \) has occurred along the path, and the resulting other component of the new state estimate is \((7N,10F1)\). Therefore the state of \( G_d^e \) is \{\( (6N,8N), (7N,10F1) \)\} after the occurrence of the observable event \( d \). All other extended diagnoser states are constructed by following a similar procedure.

![Figure 2: The system \( G \) for Example 1.](image)

<table>
<thead>
<tr>
<th>( (1N,1N) )</th>
<th>( (1N,1N) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (1N,2N),(1N,5N) )</td>
<td>( (1N,1N),(1N,4N) )</td>
</tr>
<tr>
<td>( (2N,6N),(5N,7N) )</td>
<td>( (2N,5N),(5N,8N) )</td>
</tr>
<tr>
<td>( (6N,8N),(10F1) )</td>
<td>( (6N,8N),(9F1) )</td>
</tr>
<tr>
<td>( (12N,11N),(9F1) )</td>
<td>( (12N,11N),(10F1) )</td>
</tr>
<tr>
<td>( (4N,6N) )</td>
<td>( (4N,6N) )</td>
</tr>
</tbody>
</table>

![Figure 3: The extended diagnoser \( G_{d1}^e \) (left) and \( G_{d2}^e \) for Example 1.](image)

**4.2 Communication rules**

To define the communication rules, we first note that right after the occurrence of an event that is observable only by one site, say \( i \), the state of the extended diagnoser at site \( j \neq i \) does not contain the true system state. Therefore, for the purpose of communicating information from a local site to the coordinator, we need to augment the state of the extended diagnoser with some additional information, the unobservable reach, which is defined as follows:

**Definition 4** Let \( q = \{((x_1, l_1), (x_1', l_1')), \ldots, (x_n, l_n), (x_n', l_n'))\} \) be a state of the extended diagnoser \( G_{dj}^e \), \( j \in \{1, 2\} \). Define the set

\[
S_j(q) = \{ s \in (\Sigma \setminus \Sigma_{oj})^* : s \in L_\sigma(G, x_s') \text{ for some } \sigma \in \Sigma_{oi}, i \in \{1, 2\} \} \setminus \{j\}, \text{ and some } k \in \{1, \ldots, n\}\}
\]

Then the unobservable reach of \( q \) with respect to \( \Sigma \setminus \Sigma_{oj} \) is defined as follows:

\[
UR_j(q) = \{q\} \cup \bigcup_{s \in S_j(q)} \{(y_s, l_s), (y_s', l_s')\}
\]

where (i) \( y_s' \) is the successor of some \( x_k' \), \( k \in \{1, \ldots, n\} \), after subtrace \( s \in S_j(q) \), (ii) \( y_s \) is the immediate predecessor along \( s \) of \( y_s' \) in \( G \), and (iii) \( l_s, l_s' \) are the failure labels corresponding to \( y_s, y_s' \), obtained by propagating the label \( l_k' \) of \( x_k' \) according to the label propagation function defined in [1].

The unobservable reach appends to the components of each state of the extended diagnoser \( G_{dj}^e \) some additional components (along with failure labels and predecessors) that may have been reached following an additional event or a sequence of events that are not observable by the local site \( j \). Note here that in the above definition, \( y_s \) may not be equal to \( x_k' \). Also note that while we call \( UR_j(q) \) the unobservable reach of \( q \) with respect to \( \Sigma \setminus \Sigma_{oj} \), its definition stipulates that the subtraces that are used to generate it end with an event in \( \Sigma_{oi} \), the other set of observable events.

**Example 2** Consider the state \( q = \{((1N,3N),(1N,4N))\} \) in \( G_{d2}^e \) of the preceding example. To compute the unobservable reach of \( q \) with respect to \( \Sigma \setminus \Sigma_{oi} \), we first find the set \( S_2(q) = \{a, c, ac\} \). The successors of state 3 after traces \( a \) and \( ac \) are 5 and 7, respectively, while the successor of state 4 after trace \( c \) is 6. Therefore, \( UR_2(q) = \{(1N,3N),(1N,4N),(3N,5N),(5N,7N),(4N,6N)\} \). All state labels are \( N \) since there were no failure events along any trace. Note here that although state 7 is a successor of state 3 along the trace \( ac \), the immediate predecessor of 7 in \( G \) (not pictured) is state 5, so the corresponding tuple (after adding the failure labels) is \((5N,7N)\).

We define the communication rules \( CR := \{CR_{1}, CR_{2}\} \) as follows:

\[
[CR_{1}] \ \ i = 1, 2: \text{ After the agent at site } i \text{ observes an event } \sigma \in \Sigma_{oi}, \text{ it communicates to the coordinator the corresponding state } q_i \text{ of its extended diagnoser } G_{d_{1i}}^e, \text{ its unobservable reach } UR_i(q_i) \text{ with respect to } \Sigma \setminus \Sigma_{oi}, \text{ and a status bit, } SB_i, \text{ that takes the values } SB_i = 1 \text{ when } \sigma \in \Sigma_{oi}, j \in \{1, 2\}, j \neq i, \text{ or } SB_i = 0 \text{ when } \sigma \notin \Sigma_{oj}.
\]
4.3 Decision rule

The decision rule of the coordinator consists of two components: (1) a rule according to which its information is updated; and (2) a rule according to which failure occurrences are declared and broadcasted to the failure recovery module.

As stated earlier, the coordinator declares that a failure of type $F_i$ has occurred when its diagnostic information $C$ is $F_i$-certain. To specify the information update rule we first need to define two “intersection” operations.

**Definition 5** Let $q_1 = \{(x_1,l_1),(x_1',l'_1),\ldots,(x_n,l_n),(x_n',l'_n)\}$ and $q_2 = \{(y_1,l_1),(y_1',l'_1),\ldots,(y_m,l_m),(y_m',l'_m)\}$ belong to $Q_o \times Q_o$. We denote by $\cap_i q_1 \cap_i q_2$ the intersection scheme that acts on $q_1$ and $q_2$, and we define it as follows:

$$q_1 \cap_i q_2 \triangleq \{(z,l),(z',l') \in Q_o \times Q_o ; (z',l') = (x_i',l'_i) \text{ for some } i,j, i \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,m\}, \text{ and } (z,l) = (x_i,l_i) \text{ if } i=L, \text{ otherwise } (z,l) = (y_j,l_j)\}.$$ 

This intersection scheme is a regular intersection of the components of the two system state estimates along with their failure labels. However, the intersection applies to the components corresponding to the current system state estimates and not to their immediate predecessors. The components of $q_1 \cap_i q_2$ corresponding to the immediate predecessors are determined by operator $i$.

**Example 3** Let $q_1 = \{(6N,8N),(7N,10F1)\}$ and $q_2 = \{(3N,11N),(3N,10F1),(4N,8N)\}$. To compute $q_1 \cap_i q_2$ we find the common components in the two current system state estimates, namely $8N$ and $10F1$, and we append the predecessors of $8N$ and $10F1$ in $q_1$ to the states to get $q_1 \cap_i q_2 = \{(6N,8N),(7N,10F1)\}$. Similarly, $q_1 \cap_i q_2 = \{(4N,8N),(3N,10F1)\}$.

**Definition 6** Let $q_1 = \{(x_1,l_1),(x_1',l'_1),\ldots,(x_n,l_n),(x_n',l'_n)\}$ and $q_0 = \{(y_1,l_1),(y_1',l'_1),\ldots,(y_m,l_m),(y_m',l'_m)\}$ belong to $Q_o \times Q_o$. We denote by $\cap_i q_1 \cap_i q_0 = \{(z,l),(z',l') \in Q_o \times Q_o ; (z',l') = (x_i',l'_i) \text{ for some } i,j, i \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,m\}, \text{ and } (z,l) = (x_i,l_i) \text{ if } i=L, \text{ otherwise } (z,l) = (y_j,l_j)\}$.

**Example 4** Let $q_1 = \{(6N,8N),(7N,10F1)\}$ and $q_0 = \{(4N,8N)\}$. We need to describe the structure of the coordinator before we precisely specify its information update rule. In addition to the register $C$ where the coordinator stores its current diagnostic information, eight supplementary registers are used by the coordinator for storing messages and previous relevant values necessary for the update of its information. These registers are: $R_1, R_2, R_3, R_4, C_{d,d}, SB, SB_{1,d,d}, SB_{2,d,d}$. $R_1$ and $R_2$ hold the latest states of $G_{d}'$ and $G_{d}''$, respectively, $R_3$ and $R_4$ hold the latest unobservable reaches of $G_{d}'$ and $G_{d}''$, respectively, $C_{d,d}$ holds the previous coordinator diagnostic information, $SB$ specifies whether the last event observed is by both sites (1) or not (0) and $SB_{1,d,d}, SB_{2,d,d}$ provide necessary information to compute the new coordinator diagnostic information.

The information update rule is given in Table 1. The rule picks one of the actions DR1–DR6 depending on the available information, i.e., which site observed the last and previous to the last events, and who sent the last message to the coordinator. Once the coordinator’s diagnostic information $C$ is $F_i$-certain, it broadcasts to the failure recovery module that a failure of type $F_i$ has occurred.

<table>
<thead>
<tr>
<th>Last report received from $G_{d}'$</th>
<th>$SB$</th>
<th>$SB_d$</th>
<th>$C$</th>
<th>New $SB$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DR1</strong></td>
<td>0</td>
<td>0</td>
<td>$(R_1 \cap_i R_4) \cap_i C_{d,d}$</td>
<td>0</td>
</tr>
<tr>
<td><strong>DR2</strong></td>
<td>0</td>
<td>1</td>
<td>Wait</td>
<td>1</td>
</tr>
<tr>
<td><strong>DR3</strong></td>
<td>1</td>
<td>1</td>
<td>$(R_1 \cap_i R_3) \cap_i C_{d,d}$</td>
<td>0</td>
</tr>
<tr>
<td>Last report received from $G_{d}''$</td>
<td>$SB$</td>
<td>$SB_d$</td>
<td>$C$</td>
<td>New $SB$</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td><strong>DR4</strong></td>
<td>0</td>
<td>0</td>
<td>$(R_2 \cap_i R_5) \cap_i C_{d,d}$</td>
<td>0</td>
</tr>
<tr>
<td><strong>DR5</strong></td>
<td>0</td>
<td>1</td>
<td>Wait</td>
<td>1</td>
</tr>
<tr>
<td><strong>DR6</strong></td>
<td>1</td>
<td>1</td>
<td>$(R_1 \cap_i R_5) \cap_i C_{d,d}$</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Information update rule at the coordinator site

4.4 Diagnostic properties of Protocol 1

**Theorem 1** Under Assumptions 1 - 8 in Section 3.1, we have the following: (i) the coordinator’s diagnostic information $C$ under Protocol 1 is the same as the state of the centralized extended diagnoser $G_d'$, and (ii) Protocol 1 achieves the same diagnostic performance as the centralized diagnoser does.

Note that, according to Assumption 8, the coordinator has no knowledge of the system model, and has limited memory and processing capabilities. Yet, if the coordinator has the memory and processing capabilities required by the decision rule described in
Section 4.3, it can diagnose the same types of failures as the centralized diagnoser. Consequently, building from the results in [1], the necessary and sufficient conditions for diagnosability with respect to Protocol 1 can be stated with respect to the centralized diagnoser as follows:

**Theorem 2** A live and prefix-closed language $L$ is diagnosable with respect to Protocol 1, the set of projections $P_1$, $P_2$ and the failure partition $\Pi_f$ on $\Sigma_f$ if and only if the centralized diagnoser $G_d$ does not have $F_i$-indeterminate cycles for all failure types $F_i$.

5 Discussion

We first note that the partitioning of observable events does not affect the diagnostic capabilities of Protocol 1: irrespective of the partitioning of the set of observable events $\Sigma_o$, as long as the centralized diagnoser is capable of identifying all failure types, so is Protocol 1, and vice-versa. This is a direct consequence of Theorem 1.

Based on the preceding results, one may ask the following question: is it possible to replace the extended diagnosers $G_{d1}$ and $G_{d2}$ by the diagnosers $G_{d1}$ and $G_{d2}$, respectively, while maintaining the same fundamental structure, i.e., the same functional form of the communication rules and the coordinator’s decision rule (with the obvious modifications dictated by the change of extended diagnosers to diagnosers) so that the resulting protocol achieves the same diagnostic performance as Protocol 1? The answer is, in general, no. In [10] we present another protocol, called Protocol 2, that obeys the above mentioned specifications; however, this protocol does not achieve the same diagnostic capabilities as Protocol 1. Consequently, since the objective of diagnosing all failure types that are diagnosed using the centralized diagnoser cannot be achieved using Protocol 2, one may seek conditions on the system model $G$ under which this objective can be met. Such conditions do exist as we show in [10].

The results of this paper can be extended in a straightforward manner to a coordinated decentralized architecture similar to that of Section 3 and consisting of $m$ ($m > 2$) local sites. This is explained in [10].

Finally, our analysis was based on a set of assumptions, some of which, namely the liveness of the language and the nonexistence of cycles of unobservable events, can be relaxed easily as discussed in [8]. However, the assumptions on the order of message reception at the coordinator site are indeed critical. In [10] we present an example where the violation of Assumption 5 (on preserving the global order of reception of messages at the coordinator site) leads to declaring a false positive indefinitely. The example reveals some fundamental limitations of the untimed DES mathematical model that is used.

References


