

# New Results on Decentralized Diagnosis of Discrete Event Systems

Yin Wang  
Dept. of EECS  
The University of Michigan  
1301 Beal Ave, Ann Arbor  
MI 48109-2122  
yinw@eecs.umich.edu

Tae-Sic Yoo  
Argonne  
National Laboratory  
Idaho Falls, ID 83403-2528  
tyoo@anlw.anl.gov

Stéphane Lafortune  
Dept. of EECS  
The University of Michigan  
1301 Beal Ave, Ann Arbor  
MI 48109-2122  
stephane@eecs.umich.edu

## Abstract

The past decade has witnessed the development of a body of theory, with associated applications, for fault diagnosis of dynamic systems that can be modeled in a discrete event systems framework. This paper presents several new notions of diagnosability, together with on-line diagnosis decision rules, in the context of a general decentralized architecture that allows for the use of “conditional decisions” by local diagnosers. The properties of these new notions of diagnosability are presented and their relationship with existing work discussed. Verification algorithms and local diagnoser synthesis methods are briefly outlined.

**Keywords:** Discrete event systems, fault diagnosis, decentralized diagnosis.

## 1. Introduction

Fault diagnosis in Discrete Event Systems (DES) consists of detecting unobservable fault events occurring in a system by performing model-based inferencing driven by sequences of observable events; see [2,6,7] and the references therein. Decentralized and distributed diagnostic protocols become necessary to deal with fault diagnosis in distributed systems where the information is decentralized [1,8]. In decentralized architectures, there are several local “sites” where sensors report their data and diagnosers run at each site processing the local observations and performing model-based inferencing on the basis of the projection of the system model on the locally observable events; see [1]. Local diagnosers then report their decisions about system faults; these decisions may or may not be fused at a coordinating site, according to the properties of the architecture. Generally speaking, distributed architectures for fault diagnosis differ from decentralized ones in terms of the local models used at the different sites for model-based inferencing and in terms of the ability for local diagnosers to communicate among each other in real-time.

In this paper, we are interested in decentralized architectures where diagnosers at local sites operate independently (namely, without communicating among each other) and where local decisions about (potential) system faults are merged by simple memoryless Boolean operations, in the spirit of the so-called Protocol 3 in [1]. Namely, in the first part of the paper, we consider “unconditional architectures” where there is essentially no need for a coordinating site; i.e., the decisions of the respective diagnosers will not require to be merged other than trivially. In the second part of the paper, we consider “conditional architectures” where diagnosers can issue conditional decisions about fault detection and isolation such as the decision “Fault if no other site says No Fault.” Conditional decisions have to be combined at a

coordinating site, but the fusion rule will be simple and memoryless. Our approach builds on the results in [1] regarding Protocol 3 and is inspired by recent work in [10-11] on decentralized control of DES, where conditional decisions are used to obtain more powerful control architectures and relax the condition of coobservability that arises in the necessary and sufficient conditions for supervisor existence. The use of conditional diagnosis decisions differentiates our approach from that used in [1] to improve upon Protocol 3, namely our results are different in nature from Protocols 1 and 2 in [1] which employ fusion rules based on *diagnoser state intersections* (with memory in the case of Protocol 1).

The new notions of unconditional and conditional decentralized diagnosis presented in this paper are all polynomially verifiable in the number of states of the overall system and lead to decentralized architectures requiring the synthesis of diagnosers as in [6] (or variants of these). This is to be contrasted to the approach in [8] where a notion of decentralized diagnosis is presented in the context of an architecture with communicating diagnosers. It was shown in [9] that the approach in [8] leads to undecidability of decentralized diagnosis in the case of arbitrarily long delays in the communications among diagnosers. Our polynomial decidability results are proved by following the technique presented in [12] for centralized diagnosability and recently extended in [3] for the decentralized architecture corresponding to Protocol 3 in [1].

Our development follows, in the context of diagnosis problems, the approach adopted in [10] for the so-called “general architecture” for decentralized control and its extension in [11] to conditional architectures where supervisors are allowed to issue conditional decisions such as “Enable if no other site Disables.” The work in [10-11] has led to weaker notions of coobservability as compared with the original notion of coobservability defined in [4] and studied in [5] from a computational viewpoint. The results in this paper show that unconditional and conditional architectures can be applied to the decentralized diagnosis problems and lead to relaxed notions of decentralized diagnosability. Indeed, if one associates the definition of diagnosability in the context of Protocol 3 in [1] to the notion of conjunctive coobservability of [4] in decentralized control, then the new notions of decentralized diagnosis introduced in this paper correspond the notions of disjunctive coobservability, coobservability, and conditional coobservability, respectively, from [10-11]. Moreover, in the case of diagnosis problems, the duality between the “enable” and “disable” decisions in the control architectures of [10-11] becomes one between detection of the *presence* of faults and detection of the *absence* of faults.

Due to space constraints, the presentation of our main results will be descriptive and rely upon several examples. Proofs and several technical details are omitted and can be obtained from the authors upon request.

## 2. Preliminaries

Let us consider the decentralized diagnosis architecture depicted in Fig. 1. The system is modeled as a finite state automaton  $G=(Q, \Sigma, \delta, q_0)$ , where  $Q$  is the state space,  $\Sigma$  is the set of events,  $\delta$  is the partial transition function, and  $q_0$  is the initial state. The model  $G$  accounts for the normal and faulty behavior of the system. The behavior of the system is described by the prefix-closed language  $L(G)$  generated by  $G$ , denoted by  $L$  hereafter for the sake of simplicity. The event set is partitioned as  $\Sigma=\Sigma_o\cup\Sigma_{uo}$  for observable and unobservable events, respectively. Let us first assume there is only one fault event  $f\in\Sigma_{uo}$ ; we will see later that extension to multiple fault events is straightforward. A string or trace  $s\in L$  is called *faulty* if it contains  $f$ , i.e., if there exist  $u, v\in\Sigma^*$  such that  $s=ufv$ .

As shown in Fig. 1, there are  $n$  local sites jointly diagnosing the system  $G$  by observing subsets of the set of observable events  $\Sigma_o$ , denoted by  $\Sigma_{o,1}\dots\Sigma_{o,n}$ , respectively. The blocks  $P_1, P_2,\dots P_n$  in the figure denote the projection operations from  $\Sigma^*$  to  $\Sigma_{o,i}^*$ . The decision fusion

block in Fig. 1 is assumed to be a simple memoryless Boolean function that merges the diagnosis decisions of the local sites. As mentioned in the introduction, we do *not* consider more complicated decision fusion blocks such as “coordinators” that would receive state estimates from local sites and process them in order to compute online the overall diagnosis decisions (cf. Protocols 1 and 2 of [1]). In contrast, our objective is to study the properties of decentralized architectures with the simplest possible types of fusion of local, possibly conditional, decisions.

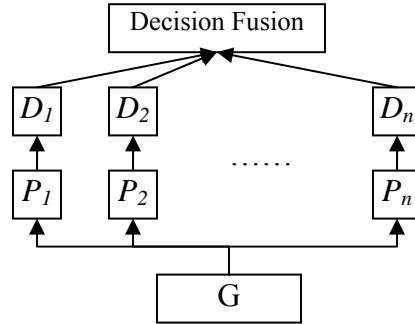


Fig. 1. Decentralized Architecture

For the sake of simplicity, we make the following standard assumptions:

**A1**  $L(G)$  is live.

**A2** Every cycle of  $G$  must contain at least one event observable at some local site.

Assumption **A1** can be relaxed easily at the expense of extra statements regarding the diagnosability of terminating traces. Assumption **A2** ensures that the system will not generate arbitrarily long sequences of unobservable events, which of course would prevent diagnosis within bounded delays.

Given that  $P_i$  is the standard projection operation from  $\Sigma^*$  to  $\Sigma_{o,i}^*$ , we have that  $P_i^{-1}(s) := \{t \in \Sigma^* : P_i(t) = s\}$ . We introduce the notation  $E_i(s) = P_i^{-1}P_i(s) \cap L$  to denote the set of estimated traces by site  $i$ , assuming  $s$  is executed by the system. Let us also introduce the notation  $ENF_i(s)$  for  $E_i(s)$  where the unobservable suffix of each element is removed. Formally,  $ENF_i(s) = E_i(s) / \Sigma_{uo,i}^*$  where  $/$  denotes the quotient operation for languages.

We present the following definition of decentralized diagnosis which will be the starting point for our development.

*Definition 1:* A prefix-closed and live language  $L$  is said to be F-codiagnosable w.r.t.  $f, P_1, P_2, \dots, P_n$ , if the following is true:

$$(\exists k \in \mathbb{N})(\forall s \in L, s \text{ is faulty})(\forall st \in L, |t| \geq k)(\exists i \in \{1, 2, \dots, n\})(\forall u \in E_i(st), u \text{ is faulty}).$$

The above definition means the following. Let  $s$  be a faulty trace and let  $t$  be a sufficient long continuation of  $s$  in  $L$ . Then there must exist at least one local site  $i$  such that any trace in  $L$  indistinguishable from  $st$  at site  $i$  is also faulty. This definition is exactly the same as the definition in [1] of “diagnosability under Protocol 3,” which is revisited in [3] under the name “co-diagnosability.” We adopt here the name “F-codiagnosability” in order to facilitate comparisons between our work and that in [10-11] for coobservability and decentralized control. It is important to note that in F-codiagnosability, the only local decision made by diagnosers is “Fault,” and the system is diagnosed to be faulty if there is at least one diagnoser reporting “Fault.” Thus, this architecture is closely analogous to the *conjunctive architecture* considered in [4, 10] for decentralized control, where “disable” is the only local decision employed and an event is disabled if at least one site disables it. In the next section, we will consider the dual problem of detecting the absence of faults in a decentralized setting and introduce the corresponding notion of NF-codiagnosability.

### 3. Architecture without Conditional Decisions

#### 3.1 Notions of Codiagnosability

Let us first look at a motivating example.

*Example 1.* Consider the system  $G$  shown in Fig. 2, where  $\Sigma=\{a, b, c, f\}$  and  $\Sigma_o=\{a, b, c\}$ . There are two local sites, i.e.,  $n=2$ .  $\Sigma_{o,1}=\{a, c\}$  and  $\Sigma_{o,2}=\{b, c\}$ .

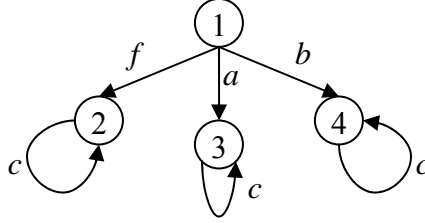


Fig. 2. The system  $G$  for Example 1

This system is not F-codiagnosable because the arbitrarily long faulty trace  $fc^n$  is indistinguishable from fault-free trace  $bc^n$  at site 1 and indistinguishable from  $ac^n$  at site 2. Recall that in the decentralized architecture corresponding to F-codiagnosability, sites are only allowed to issue “Fault” decisions. A faulty trace can therefore be diagnosed only if some site is certain about the occurrence of the fault. In this example, to diagnose faulty trace  $fc^n$ , cooperation between the two sites would be necessary.

However, we observe that the fault-free traces in Example 1 can be detected with certainty by the local sites; for instance, observation of event  $a$  at site 1 is an indication that fault event  $f$  has not occurred. Inspired by this observation, as well as by the notion of “disjunctive architectures” for decentralized supervisory control introduced in [10], we propose the related concept of NF-codiagnosability, which allows local sites to say “No Fault”. This leads to the following definition.

*Definition 2:* A prefix-closed and live language  $L$  is said to be NF-codiagnosable w.r.t.  $f, P_1, P_2, \dots, P_n$ , if the following is true:

$(\exists k \in \mathbb{N}) (\forall s \in L, s \text{ is not faulty}) (\forall st \in L, st \text{ is not faulty, } |t| \geq k) (\exists i \in \{1, 2, \dots, n\}) (\forall u \in ENF_i(st), u \text{ is not faulty})$ .

The above definition is related to the ability to detect the *absence* of a fault, i.e., if trace  $s$  is not faulty, and  $t$  is a sufficiently long fault-free extension in  $L$ , there must exist one local site  $i$  such that any trace in  $L$  indistinguishable from  $st$  at site  $i$  is also fault-free *up to the last observable event*. The reason for using  $ENF_i(st)$  in Definition 2 is to avoid the case where the fault event occurs in the unobservable suffix of the trace, after the last observable event. To motivate this choice, consider the following (centralized) example. If  $L = a^*fb^*$  where  $a$  and  $b$  are observable and  $f$  is the fault event, then as long as event  $a$  occurs, we know that  $f$  has not occurred until at least the moment when the last  $a$  is observed. So such an  $L$  should be considered to be (centralized) NF-diagnosable. We note that the notion of NF-codiagnosability was independently proposed in [3] where it is termed “strong codiagnosability” and with the difference that the estimate function  $E$  is used instead of  $ENF$ . We also note that according to Definition 2,  $K = a^*fab^*$  is not (centralized) NF-diagnosable, since the fault event  $f$  could have occurred just before the last observed  $a$ . However  $K$  is (centralized) F-diagnosable. It might be worthwhile to consider variants of Definition 2 where the centralized versions of F-codiagnosability and NF-codiagnosability coincide; this is left open for future work.

It is not difficult to verify that the system in Example 1 above is NF-codiagnosable. The fault-free traces are  $ac^*$  or  $bc^*$ , and each one will be unambiguously detected by site 1 and site 2, respectively.

Consider next the situation where instead of a single fault event  $f$ , there is a set of fault events denoted by  $\Sigma_f \subseteq \Sigma_{uo}$ . Assume there are  $m$  fault events,  $\Sigma_f = \{f_1, f_2 \dots f_m\}$ . A trace  $s$  is called  $f_i$ -faulty, if it contains fault event  $f_i$ .

*Definition 3:* A prefix-closed and live language  $L$  is said to be F-codiagnosable w.r.t.  $f_1, f_2, \dots, f_m, P_1, P_2, \dots, P_n$ , if the following is true:

$(\forall j \in \{1, \dots, m\})(\exists k_j \in N)(\forall s \in L, s \text{ is } f_j\text{-faulty})(\forall st \in L, |t| \geq k_j)(\exists i \in \{1, 2, \dots, n\})(\forall u \in E_i(st), u \text{ is } f_j\text{-faulty})$ .

*Definition 4:* A prefix-closed and live language  $L$  is said to be NF-codiagnosable w.r.t.  $f_1, f_2, \dots, f_m, P_1, P_2, \dots, P_n$ , if the following is true:

$(\forall j \in \{1, \dots, m\})(\exists k_j \in N)(\forall s \in L, s \text{ is not } f_j\text{-faulty})(\forall st \in L, st \text{ is not } f_j\text{-faulty}, |t| \geq k_j)(\exists i \in \{1, 2, \dots, n\})(\forall u \in ENF_i(st), u \text{ is not } f_j\text{-faulty})$ .

If every fault event in  $\Sigma_f$  is F[NF]-codiagnosable, then we say the system is F[NF]-codiagnosable. However, it is possible that some fault events will be F-codiagnosable while others will be NF-codiagnosable. To account for this situation, we introduce the notion of *codiagnosability*. Inspired by the notion of coobservability in the context of the ‘‘general architecture’’ in [10], we partition  $\Sigma_f$  as  $\Sigma_f = \Sigma_{f,F} \cup \Sigma_{f,NF}$ , where  $\Sigma_{f,F}$  is the set of fault events whose occurrence can be diagnosed and  $\Sigma_{f,NF}$  is the set of fault events whose absence can be diagnosed.

*Definition 5:* A prefix-closed and live language  $L$  is said to be codiagnosable w.r.t.  $\Sigma_{f,F}, \Sigma_{f,NF}, P_1, P_2, \dots, P_n$ , if

1.  $L$  is F-codiagnosable w.r.t.  $\Sigma_{f,F}, P_1, P_2, \dots, P_n$ ;
2.  $L$  is NF-codiagnosable w.r.t.  $\Sigma_{f,NF}, P_1, P_2, \dots, P_n$ .

### 3.2 Properties of Codiagnosability

In this section, we present, via examples, several results that show the relationship of various notions of diagnosability

*Proposition 1:* F-codiagnosability and NF-codiagnosability are incomparable w.r.t. the same fault event and projections  $P_1, P_2, \dots, P_n$ .

One part of Proposition 1 is proved by Example 1 above; the other part is proved by Example 2 below.

*Example 2:* (F-codiagnosable but not NF-codiagnosable) Consider (nondeterministic) system  $G$  shown in Fig. 3, with  $\Sigma = \{a, b, c, f\}$ ,  $\Sigma_o = \{a, b, c\}$ , and  $\Sigma_{uo} = \Sigma_f = \{f\}$ . There are two local sites, i.e.,  $n=2$ .  $\Sigma_{o,1} = \{a, c\}$  and  $\Sigma_{o,2} = \{b, c\}$ .

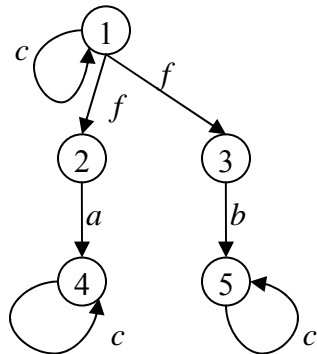


Fig. 3. F-codiag but not NF-codiag

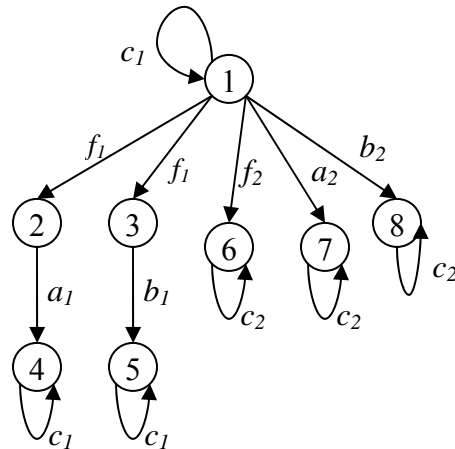


Fig. 4. The system  $G$  for Example 3

*Proposition 2:* F-codiagnosability or NF-codiagnosability implies codiagnosability, w.r.t. the same set of fault events and projections. The reverse implication is not true in general.

*Example 3:* Consider system  $G$  shown in Fig. 4 where  $\Sigma_0 = \{a_1, a_2, b_1, b_2, c_1, c_2\}$ ,  $\Sigma_{uo} = \Sigma_f = \{f_1, f_2\}$ , and where  $\Sigma_f$  is partitioned into two fault events,  $\Sigma_{f1} = \{f_1\}$ ,  $\Sigma_{f2} = \{f_2\}$ . There are two local sites, i.e.,  $n=2$ .  $\Sigma_{o,1} = \{a_1, a_2, c_1, c_2\}$  and  $\Sigma_{o,2} = \{b_1, b_2, c_1, c_2\}$ . With the preceding partition, we get codiagnosability, with  $f_1$  F-codiagnosable and  $f_2$  NF-codiagnosable.

*Proposition 3:* Codiagnosability w.r.t.  $\Sigma_{f,F}$ ,  $\Sigma_{f,NF}$ ,  $\Sigma_{o,1}$ ,  $\Sigma_{o,2}, \dots, \Sigma_{o,n}$  implies centralized diagnosability w.r.t. every fault event in  $\Sigma_{f,F}$  and  $\Sigma_{f,NF}$  and projection  $\Sigma_o = \Sigma_{o,1} \cup \Sigma_{o,2} \cup \dots \cup \Sigma_{o,n}$ . The reverse implication is not true in general.

The reverse direction is proved by the example in Fig. 5, where  $\Sigma_{o,1} = \{a, c\}$  and  $\Sigma_{o,2} = \{b, c\}$ . Here, fault  $f$  is not F-codiagnosable nor NF-codiagnosable, hence the system is not codiagnosable.

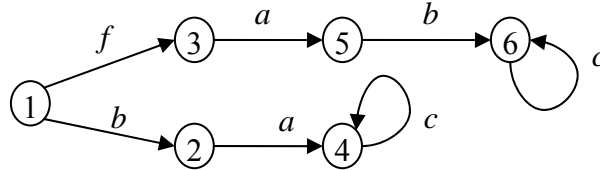


Fig. 5. A diagnosable system that is not codiagnosable

Figure 6 below summarizes the relationship among the various notions of codiagnosability discussed in this section (abbreviations have been used). Note that the set  $F\text{-codiag} \cap NF\text{-codiag}$  represents the class of systems for which the presence of faults *and* the absence of faults are both codiagnosable.

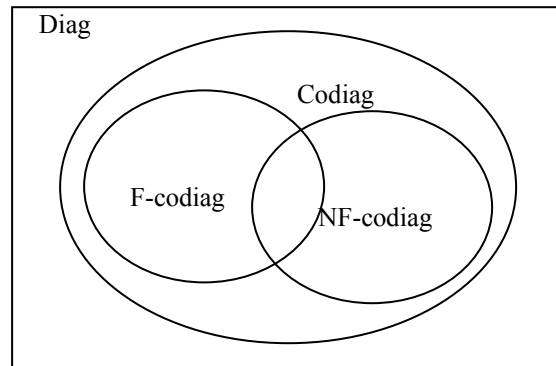


Fig. 6. Relationship among notions of diagnosability

### 3.3 Discussion

Necessary and sufficient conditions for F-codiagnosability as well as tests for these conditions are presented in [1] – recall that F-codiagnosability is the same as diagnosability w.r.t. Protocol 3 in [1]. Clearly, the same testing procedure can be extended to NF-codiagnosability and codiagnosability, as defined in the preceding section. However, such tests involve building diagnosers, which in the worst case have exponential complexity in the size of the state space of the system. It is well known that polynomial tests for the various notions of coobservability exist, based on the construction of a nondeterministic automaton that tracks relevant traces that have the same projection and can identify violations of coobservability. The same strategy was exploited in [12] where so-called “verifiers” are defined to test (centralized) diagnosability in polynomial time in the size of the state space of the system. This approach was recently extended to F-codiagnosability in [3] resulting in a polynomial test for F-codiagnosability. We can build on the approach in [12] and [3] and construct special auto-

mata for testing, in polynomial time, the property of NF-codiagnosability. Therefore, codiagnosability is verifiable in polynomial time as well. These results are not presented here.

Once the diagnosability properties of a system have been determined off-line for the decentralized architecture without conditional decisions, the on-line implementation of diagnosis functions is straightforward. It can be shown that it suffices to build diagnosers at each site (for the set of events observable at that site). These diagnosers are as in [6,1]. Faults that are F-codiagnosable will be detected when one of the diagnosers will enter a state that is F-certain for those, while the absence of faults that are NF-codiagnosable will be guaranteed as long as one of the diagnosers will be in a state that is NF-certain for those.

*Example 4:* The two diagnosers at sites 1 and 2, respectively, for Example 3 are shown in Fig. 7. The diagnosis decisions are indicated in parentheses inside the diagnoser state, where  $F_1$  and  $F_2$  refer to fault events  $f_1$  and  $f_2$ , respectively, and  $NF_1$  and  $NF_2$  means no fault  $f_1$  and no fault  $f_2$ , respectively. Recall that in this example,  $f_1$  is F-codiagnosable and  $f_2$  is NF-codiagnosable.

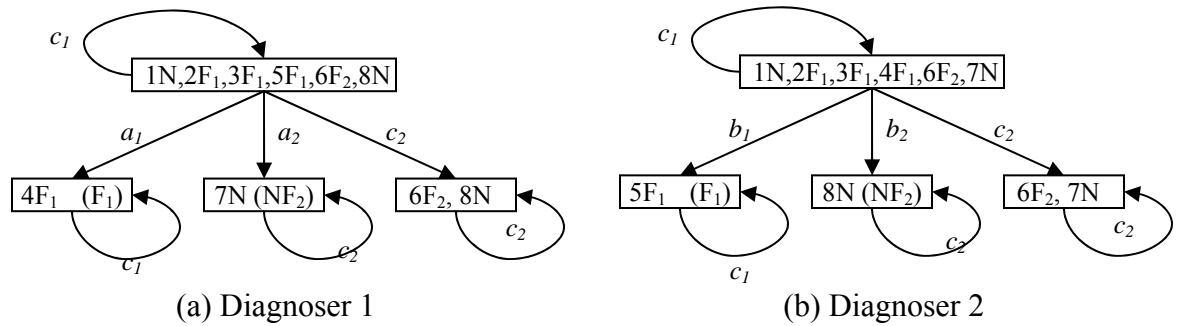


Fig. 7. Diagnosers for Examples 3 and 4

#### 4. Conditional Architecture

In the unconditional architecture, each local site makes “Fault” or “No Fault” decisions without any condition, and the global decision fusion block simply takes the disjunction of these local decisions. (In fact, no such fusion block is actually needed.) Under this architecture, Example 1 is NF-codiagnosable but not F-codiagnosable, which means that only fault-free traces can be detected with certainty. To diagnose faults in Example 1, we consider a decentralized diagnosis architecture where local diagnosis engines are allowed to make *conditional* decisions such as “Fault if nobody says No Fault” and “No Fault if nobody says Fault”. In analogy with [11], this architecture is called the *conditional architecture*. The global decision fusion block merges decentralized unconditional and conditional decisions. Inspired by the work in [11], we adopt the decision rules indicated in Table 1 below.

Table 1. Local decisions and their fusion in the conditional architecture

Case	Local Decision 1	Local Decision 2	Global Decision
1	Fault	Nothing	Fault
2	No Fault	Nothing	No Fault
3	Fault if nobody says No Fault	Nothing	Fault
4	Fault if nobody says No Fault	Fault	Fault
5	Fault if nobody says No Fault	No Fault	No Fault
6	No Fault if nobody says Fault	Nothing	No Fault
7	No Fault if nobody says Fault	Fault	Fault
8	No Fault if nobody says Fault	No Fault	No Fault
9	Nothing	Nothing	Nothing
10	Fault	No Fault	Diagnosis-conflict
11	Fault if nobody says No Fault	No Fault if nobody says Fault	Diagnosis-conflict

As can be seen from Cases 3-8, the conditional decisions “Fault if nobody says No Fault” and “No Fault if nobody says Fault” can be interpreted as “Fault” and “No Fault” decisions, respectively, but with lower priority. Namely, these conditional decisions take effect only if other sites are silent. The unconditional decisions “Fault” and “No Fault” override conditional decisions. There is a diagnosis conflict if and only if contradictory decisions of the same priority occur, i.e., contradictory unconditional decisions or contradictory conditional decisions. The properties of conditional diagnosability introduced in the next section will, by their very definitions, ensure that no such diagnosis conflicts occur.

#### 4.1 Notions of Conditional Codiagnosability

To draw parallels with Section 3 and the results in [11], we start by considering diagnosability properties associated with two special cases of the conditional architecture described in Table 1: *conditional F-codiagnosability* for the so-called conditional F-architecture and *conditional NF-codiagnosability* for the so-called conditional NF-architecture.

Under the conditional F-architecture, local sites have three types of decisions to choose from: “Fault”, “No Fault”, and “Fault if nobody says No Fault”. The fusion rules correspond to cases 1, 2, 3, 4, 5 and 9 in Table 1.

*Definition 6:* A prefix-closed and live language  $L$  is said to be conditionally F-codiagnosable w.r.t.  $\Sigma_f, P_1, P_2, \dots, P_n$ , if the following is true:

$(\exists k \in \mathbb{N})(\forall s \in L, s \text{ is faulty})(\forall st \in L, st \text{ is faulty}, |t| \geq k) (\exists i \in \{1, \dots, n\}, \forall u \in E_i(st), \text{ if } u \text{ is not faulty, then } \exists j \in \{1, \dots, n\}, \forall v \in ENF_j(u), v \text{ is not faulty}).$

In words, this definition means the following. For each sufficient long faulty trace  $st$ , there is a site  $i$  for which  $st$  might have the same projection as fault-free traces  $u_1, u_2, \dots$ , but for every fault-free trace  $u_k$  that belongs to site  $i$ 's estimate, there is a site  $j$  that can ensure that  $u_k$  is fault-free. That is, site  $i$  can infer that there is another site,  $j$ , that can recognize the fault-free trace with certainty. Therefore, site  $i$  can use the “Fault if nobody says No Fault” decision and site  $j$  will issue the “No Fault” decision overriding site  $i$  if  $u_k$  is the trace that the system executes. Note the use of the *ENF* estimator when considering non-faulty traces, as in Definition 2.

Under the dual conditional NF-architecture, local sites have three types of decisions to choose from: “No Fault”, “Fault”, and “No Fault if nobody says Fault”. The fusion rules correspond to cases 1, 2, 6, 7, 8 and 9 in Table 1.

*Definition 7:* A prefix-closed and live language  $L$  is said to be conditionally NF-codiagnosable w.r.t.  $\Sigma_f, P_1, P_2, \dots, P_n$ , if the following is true:

$(\exists k \in \mathbb{N})(\forall s \in L, s \text{ is not faulty})(\forall st \in L, st \text{ is not faulty}, |t| \geq k) (\exists i \in \{1, \dots, n\}, \forall u \in ENF_i(st), \text{ if } u \text{ is faulty, then } \exists j \in \{1, \dots, n\}, \forall v \in E_j(u), v \text{ is faulty}).$

The interpretation of this definition is as follows. For each sufficient long fault-free trace  $st$ , there is a site  $i$  for which  $st$  might have the same projection as faulty traces  $u_1, u_2, \dots$ , but for every faulty trace  $u_k$  that belongs to site  $i$ 's estimate, there is a site  $j$  that can ensure that  $u_k$  is faulty. That is, site  $i$  can infer that there is another site,  $j$ , that can recognize the faulty trace with certainty. Therefore, site  $i$  can use the “No Fault if nobody says Fault” decision and site  $j$  will issue the “Fault” decision overriding site  $i$  if  $u_k$  happens.

The two preceding definitions can be extended in a straightforward manner to the case of multiple faults, as was done in Definitions 3 and 4 in Section 3.1. We omit these definitions here and proceed directly to the case of conditional codiagnosability, the conditional version of Definition 5 in Section 3.1. Let us partition  $\Sigma_f$  as  $\Sigma_f = \Sigma_{f,F} \cup \Sigma_{f,NF}$ , where  $\Sigma_{f,F}$  is the set of fault events whose occurrence can be diagnosed and  $\Sigma_{f,NF}$  is the set of fault events whose absence can be diagnosed.

*Definition 8:* A prefix-closed and live language  $L$  is said to be conditionally codiagnosable w.r.t.  $\Sigma_{f,F}, \Sigma_{f,NF}, P_1, P_2, \dots, P_n$ , if

1.  $L$  is conditionally F-codiagnosable w.r.t.  $\Sigma_{f,F}, P_1, P_2, \dots, P_n$ ;
2.  $L$  is conditionally NF-codiagnosable w.r.t.  $\Sigma_{f,NF}, P_1, P_2, \dots, P_n$ .

#### 4.2 Properties of Conditional Codiagnosability

*Proposition 4:* If language  $L$  is codiagnosable w.r.t.  $\Sigma_{f,F}, \Sigma_{f,NF}, P_1, P_2, \dots, P_n$ , then it is conditionally F-codiagnosable and NF-codiagnosable w.r.t.  $\Sigma_{f,F} \cup \Sigma_{f,NF}, P_1, P_2, \dots, P_n$ .

*Proposition 5:* Conditional F-codiagnosability and conditional NF-codiagnosability are incomparable w.r.t. the same fault event and local projections.

Proposition 5 is proved by the two examples shown in Fig. 8, where  $\Sigma_{o,1} = \{a_1, a_2, c\}$  and  $\Sigma_{o,2} = \{b_1, b_2, c\}$ .

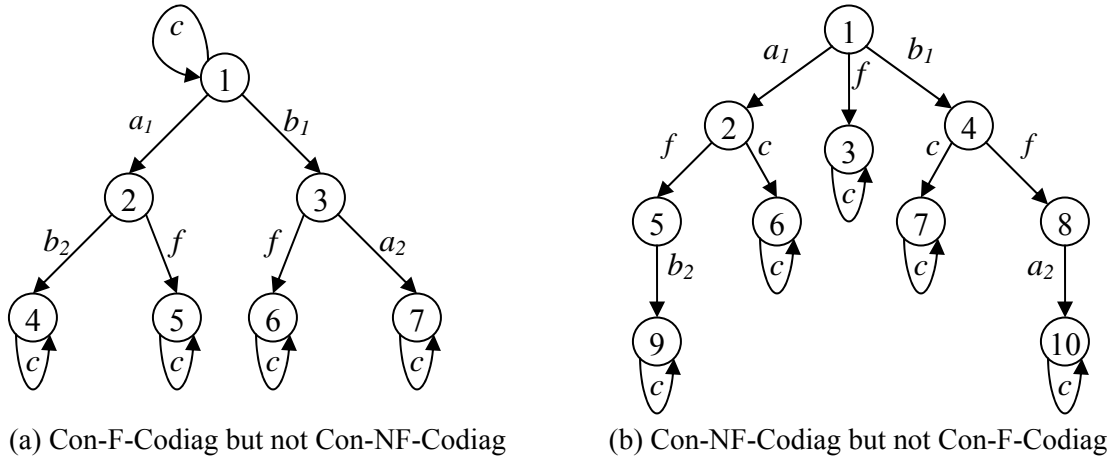


Fig. 8. Conditional F and NF codiagnosability are incomparable

*Proposition 6:* Conditional F-codiagnosability or NF-codiagnosability implies conditional codiagnosability, with the same fault events and projections. The reverse implication is not true in general.

*Proposition 7:* Conditional codiagnosability w.r.t.  $\Sigma_{f,F}, \Sigma_{f,NF}, \Sigma_{o,1}, \Sigma_{o,2}, \dots, \Sigma_{o,n}$  implies centralized diagnosability w.r.t. every fault event in  $\Sigma_{f,F}$  and  $\Sigma_{f,NF}$  and projection  $\Sigma_o = \Sigma_{o,1} \cup \Sigma_{o,2} \cup \dots \cup \Sigma_{o,n}$ . The reverse implication is not true in general.

In conclusion, the relationship among the different notions of codiagnosability introduced above is shown in Fig 9, where a directed arc indicates “implies”.

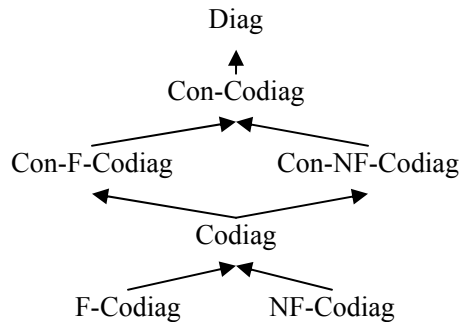


Fig. 9. Relationship among notions of codiagnosability

### 4.3 Discussion

It can be shown that the technique mentioned in Section 3.3 for verifying (unconditional) codiagnosability can be extended to develop polynomial time algorithms for testing conditional codiagnosability. The synthesis of special types of diagnosers to implement conditional decisions is a more intricate problem and it is not discussed here.

## 5. Conclusion

This paper has outlined the main features of a strategy for performing decentralized diagnosis of DES using architectures where local sites can issue several types of diagnosis decisions about the presence or absence of each fault, including so-called conditional decisions of the type “Fault if nobody says No Fault” and “No Fault if nobody says Fault.” The use of such decentralized architectures allows for diagnosing larger classes of systems that can be diagnosed under the decentralized architecture corresponding to Protocol 3 in [1]. Moreover, the various notions of codiagnosability that characterize these new architectures are verifiable in polynomial time in the size of the state space of the system.

## Acknowledgements

This research is supported in part by NSF under grant CCR-0325571 and by ONR under grant N00014-03-1-0232.

## References

- [1] R. Debouk, S. Lafortune and D. Teneketzis, “Coordinated decentralized protocols for failure diagnosis of discrete event systems,” *Discrete Event Dynamic Systems: Theory and Applications*, vol.10, no.1-2, pp.33–86, 2000.
- [2] S., Lafortune, D. Teneketzis, M. Sampath, R. Sengupta, and K. Sinnamohideen, “Failure diagnosis of dynamic systems: An approach based on discrete event systems,” in *Proc. 2001 American Control Conference*. Arlington, VA, USA. pp. 2058–2071, 2001.
- [3] W. Qiu and R. Kumar, “Decentralized Failure Diagnosis of Discrete Event Systems,” in *Proc. of 7<sup>th</sup> Int. Workshop on Discrete Event Systems*, Reims, France, Sept. 22-24, 2004.
- [4] K. Rudie and W. M. Wonham, “Think globally, act locally: Decentralized supervisory control,” *IEEE Trans. Automatic Control*, 37(11), 1692-1708, Nov. 1992.
- [5] K. Rudie and J. Willems, “The Computational Complexity of Decentralized Discrete Event Control Problems,” *IEEE Trans. Automatic Control*, 40(7), 1313-1319, 1995.
- [6] M. Sampath, R. Sengupta, K. Sinnamohideen, S. Lafortune and D. Teneketzis, “Diagnosability of discrete event systems,” *IEEE Trans. Automatic Control*, 40(9), 1555–1575. Sept. 1995.
- [7] M. Sampath, R. Sengupta, K. Sinnamohideen, S. Lafortune and D. Teneketzis, “Failure diagnosis using discrete event models,” *IEEE Trans. Control Systems Technology*, 4(2), 105–124, 1996.
- [8] R. Sengupta, “Diagnosis and Communication in Distributed Systems,” in *Proc. of 4<sup>th</sup> International Workshop on Discrete Event Systems*, Cagliari, Italy, Aug 26-28, 1998.
- [9] R. Sengupta and S. Tripakis, “Decentralized diagnosability of regular languages is undecidable,” in *Proc. of 41st IEEE Conf. on Decision and Control*, vol. 1, pp.423-428, 2002.
- [10] T.-S. Yoo and S. Lafortune, “A General Architecture for Decentralized Supervisory Control of Discrete-event Systems,” *Discrete Event Dynamic Systems: Theory and Applications*, Vol. 12, No. 3, pp. 335-377, July, 2002.
- [11] T.-S. Yoo and S. Lafortune, “Decentralized supervisory control with conditional decisions: supervisor existence,” *IEEE Trans. Automatic Control*, In print.
- [12] T.-S. Yoo and S. Lafortune, “Polynomial-time verification of diagnosability of partially-observed discrete-event systems,” *IEEE Trans. Automatic Control* 47(9), 1491-1495, 2002.