Hierarchical Distributed Diagnosis under Global Consistency

R. Su and **W.M. Wonham** [1]

Abstract. In this paper we first introduce the concept of the supremal global support to capture interaction among local components. Then we propose a hierarchical diagnosis approach, whose main objective is to achieve supremal global support at the bottom level.

1 Introduction

There is a large body of research on fault diagnosis for discrete-event systems (DES), e.g. approaches based on finite-state automata [9] [13], and Petri nets [1]. A major challenge in fault diagnosis for DES is the high space complexity of storing diagnosers and generating diagnostic results, in centralized approaches [8] [16]. To reduce space complexity, researchers have turned their attention to distributed architecture, consisting of a set of simple modular components linked in a communication network, e.g. [10] [1] [7]. The resulting diagnoser consists of a set of relatively simple local diagnosers. Therefore, memory usage for storing the distributed diagnoser is low, but with a price of temporal delay introduced by inter-component communication. Approaches like [10] [1] [7] have the shortcoming that the £nal disgnoser is derived from a centralized intermediate system model, typically the synchronous product of all local modules. Therefore they still must face high space complexity during the intermediate design stage before the final diagnoser is constructed. To avoid such a centralized intermediate system model, some other distributed approaches like [14] [5] [3] propose a suitable consistency property which is required to be held among local modules during online diagnosis. By this means those approaches have low space complexity not only in the diagnoser design stage but also in online diagnosis. Their shortcoming is that the inter-component communication procedures, designed to achieve the corresponding consistency property, may not always £nitely terminate when a general system with an arbitrary network structure is under consideration. Therefore the temporal delay during online diagnosis becomes a major concern.

There is another way to manage complexity in large-scale systems - hierarchical approaches, which have been studied in the AI community for many years, e.g. [2] [11] [6]. Nevertheless, in those approaches each level possesses a centralized model instead of a distributed model. This is because the authors deal mainly with digital circuits whose behavior can be concisely expressed by simple logic expressions. So the size of a centralized model is not of great import; time complexity is the authors' major concern. As we have seen, a centralized model for automaton-based diagnosis usually leads to high space complexity. Another feature of AI hierarchical approaches is that their construction procedures for high-level abstract models usually are rather complicated, inasmuch as the models lack language structure, making the projection abstraction operation complicated. Thus checking consistency between two different levels is not a trivial matter. By contrast, we will see later that projection is not problematic for language-based fault diagnosis. Some work on distributed systems addresses hierarchical distributed diagnosis, e.g. [12] [4]. Nevertheless, these approaches treat each local component simply as a node without any further evolution or input-out structure, and their objective is to determine which node fails to make a correct response. So this work is more about distributed fault detection than distributed fault diagnosis, which includes fault isolation, and its major concern is time complexity rather than space complexity.

This paper is organized as follows. In Section 2 we introduce the concept of hierarchical distributed model. Then in Section 3 we propose a hierarchical distributed diagnosis procedure and draw conclusions in Section 4.

2 Hierarchical Distributed Model

2.1 Supremal Global Support

Let I be an index set. For each $i \in I$ let Σ_i be an event set. As usual [15], we use "||" to represent synchronous product. All natural projections in this paper follow the following notational rules: given an index set J with a predefined collection of event sets $\{\Sigma_j | j \in J\}$, we define

$$\Sigma_J := \cup_{j \in J} \Sigma_j$$

We use $\mathrm{P}_{J,i}$ to mean the following natural projection,

$$\mathrm{P}_{J,i} : \Sigma_J^* \to (\Sigma_J \cap \Sigma_i)^*$$

Let $\mathcal{L} = \{\mathrm{L}_i \subseteq \Sigma_i^* | i \in I\}$ be a set of languages.

Definition 2.1 $\mathcal{L} = \{\mathrm{L}_i | i \in I\}$ is *globally consistent* if for each $i \in I$, $\mathrm{L}_i = \mathrm{P}_{I,i}(||_{j \in I} \mathrm{L}_j)$. □

Given $\mathcal{L} = \{\mathrm{L}_i \subseteq \Sigma_i^* | i \in I\}$, a set $\{\mathrm{E}_i \subseteq \mathrm{L}_i | i \in I\}$ is a *global support* of $\mathcal{L}$ if it is globally consistent, namely

$$(\forall i \in I)\, \mathrm{E}_i = \mathrm{P}_{I,i}(||_{j \in I} \mathrm{E}_j)$$

Let $\Delta(\mathcal{L})$ be the set of all global supports of $\mathcal{L}$. Clearly $\Delta(\mathcal{L})$ is not empty because it contains $\{\mathrm{E}_i = \varnothing | i \in I\}$. We define a binary relation "$\leq$" among elements in $\prod_{i \in I} \mathrm{Pwr}(\Sigma_i^*)$ as follows:

[1] Dept. ECE, University of Toronto,10 King's College Road, Toronto, ON M5S 3G4, Canada email: surong,wonham@control.utoronto.ca, telephone:416-9784124, fax:416-9780804

$$\{\mathrm{E}_i | i \in I\} \leq \{\tilde{\mathrm{E}}_i | i \in I\} \iff (\forall i \in I)\, \mathrm{E}_i \subseteq \tilde{\mathrm{E}}_i$$

An element $\mathcal{E} \in \Delta(\mathcal{L})$ is called the *supremal global support* of $\Delta(\mathcal{L})$, written $\mathrm{Sup}\Delta(\mathcal{L})$, if $(\forall \mathcal{E}' \in \Delta(\mathcal{L}))\, \mathcal{E}' \leq \mathcal{E}$.

Proposition 2.1 $\mathrm{Sup}\Delta(\mathcal{L}) = \{\mathrm{E}_i := \mathrm{P}_{I,i}(||_{j \in I} \mathrm{L}_j) | i \in I\}$. □

Definition 2.2 Let $\{\Sigma_i | i \in I\}$ be a collection of event sets. A *distributed reference model* $\mathcal{L}$ is a set of closed languages $\{\mathrm{L}_i \subseteq \Sigma_i^* | i \in I\}$, where L_i is a *local component of* $\mathcal{L}$. There is a subset $\Sigma_{io} \subseteq \Sigma_i$ called the *observable event set of* i, which is not necessarily pairwise disjoint with other observable event sets, and another set $\Sigma_{if} \subseteq \Sigma_i$ called *fault event set of* i, which is pairwise disjoint with any other event set. □

Suppose we have a reference model $\mathcal{L} = \{\mathrm{L}_i \subseteq \Sigma_i^* | i \in I\}$. For each $i \in I$ we have a local observable event set $\Sigma_{io} \subseteq \Sigma_i$. Let

$$\mathrm{P}_{i,o} : \Sigma_i^* \to \Sigma_{io}^*$$

be the natural projection for component i. The *symptom* of a distributed system is an I-tuple $\mathcal{O} = \{u_i \in \Sigma_{io}^* | i \in I\}$, where each u_i is a *local symptom* of component i. Now we propose a distributed diagnosis problem under global consistency.

Distributed Diagnosis under Global Consistency

1. **Local Computation**: $(\forall i \in I)\, \mathrm{M}_i := \mathrm{P}_{i,o}^{-1}(u_i) \cap \mathrm{L}_i$
2. **Global Consistency**: $\mathcal{E} := \mathrm{Sup}\Delta(\{\mathrm{M}_i | i \in I\})$ □

The motivation to formulate a distributed diagnosis problem as above can be simply explained as follows. Local computation is nothing but an estimation process, namely given observation u_i ($i \in I$) we find all strings that can exhibit u_i. Once local estimates are obtained, we use supremal global support to capture "agreement" among these local estimates. It turns out that achieving the supremal global support is quite time consuming when a system consists of many local components. To reduce time complexity, we propose in the rest of this paper a hierarchical approach.

2.2 Hierarchical Model Construction

In this paper we con£ne attention to a two-level hierarchy. It will be clear that the approach can be easily extended to a general hierarchy. Suppose we have a distributed reference model $\mathcal{L} = \{\mathrm{L}_i \subseteq \Sigma_i^* | i \in I\}$. Let $\hat{I}$ be a new index set with a given partition $\{J_{\hat{k}} \subseteq I | \hat{k} \in \hat{I}\}$ on I. For each $i \in I$ let $\Sigma_{io} \subseteq \Sigma_i$ be the observable event set of component i. To preserve the consistency of observation, we assume that **if an event is observable in a local component, then it is observable in any local component that contains it as a local event**, namely

$$(\forall i, j \in I)(\forall \sigma \in \Sigma_i \cap \Sigma_j)\, \sigma \in \Sigma_{io} \iff \sigma \in \Sigma_{jo}$$

For notational consistency, elements and subsets of $\hat{I}$ have hat symbol, e.g. $\hat{i}$, $\hat{j}$, but no elements and subsets of I have it. For each $\hat{k} \in \hat{I}$, let $\Sigma_{\hat{k}o} := \cup_{i \in J_{\hat{k}}} \Sigma_{io}$. We call $\Sigma_{\hat{k}o}$ the *observable event set* of module $J_{\hat{k}}$. We define

$$\Sigma_{\hat{k}} := \Sigma_{\hat{k}o} \cup (\Sigma_{J_{\hat{k}}} \cap \Sigma_{I - J_{\hat{k}}}) \tag{1}$$

as the *critical event set* of module $J_{\hat{k}}$. Thus, $\Sigma_{\hat{k}}$ contains the observable events of each component $i \in J_{\hat{k}}$ and all events shared by $J_{\hat{k}}$ and other modules $I - J_{\hat{k}}$. Recall that, with the same notational rules for natural projection described in Section 2.1, $\mathrm{P}_{J_{\hat{k}},\hat{k}}$ represents the natural projection $\mathrm{P}_{J_{\hat{k}},\hat{k}} : \Sigma_{J_{\hat{k}}}^* \to \Sigma_{\hat{k}}^*$. We define

$$\mathrm{L}_{\hat{k}} := \mathrm{P}_{J_{\hat{k}},\hat{k}}(||_{i \in J_{\hat{k}}} \mathrm{L}_i) \tag{2}$$

as an *abstract local module* of $\mathcal{L}$. The set $\hat{\mathcal{L}} := \{\mathrm{L}_{\hat{k}} | \hat{k} \in \hat{I}\}$ is called the *abstract model* of $\mathcal{L}$. To see how to build an abstract model, consider the following example.

Figure 1 depicts a simple paper acquisition model (SPAM) which

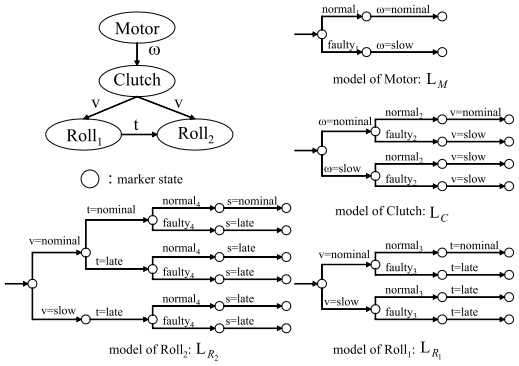

Figure 1. Simple Paper Acquisition Model

consists of one motor, one clutch and two rolls. The index set is $I = \{M, C, R_1, R_2\}$ and the corresponding distributed reference model is $\mathcal{L} = \{\mathrm{L}_M, \mathrm{L}_C, \mathrm{L}_{R_1}, \mathrm{L}_{R_2}\}$. To construct a high-level abstract model, we partition I into two subsets $J_1 = \{M, C\}$ and $J_2 = \{R_1, R_2\}$. Since there are no observable events in either Motor or Clutch, we have $\Sigma_{1o} = \varnothing$, namely the observable event set of module J_1 is empty. In module J_2, Roll_1 has no observable events but Roll_2 has observable events $\{s = nominal, s = late\}$. Thus $\Sigma_{2o} = \{s = nominal, s = late\}$. By expression (1) the critical event set of J_1 is

$$\Sigma_1 := \Sigma_{J_1} \cap \Sigma_{J_2} = (\Sigma_M \cup \Sigma_C) \cap (\Sigma_{R_1} \cup \Sigma_{R_2})$$

Similarly, we have for the critical event set of J_2

$$\Sigma_2 = \Sigma_{2o} \cup (\Sigma_{J_1} \cap \Sigma_{J_2})$$

By the notational rules for projections, we write

$$\mathrm{P}_{J_1,1} : \Sigma_{J_1}^* \to \Sigma_1^* \text{ and } \mathrm{P}_{J_2,2} : \Sigma_{J_2}^* \to \Sigma_2^*$$

Then by expression (2) we have abstract local modules

$$\begin{aligned} \mathrm{L}_1 &= \mathrm{P}_{J_1,1}(\mathrm{L}_M || \mathrm{L}_C) \\ \mathrm{L}_2 &= \mathrm{P}_{J_2,2}(\mathrm{L}_{R_1} || \mathrm{L}_{R_2}) \end{aligned}$$

Figure 2 depicts L_1 and L_2, where the symbol v between two local modules in Figure 2 means that two modules share events assigning values to variable v, namely $v = nominal$ and $v = slow$. ♢

Suppose the symptom set is $\mathcal{O} = \{u_i \in \Sigma_{io}^* | i \in I\}$. Then we can perform local computation in each local component $i \in I$ to obtain the preliminary estimate,

$$\mathrm{M}_i := \mathrm{P}_{i,o}^{-1}(u_i) \cap \mathrm{L}_i \tag{3}$$

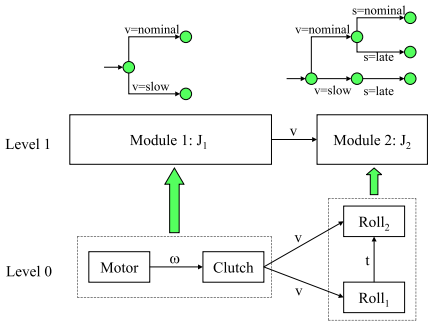

Figure 2. Abstract Model for SPAM

Let $\mathcal{M} := \{\mathrm{M}_i | i \in I\}$. For each $\hat{k} \in \hat{I}$ let

$$\mathrm{M}_{\hat{k}} := \mathrm{P}_{J_{\hat{k}},\hat{k}}(||_{i \in J_{\hat{k}}} \mathrm{M}_i) \tag{4}$$

Thus, $\hat{\mathcal{M}} := \{\mathrm{M}_{\hat{k}} | \hat{k} \in \hat{I}\}$ is an abstract model of $\mathcal{M}$. Let

$$\mathrm{P}_{\hat{k},\hat{k}o} : \Sigma^*_{\hat{k}} \to \Sigma^*_{\hat{k}o}$$

be the natural projection. We have the following result.

Proposition 2.2 $(\forall \hat{k} \in \hat{I})\, \mathrm{M}_{\hat{k}} = \mathrm{L}_{\hat{k}} \cap \mathrm{P}^{-1}_{\hat{k},\hat{k}o}(||_{i \in J_{\hat{k}}} \{u_i\})$ □

Prop. 2.2 tells us how to compute $\hat{\mathcal{M}}$ directly from the abstract model $\hat{\mathcal{L}}$ and the symptom set $\mathcal{O}$ without actually using expression (4). On the other hand, we can derive that

$$\mathrm{P}_{I,\hat{I}}(||_{i \in I} \mathrm{M}_i) = ||_{\hat{k} \in \hat{I}} \mathrm{M}_{\hat{k}} \tag{5}$$

We can define the supremal global support of $\Delta(\hat{\mathcal{M}})$ in the same way as for $\Delta(\mathcal{L})$. Then by Prop. 2.1 we have

$$\mathrm{Sup}\Delta(\hat{\mathcal{M}}) = \{\mathrm{E}_{\hat{i}} := \mathrm{P}_{\hat{I},\hat{i}}(||_{\hat{k} \in \hat{I}} \mathrm{M}_{\hat{k}}) | \hat{i} \in \hat{I}\}$$

From (5) we derive the relation between $\mathrm{Sup}\Delta(\mathcal{L}) = \{\mathrm{E}_j | j \in I\}$ and $\mathrm{Sup}\Delta(\hat{\mathcal{L}}) = \{\mathrm{E}_{\hat{i}} | \hat{i} \in \hat{I}\}$ as follows:

$$(\forall \hat{i} \in \hat{I})\, \mathrm{E}_{\hat{i}} = \mathrm{P}_{I,\hat{i}}(||_{j \in I} \mathrm{E}_j) \tag{6}$$

Let

$$\mathbb{W} := \textstyle\prod_{i \in I} \mathrm{Pwr}(\Sigma^*_i) \quad \mathbb{O} := \prod_{i \in I} \Sigma^*_{io}$$
$$\hat{\mathbb{W}} := \textstyle\prod_{\hat{k} \in \hat{I}} \mathrm{Pwr}(\Sigma^*_{\hat{k}}) \quad \hat{\mathbb{O}} := \prod_{\hat{k} \in \hat{I}} \mathrm{Pwr}(\Sigma^*_{\hat{k}o})$$

Then Proposition 2.2, together with (2), (3), (4), (5) and (6), makes the following diagram commute, where maps f, g, h, $\hat{h}$, p, $\hat{p}$ and q are defined below. For brevity of notation, we first define some new operations.

1. $\mathrm{P}_o^{-1} : \mathbb{O} \to \mathbb{W}$,

$$\{u_i | i \in I\} \mapsto \mathrm{P}_o^{-1}(\{u_i | i \in I\}) := \{\mathrm{P}^{-1}_{i,o}(u_i) | i \in I\}$$

2. $\sqcap : \mathbb{W} \times \mathbb{W} \to \mathbb{W}$,

$$\{A_i | i \in I\} \sqcap \{B_i | i \in I\} := \{A_i \cap B_i | i \in I\}$$

3. $\hat{\mathrm{P}}_o^{-1} : \hat{\mathbb{O}} \to \hat{\mathbb{W}}$, for each $\hat{\mathcal{O}} = \{\mathrm{U}_{\hat{k}} | \hat{k} \in \hat{I}\} \in \hat{\mathbb{O}}$,

$$\hat{\mathcal{O}} \mapsto \hat{\mathrm{P}}_o^{-1}(\hat{\mathcal{O}}) := \{\mathrm{P}^{-1}_{\hat{k},\hat{k}o}(\mathrm{U}_{\hat{k}}) | \hat{k} \in \hat{I}\}$$

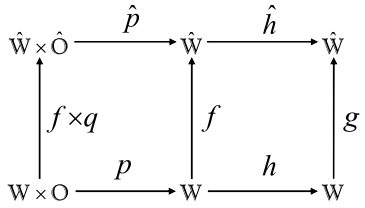

Figure 3. Two-Level Hierarchical Computation

4. $\hat{\sqcap} : \hat{\mathbb{W}} \times \hat{\mathbb{W}} \to \hat{\mathbb{W}}$, for each $\{A_{\hat{k}} | \hat{k} \in \hat{I}\}$, $\{B_{\hat{k}} | \hat{k} \in \hat{I}\} \in \hat{\mathbb{W}}$,

$$\{A_{\hat{k}} | \hat{k} \in \hat{I}\} \hat{\sqcap} \{B_{\hat{k}} | \hat{k} \in \hat{I}\} := \{A_{\hat{k}} \cap B_{\hat{k}} | \hat{k} \in \hat{I}\}$$

Now we can define the maps in Figure 3.

1. $f : \mathbb{W} \to \hat{\mathbb{W}}$,

$$f(\{\mathrm{L}_i | i \in I\}) := \{\mathrm{L}_{\hat{k}} = \mathrm{P}_{J_{\hat{k}},\hat{k}}(||_{j \in J_{\hat{k}}} \mathrm{L}_j) | \hat{k} \in \hat{I}\}$$

2. $g : \mathbb{W} \to \hat{\mathbb{W}}$,

$$g(\{\mathrm{E}_i | i \in I\}) := \{\mathrm{E}_{\hat{k}} = \mathrm{P}_{I,\hat{k}}(||_{j \in I} \mathrm{E}_j) | \hat{k} \in \hat{I}\}$$

3. $h : \mathbb{W} \to \mathbb{W} : \mathcal{M} \mapsto h(\mathcal{M}) := \mathrm{Sup}\Delta(\mathcal{M})$
4. $\hat{h} : \hat{\mathbb{W}} \to \hat{\mathbb{W}} : \hat{\mathcal{M}} \mapsto \hat{h}(\hat{\mathcal{M}}) := \mathrm{Sup}\Delta(\hat{\mathcal{M}})$
5. $p : \mathbb{W} \times \mathbb{O} \to \mathbb{W} : (\mathcal{L}, \mathcal{O}) \mapsto p(\mathcal{L}, \mathcal{O}) := \mathrm{P}_o^{-1}(\mathcal{O}) \sqcap \mathcal{L}$
6. $\hat{p} : \hat{\mathbb{W}} \times \hat{\mathbb{O}} \to \hat{\mathbb{W}} : (\hat{\mathcal{L}}, \hat{\mathcal{O}}) \mapsto \hat{p}(\hat{\mathcal{L}}, \hat{\mathcal{O}}) := \hat{\mathrm{P}}_o^{-1}(\hat{\mathcal{O}}) \hat{\sqcap} \hat{\mathcal{L}}$
7. $q : \mathbb{O} \to \hat{\mathbb{O}}$,

$$q(\{u_i | i \in I\}) := \{\mathrm{U}_{\hat{k}} = \mathrm{P}_{J_{\hat{k}},\hat{k}o}(||_{j \in J_{\hat{k}}} \{u_i\}) | \hat{k} \in \hat{I}\}$$

Given $\mathcal{L} \in \mathbb{W}$ and $\mathcal{O} \in \mathbb{O}$, our objective is to compute

$$\mathcal{E} := \mathrm{Sup}\Delta(\mathrm{P}_o^{-1}(\mathcal{O}) \sqcap \mathcal{L})$$

By Figure 3, $\mathcal{E}$ is given by

$$\mathcal{E} = h(p(\mathcal{L}, \mathcal{O}))$$

But instead of using $h \circ p$, where "$\circ$" represents map composition, we first compute

$$\begin{aligned} \hat{\mathcal{E}} &:= \hat{h} \circ \hat{p} \circ (f \times q)(\mathcal{L}, \mathcal{O}) = \hat{h}(\hat{p}(f(\mathcal{L}), q(\mathcal{O}))) \\ &= \mathrm{Sup}\Delta(\mathrm{P}_{\hat{o}}^{-1}(q(\mathcal{O})) \hat{\sqcap} f(\mathcal{L})) \end{aligned}$$

The reason is that computing supremal global support at the high level is usually easier than doing so at the bottom level, thanks to abstraction. The following result shows how to obtain $\mathcal{E}$ from $\hat{\mathcal{E}}$.

Proposition 2.3 Let $\mathcal{M} = \{\mathrm{M}_i | i \in I\} \in \mathbb{W}$, $\hat{\mathcal{M}} = \{\mathrm{M}_{\hat{k}} | \hat{k} \in \hat{I}\} \in \hat{\mathbb{W}}$. Let f be defined as in Figure 3. Suppose

$$f(\mathcal{M}) = \hat{\mathcal{M}}$$

If $\{\mathrm{E}_i | i \in I\} = \mathrm{Sup}\Delta(\mathcal{M})$, $\{\mathrm{E}_{\hat{k}} | \hat{k} \in \hat{I}\} = \mathrm{Sup}\Delta(\hat{\mathcal{M}})$, then we have that $(\forall \hat{k} \in \hat{I})(\forall i \in J_{\hat{k}})\, \mathrm{E}_i = \mathrm{P}_{J_{\hat{k}},i}((||_{j \in J_{\hat{k}}} \mathrm{M}_j) || \mathrm{E}_{\hat{k}})$. □

Once we have the supremal global support at the high level, say $\hat{\mathcal{E}}$, Prop. 2.3 tells us how to use $\hat{\mathcal{E}}$ to obtain the supremal global support $\mathcal{E}$ at the bottom level. Notice that concurrent computation can be done in different modules at the bottom level, which can substantially

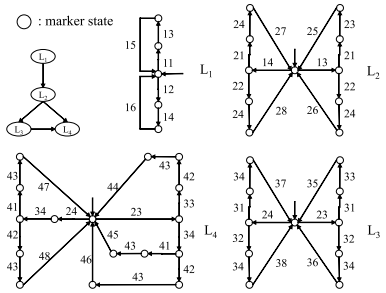

Figure 4. Simple Example

improve time complexity of achieving the supremal global support. This is the principal reason for introducing the hierarchical approach.

As an illustration, consider the following example depicted in Figure 4, where $I = \{1, 2, 3, 4\}$. Since all local components are pre£x-closed languages, each state in Figure 4 is a marker state. Suppose the local observable event sets are $\Sigma_{1o} = \varnothing$, $\Sigma_{2o} = \varnothing$, $\Sigma_{3o} = \{33, 34\}$ and $\Sigma_{4o} = \{33, 34, 41, 42\}$. Let $\hat{I} := \{a, b\}$, where $J_a = \{1, 2\}$ and $J_b = \{3, 4\}$. According to expression (1) we have the following critical event sets,

$$\Sigma_a := \{23, 24\} \text{ and } \Sigma_b := \{23, 24, 33, 34, 41, 42\}$$

At the second level, we build two modules: $\mathrm{L}_a := \mathrm{P}_{J_a,a}(\mathrm{L}_1||\mathrm{L}_2)$ and $\mathrm{L}_b := \mathrm{P}_{J_b,b}(\mathrm{L}_3||\mathrm{L}_4)$, shown in Figure 5. Suppose the lo-

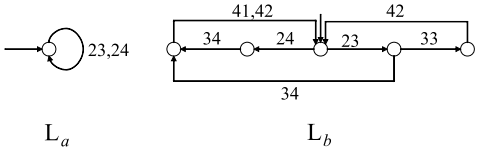

Figure 5. Second Level Abstract Model $\hat{\mathcal{L}} = \{\mathrm{L}_a, \mathrm{L}_b\}$

cal symptom set at the bottom level is: $\mathcal{O} = \{\epsilon, \epsilon, 34, (34)(41)\}$, namely event 34 has been executed in L_3 and events 34, 41 in L_4. Then we can construct the local symptom set $\hat{\mathcal{O}} = \{\mathrm{U}_a = ||_{j\in J_a}\{u_j\}, \mathrm{U}_b = ||_{j\in J_b}\{u_j\}\}$ at the second level. So in L_a we have $\mathrm{U}_a = \{\epsilon\} = \{\epsilon\}||\{\epsilon\}$ as the local observation, and in L_b we have $\mathrm{U}_b = \{34\}||\{(34)(41)\} = \{(34)(41)\}$. By Proposition 2.2 we get that

$$\mathrm{M}_a = \mathrm{L}_a||\mathrm{P}^{-1}_{a,ao}(\{\epsilon\}) \text{ and } \mathrm{M}_b = \mathrm{L}_b||\mathrm{P}^{-1}_{b,bo}(\{34\}||\{(34)(41)\})$$

Figure 6 depicts results of $\hat{\mathcal{M}} := \{\mathrm{M}_a, \mathrm{M}_b\}$. Then we compute $\hat{\mathcal{E}} := \{\mathrm{E}_a, \mathrm{E}_b\} = \mathrm{Sup}\Delta(\hat{\mathcal{M}})$, which are also shown in Figure 6. So far, we have finished computation at the second level. Now we descend to the first level and compute the supremal global support based on the local observation set $\mathcal{O} = \{\epsilon, \epsilon, 34, (34)(41)\}$ and messages E_a, E_b passed down from the second level. Since no observation is made from L_1 and L_2, we have $\mathrm{M}_1 = \mathrm{L}_1$ and $\mathrm{M}_2 = \mathrm{L}_2$. By Prop. 2.3, we can compute their local estimates based on E_a as follows:

$$\mathrm{E}_1 = \mathrm{P}_{J_a,1}(\mathrm{M}_1||\mathrm{M}_2||\mathrm{E}_a) \text{ and } \mathrm{E}_2 = \mathrm{P}_{J_a,2}(\mathrm{M}_1||\mathrm{M}_2||\mathrm{E}_a)$$

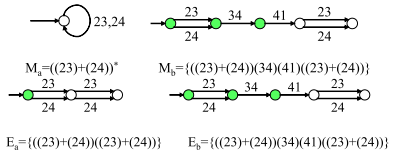

Figure 6. M_a, M_b, E_a and E_b

Figure 7 depicts the results of E_1 and E_2, showing clearly how we manipulate finite-state automata to implement the distributed diag-

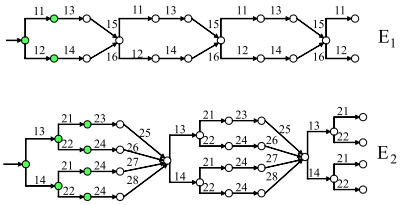

Figure 7. E_1 and E_2

nosis procedure as proposed in this paper and based on languages. Now we turn our attention to L_3 and L_4. Since each of them has local observation, we first compute their preliminary estimate M_3 and M_4:

$$\mathrm{M}_3 := \mathrm{L}_3 \cap \mathrm{P}^{-1}_{3,o}(\{34\}) \text{ and } \mathrm{M}_4 := \mathrm{L}_4 \cap \mathrm{P}^{-1}_{4,o}(\{(34)(41)\})$$

The left picture of Figure 8 depicts the results of M_3 and M_4. Then

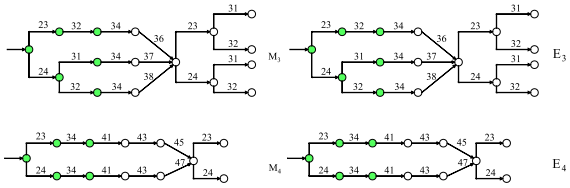

Figure 8. M_3, M_4 (Left) and E_3, E_4 (Right)

by Prop. 2.3 we can compute E_3 and E_4 based on E_b as follows:

$$\mathrm{E}_3 = \mathrm{P}_{J_b,3}(\mathrm{M}_3||\mathrm{M}_4||\mathrm{E}_b) \text{ and } \mathrm{E}_4 = \mathrm{P}_{J_b,4}(\mathrm{M}_3||\mathrm{M}_4||\mathrm{E}_b)$$

The right picture of Figure 8 depicts the results of E_3 and E_4. So far, we have the supremal global support $\mathcal{E} = \{\mathrm{E}_1, \mathrm{E}_2, \mathrm{E}_3, \mathrm{E}_4\}$ at the first level. On using the "brute-force" method (which, by the way, yields 1153 states and 3118 transitions for the size of $\mathrm{M}_1||\mathrm{M}_2||\mathrm{M}_3||\mathrm{M}_4$), we confirm that all results generated from hierarchical distributed computation are correct. In this example, at the first level, computing E_1 and E_2 is independent of computing E_3 and E_4, making it possible to do concurrent computation with the two disjoint sets ($\{\mathrm{L}_1, \mathrm{L}_2\}$ and $\{\mathrm{L}_3, \mathrm{L}_4\}$).

So far we have described how to construct a two-level hierarchy and perform hierarchical computation in it. If we assign the symptom set at the bottom level as $\mathcal{O} := \{\mathrm{U}_i := \{u_i\} \subseteq \Sigma_{io}^* | i \in I\}$, then we can easily check that all the results above hold in a general hierarchy; and between adjacent levels there is a commutative diagram according to Figure 3 and Propositions 2.2 and 2.3. As a summary, the main motivation for such hierarchical distributed computation is that computing the supremal global support at the high level is usually easier because of the relatively simple model at the high level due to abstraction. When computation reaches the bottom level, where the model is the most complicated, the problem of computing the supremal global support has been converted into many small problems where each problem only involves a single local module with a relatively small number of local components. This is nothing but another application of divide-and-conquer. Next, we introduce multi-resolution diagnosis.

3 Multi-Resolution Diagnosis

The true strength of a hierarchical computation is demonstrated in a situation where usually highly selective decisions are made. Computation should be carried out only in a module that may be potentially faulty. Now the question is, how do we know for sure that a module is functioning normally?

For simplicity, we consider only a two-level hierarchy. But the results can be easily applied to a general multiple-level hierarchy. Let $\mathcal{L} := \{\mathrm{L}_i | i \in I\}$ be the bottom level distributed reference model and $\hat{\mathcal{L}} := \{\mathrm{L}_{\hat{k}} | \hat{k} \in \hat{I}\}$ the abstract model, which is built from $\mathcal{L}$ as proposed in the previous section. As before, the observable event set of $\mathrm{L}_{\hat{k}}$ is $\Sigma_{\hat{k}o} := \cup_{i \in J_{\hat{k}}} \Sigma_{io}$. Now we can discuss hierarchical fault detection.

In model-based diagnosis a system model must be able to describe both normal and faulty behavior of the system. Fault detection is nothing but a comparison process, during which the system's current observable behavior is compared with the expected normal behavior that is observable. If some observable discrepancy is found then we say the system is *observably faulty* and fault isolation may be required. Otherwise, we say the system is *observably normal*. Without considering measurement error, an observably faulty system is faulty in its true meaning. Nevertheless, an observably normal system may not be *normal*, but simply hasn't demonstrated the ill effect of a fault that has occurred in the system but has not yet propagated to sensors. In this section we need to distinguish explicitly between a truly normal system and a system that is only observably normal, so that we can propose a procedure which carries out computation only in the modules of the hierarchy that are not truly normal. To this end, at the bottom level $\mathcal{L}$, we define a normal distributed model. For each $i \in I$ let

$$\mathrm{W}_i := (\Sigma_i - \Sigma_{if})^* \cap \mathrm{L}_i$$

Thus no string $s \in \mathrm{W}_i$ contains a fault event. Clearly, W_i is also pre£x-closed. For each $\hat{k} \in \hat{I}$ let $\mathrm{L}_{J_{\hat{k}}} := ||_{i \in J_{\hat{k}}} \mathrm{L}_i$ and $\mathrm{W}_{J_{\hat{k}}} := ||_{i \in J_{\hat{k}}} \mathrm{W}_i$. Then we define,

$$\mathrm{L}_{\hat{k},norm} := \mathrm{P}_{J_{\hat{k}},\hat{k}}(\mathrm{W}_{J_{\hat{k}}}) - \mathrm{P}_{J_{\hat{k}},\hat{k}}(\mathrm{L}_{J_{\hat{k}}} - \mathrm{W}_{J_{\hat{k}}}) \qquad (7)$$

For each string $s \in \mathrm{L}_{\hat{k},norm}$ we have

$$\mathrm{P}^{-1}_{J_{\hat{k}},\hat{k}}(s) \cap \mathrm{L}_{J_{\hat{k}}} \subseteq \mathrm{W}_{J_{\hat{k}}}$$

namely s contains no fault event. Thus, when s is executed in the abstract module $\mathrm{L}_{\hat{k}}$ we know that every local component in the module $J_{\hat{k}}$ is absolutely normal. We call $\mathrm{L}_{\hat{k},norm}$ the *truly normal abstract module* associated with $\mathrm{L}_{\hat{k}}$. Let

$$\hat{\mathcal{L}}_{norm} := \{\mathrm{L}_{\hat{k},norm} | \hat{k} \in \hat{I}\}$$

be the *truly normal abstract model of* $\mathcal{L}$. So when we build the two-level hierarchy, we build one abstract model $\hat{\mathcal{L}}$ and one truly normal abstract model $\hat{\mathcal{L}}_{norm}$, which will serve as the benchmark of normal behaviors. Now we describe how to use such a benchmark in hierarchical distributed diagnosis.

Multi-Resolution Diagnosis Procedure: (MRDP)

1. Given a symptom set $\mathcal{O} \in \mathbb{O}$, compute a symptom set of abstract model $\hat{\mathcal{L}}$, $\hat{O} := q(\mathcal{O}) \in \hat{\mathbb{O}}$, where q is defined in Figure 3.
2. Compute $\hat{\mathcal{M}} := \hat{\mathrm{P}}_o^{-1}(\hat{\mathcal{O}}) \hat{\sqcap} \hat{\mathcal{L}}$.
3. Compute $\mathrm{Sup}\Delta(\hat{\mathcal{M}}) = \{\mathrm{E}_{\hat{k}} | \hat{k} \in \hat{I}\}$.
4. For each $\hat{k} \in \hat{I}$, if $\mathrm{E}_{\hat{k}} \not\subseteq \mathrm{L}_{\hat{k},norm}$ then compute

$$(\forall i \in J_{\hat{k}})\, \mathrm{E}_i := \mathrm{P}_{J_{\hat{k}},i}((||_{j \in J_{\hat{k}}} \mathrm{E}_j) || \mathrm{E}_{\hat{k}}) \qquad \square$$

By Propositions 2.2 and 2.3 we know that

$$(\forall i \in I)\, \mathrm{E}_i \in \mathrm{Sup}\Delta(\mathrm{P}_o^{-1}(\mathcal{O}) \sqcap \mathcal{L})$$

The main feature of MRDP is that computation in a specific module, say $J_{\hat{k}}$, will be continued to a lower level on each member $i \in J_{\hat{k}}$ if some behavior not belonging to normal behavior has been detected in $\mathrm{L}_{\hat{k}}$ at the higher level, namely $\mathrm{E}_{\hat{k}} \not\subseteq \mathrm{L}_{\hat{k},norm}$. Since at step 3 the supremal global support in $\hat{\mathcal{L}}$ has been achieved, by Propositions 2.2 and 2.3 we know that if computation at some module $J_{\hat{k}}$ is halted at the high level because no abnormality is detected, then computation in other local modules will not be affected. On the other hand, we may raise a question about MRDP: is it possible that some local component in a module has become faulty, namely some string containing a fault event has been executed, but computation is halted at the corresponding high-level abstract module? The following result answers the question.

Definition 3.1 Let $\mathcal{L} = \{\mathrm{L}_i \subseteq \Sigma_i^* | i \in I\}$ be a distributed reference model. A set $S = \{s_i \in \mathrm{L}_i | i \in I\}$ is a *state* of $\mathcal{L}$ if $||_{i \in I}\{s_i\} \neq \varnothing$.

Let $R_i : \Sigma_i^* \to \mathrm{Pwr}(\Sigma_{if})$ be the local fault report map of component i such that $(\forall s \in \Sigma_i^*)\, R_i(s) := \{\sigma \in \Sigma_{if} | \sigma \in s\}$, where $\sigma \in s$ means that event σ appears at least once in s.

Proposition 3.1 Suppose $S = \{s_i | i \in I\}$ is a state of $\mathcal{L}$ and the symptom set is $\mathcal{O} = \{\mathrm{P}_{i,o}(s_i) | i \in I\}$. Then

$$(\forall \hat{k} \in \hat{I})(\forall i \in J_{\hat{k}})\, R_i(s_i) \cap \Sigma_{if} \neq \varnothing \Rightarrow \mathrm{E}_{\hat{k}} \not\subseteq \mathrm{L}_{\hat{k},norm}$$

where $\mathrm{E}_{\hat{k}}$ is obtained by MRDP on $\mathcal{O}$. $\square$

Proposition 3.1 says that if at state $S = \{s_j | j \in I\}$ some fault $\sigma \in \Sigma_{if}$ ($i \in I$) has occurred, namely $\sigma \in s_i \in S$, then $\mathrm{E}_{\hat{k}} \not\subseteq \mathrm{L}_{\hat{k},norm}$ where $\hat{k} \in \hat{I}$ and $i \in J_{\hat{k}}$. Therefore by MRDP, the computation will be carried out to each local component $j \in J_{\hat{k}}$, which includes component i. By Propositions 2.2 and 2.3 we get that the computational result is

$$\mathrm{E}_i \in \mathrm{Sup}\Delta(\mathrm{P}_o^{-1}(\mathcal{O}) \sqcap \mathcal{L})$$

Then by Prop. 2.1 we have that $s_i \in \mathrm{E}_i$. If we apply the local fault report map R_i on E_i, namely we report all faults in $\cup\{R_i(s)|s \in \mathrm{E}_i\}$ as fault candidates, then clearly the true fault $\sigma \in s_i$ will be reported at state S.

As an illustration, consider again the simple paper acquisition model (Figure 1). Besides the original abstract model (Figure 2), we construct another truly normal abstract model of the system. Figure 9 depicts normal behavior of each local component. We partition

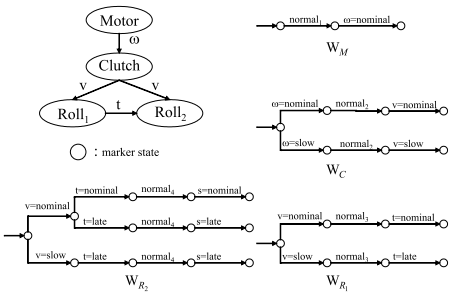

Figure 9. $\{\mathrm{W}_i|i \in I\}$ in SPAM

the system into two modules $J_1 := \{\mathrm{Motor}, \mathrm{Clutch}\}$ and $J_2 := \{\mathrm{Roll}_1, \mathrm{Roll}_2\}$. Based on $\{\mathrm{W}_i|i \in I\}$ we can compute that

$$\begin{aligned} \mathrm{L}_{1,norm} &= (v = \text{nominal}) \\ \mathrm{L}_{2,norm} &= (v = \text{nominal})(s = \text{nominal}) \end{aligned}$$

Case 1: Suppose at the bottom level, the observation is $s =$ late in Roll_2 and empty in other local components. Then by the abstract model (Figure 2) we have that

$$\begin{aligned} \mathrm{E}_1 &= \{v = \text{nominal}, v = \text{slow}\} \\ \mathrm{E}_2 &= \{(v = \text{nominal})(s = \text{late}), (v = \text{slow})(s = \text{late})\} \end{aligned}$$

Since $\mathrm{E}_1 \not\subseteq \mathrm{L}_{1,norm}$ and $\mathrm{E}_2 \not\subseteq \mathrm{L}_{2,norm}$, computation is carried to the bottom level in each local module.

Case 2: If the observation is $s =$ nominal in Roll_2 and empty in other local components. Then we get that

$$\begin{aligned} \mathrm{E}_1 &= \{v = \text{nominal}\} = \mathrm{L}_{1,norm} \\ \mathrm{E}_2 &= \{(v = \text{nominal})(s = \text{nominal})\} \subseteq \mathrm{L}_{2,norm} \end{aligned}$$

So by the proposed multi-resolution diagnosis procedure, no computation will be carried to the bottom level in any local module, and all components are declared *normal*.

4 Conclusions

In this paper we proposed a novel hierarchical distributed diagnosis approach. In this approach, at each single level of a pre-constructed hierarchy we solve a distributed diagnosis problem. Distributed diagnosis processes at different levels are linked together by information exchange between adjacent levels. Prop. 2.3 shows that after £rst achieving the supremal global support at the high level, we can apply concurrent computation to achieve the supremal global support at the low level. Furthermore, it turns out that the proposed hierarchical approach enables us to ignore any module that is confirmed to be *truly normal* at each intermediate level. Thus, computation to achieve the supremal global support at each level is only carried out in local modules that may contain faults. This selective computational procedure can further reduce time complexity of online diagnosis. Although the proposed hierarchical approach has to pay a price of extra memory to store high-level abstract models, our numerical computational results, which are not included in this paper due to limited space, indicate that the overall space complexity (which includes static memory to store models and dynamic memory to store intermediate computational results) is no worse (and in many cases much better) than a non-hierarchical approach. Finally, the simple procedure for constructing the high-level abstract models by using language-based natural projection and synchronous product is another advantage of our proposed hierarchical approach compared with other hierarchical approaches in the literature.

REFERENCES

[1] A. Aghasaryaiu, E. Fabre, A. Benveniste, R. Boubour, and C. Jard, 'A petri net approach to fault detection and diagnosis in distributed systems', in *Proc. 1997 IEEE Conference on Decision and Control (CDC'97)*, pp. 720–725, December 1997.

[2] K. Autio and R. Reiter, 'Structural abstraction in model-based diagnosis', in *Proc: ECAI98*, Munich, 1998.

[3] A. Benveniste, S. Haar, E. Fabre, and C. Jard, 'Distributed monitoring of concurrent and asynchronous systems', in *Proc. ATPN-Workshop on Discrete Event Systems Control*, Eindhoven, The Netherlands, June 2003.

[4] E. Duarte and T. Nanya, 'A hierarchical adaptive distributed system-level diagnosis algorithm', *IEEE Trans. on Computers*, **47**(1), 34–45, 1998.

[5] E. Fabre, A. Benveniste, and C. Jard, 'Distributed diagnosis for large discrete event dynamic systems', in *Proc. 15th IFAC World Congress*, Barcelona, Spain, July 2002.

[6] Igor Mozetic, 'Hierarchical model-based diagnosis', *International Journal of Man-Machine Studies*, **35**(3), 329–362, 1991.

[7] Gregory Provan, 'A model-based diagnosis framework for distributed systems', in *Proc. of International Workshop on Principles of Diagnosis (DX'02)*, Austria, May 2002.

[8] M. Sampath, S. Lafortune, and D. Teneketzis, 'Active diagnosis of discrete-event systems', *IEEE Transactions on Automatic Control*, **40**, 908–929, July 1998.

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

[10] R. Sengupta, 'Diagnosis and communication in distributed systems', in *Proc. WODES98*, pp. 144–151, 1998.

[11] D. Stroobandt and J. Van Campenhout, 'Hierarchical test generation with built-in fault diagnosis', in *Proc. 5th Asian Test Symp.*, pp. 22–28, November 1996.

[12] M.-S. Su, K. Thulasiraman, and A. Das, 'A multi-level adaptive distributed diagnosis algorithm for fault detection in a network of processors', in *Proc. 39th Annual Allerton Conf. on Communication, Control, and Computers*, 2001.

[13] R. Su and W.M. Wonham, 'Decentralized fault diagnosis for discrete-event systems', in *Proc. 2000 CISS*, pp. TP1:1–6, Princeton, New Jersey, March 2000.

[14] R. Su, W.M. Wonham, J. Kurien, and X. Koutsoukos, 'Distributed diagnosis for qualitative systems', in *Proc. WODES02*, pp. 169–174, Zaragoza, Spain, October 2002.

[15] W. M. Wonham, *Notes on Control of Discrete-Event Systems: ECE 1636F/1637S 2003-2004*, Systems Control Group, Dept. of ECE, University of Toronto. URL: www.control.utoronto.ca/DES, 2003.

[16] S. Hashtrudi Zad, R.H. Kwong, and W.M. Wonham, 'Fault diagnosis in discrete-event systems', in *Proc. 1998 IEEE Conference on Decision and Control (CDC'98)*, pp. 3769–3774, Tampa, Florida, USA, December 1998.