Distributed Diagnosis for Qualitative Systems

R. Su, W.M. Wonham
Dept. of Electrical & Computer Engineering
University of Toronto
Toronto, Ontario M5S 3G4, Canada
surong,wonham@control.utoronto.ca

J. Kurien, X. Koutsoukos
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94303, USA
jkurien,koutsouk@parc.xerox.com

Abstract

In this paper we propose a novel automaton-based architecture to build a diagnoser, based on which an efficient distributed diagnostic method consisting of local computation and communication is presented. The method proposed here is highly scalable and robust to partial failures of the overall diagnoser.

1. Introduction

The objective of diagnosis is to determine the state of a physical plant such as a printer or aircraft based upon current sensor readings from the plant and prior knowledge about the plant's structure and behavior. In order for the diagnosis to be useful for on-line control of the plant, diagnoses must be generated in a time-critical manner using limited computational resources. Most model-based diagnostic approaches for qualitative systems are centralized, e.g. [7] [6] [10] [2]. A centralized diagnoser stores a model of the entire plant, receives all sensor observations and executes a diagnosis algorithm. It has three main disadvantages: (1) high spatial complexity - its state set is usually the product of components' state sets; (2) weak robustness - any failure occurring somewhere inside the diagnoser could crash the whole diagnoser; (3) poor scalability - any structural change in the target system, e.g. adding new components, removing components or changing some input/output connections, might force us to build a completely new diagnoser.

To address these problems, attention has recently been directed to decentralized methods, e.g. [3] [8]. In [3] local diagnosers communicate with a coordination process which acts like a centralized unit, hence it still encounters the scalability and robustness problems. In [8] a decentralized diagnoser is built from a centralized diagnoser, which has high spatial complexity in the diagnoser construction stage. In [1] a ditributed method is presented which doesn't require coordination and each local diagnoser communicates directly with other diagnosers. But the structure of its local models makes the local diagnosis and communication extremely complicated, bringing temporal complexity to the fore. In this paper we propose a new automaton-based distributed diagnosis method. In this method each local component has its own local diagnoser, which is built based only on knowledge about this component. The stored size of the overall diagnoser is only the sum of state sizes of the local diagnosers, hence spatial complexity is kept under control. Each local diagnoser is connected with other local diagnosers based on the input/output relations among associated local components. Adding new components, taking components out of the system or changing the input/output relation among local components only affects the local diagnosers that are directly associated with the altered components. So high scalability of the overall diagnoser can be achieved. The local diagnosis in each local diagnoser is based mainly on its local observation, and communication is only used for refinement purposes. So if some local diagnoser or communication channel failed, other normal local diagnosers could still produce local diagnoses and refine them based on communication via undamaged communication channels. Thus the overall diagnoser is robust to partial diagnoser failure.

This paper is organized as follows. In Section 2 we provide technical details. In Section 3 we demonstrate its efficiency on a real-world example, and draw conclusions in Section 4.

2 Distributed diagnosis - technical details

2.1 Basic concepts and assumptions

A qualitative (non-numerical) component model encoded as a set of logical constraints has long been used in diagnosis, e.g. [2] [5]. In this spirit we model the transition

structure of each local diagnoser. Let $I = \{1, \cdots, n\} \subseteq \mathbb{N}$ be an index set.

Definition 2.1 A *qualitative diagnostic system* $\mathcal{D}$ is a set of finite automata

$$\mathcal{D} = \{\mathbb{G}_i = (X_i, \Sigma_i, \xi_i, \Sigma_{F,i}, x_0) | i \in I\}$$

where X_i is the *state set*, Σ_i the *event set*, $\Sigma_{F,i} \subseteq \Sigma_i$ the *fault event set*, x_0 the *initial state*, $\xi_i : X_i \times \Sigma_i \to X_i$ the *(partial) transition function*. $\mathbb{G}_i$ is a *local qualitative model* of the diagnostic system.

Figure 1 depicts a simple paper-path system (SPPS) and its qualitative diagnostic system. In this system the motor de-

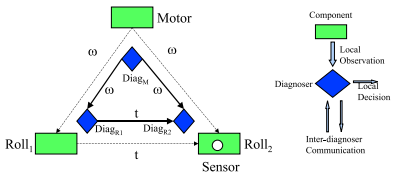

Figure 1. A Simple Paper-Path System

livers drive ω to both rolls. Thus the models describing the Motor and the two rolls all share the variable ω. Roll_1 pulls a sheet of paper into the paper path and pushes it onto Roll_2. Roll_2 then grasps the sheet and pushes until it is out of the paper path. The variable t shared by Roll_1 and Roll_2 represents the time when Roll_1 pushes a piece of paper onto Roll_2. The sensor will detect the arrival of the leading edge of a piece of paper at the exit end of the paper path and set the time of that occurrence within the model for Roll_2. In (SPPS), Motor could be *normal*, *slow* or *breakdown* and the corresponding velocity output ω could be *nominal*, *slow* or *zero*. Hence we can develop a local qualitative model for Motor as shown in Figure 2(a). In Figure 2(b) Roll_1 receives drive ω from Motor and sends an output t (the leading edge arrival time at the entrance of Roll_2) to Roll_2. Roll_1 could be *normal* or *high-friction*. Considering the effect of ω, its qualitative output t could be *nominal*, *late* or *infinite*, where the latter means that t exceeds a pre-defined maximum waiting time. In Figure 2(c) Roll_2 has a similar transition model except that it contains sensor information *leAtS*, a variable denoting "leading edge arrival time at sensor S". This can take three qualitative values: *nominal*, *late* and *infinite*, with an interpretation similar to that in Roll_1. Each assignment is explicitly represented by an event. So we have three observable events associated with sensor readings: *leAtS=nominal*, *leAtS=late* and *leAtS=infinite*. All other events are unobservable.

For each $i \in I$ the local (partial) transition function ξ_i can be extended as $\xi_i : X_i \times \Sigma_i^* \to X_i$ (see [9]). Let

$$\mathrm{L}(\mathbb{G}_i) := \{s \in \Sigma_i^* | \xi_i(x_0, s)!\}$$

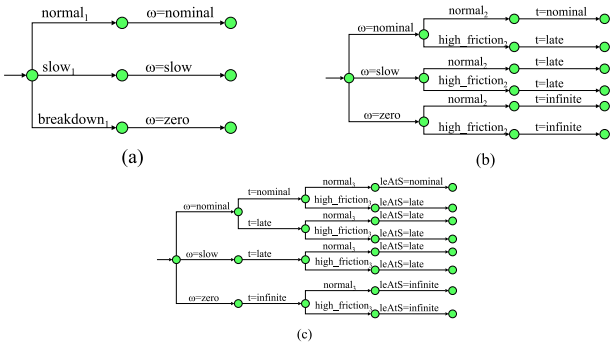

Figure 2. Motor (a), Roll_1 (b) and Roll_2 (c)

Definition 2.2 The qualitative diagnostic system $\mathcal{D}$ is *finite* if $(\forall i \in I)\, |\mathrm{L}(\mathbb{G}_i)| < \infty$.

Assumption 1: $\mathcal{D}$ is finite.

For each $i \in I$ let $\Sigma_{obs,i} \subseteq \Sigma_i$ be the set of observable events. Bring in the natural projections $\mathrm{P}_{ij} : \Sigma_i^* \to \Sigma_j^*$, $\mathrm{P}_{i,obs} : \Sigma_i^* \to \Sigma_{obs,i}^*$, $\mathrm{P}_{i,F} : \Sigma_i^* \to \Sigma_{F,i}^*$. ϵ is the empty string and $\sigma \in s$ means the event σ appears in the string s.

A system usually repeats a pre-defined function, e.g. a printer repetitively does single-side printing until the user switches the work mode to double-side printing or turns off the machine. The duration of each instantiation of the pre-defined function is a *work cycle*. In the rest of the paper, all definitions and procedures are for one work cycle. Call

$$\mathbb{S} := \left\{ \{s_i \in \mathrm{L}(\mathbb{G}_i) | i \in I\} \middle| (\forall i, j \in I)\, \mathrm{P}_{ij}(s_i) = \mathrm{P}_{ji}(s_j) \right\}$$

the *state set* of $\mathcal{D}$ and $S = \{s_i \in \mathrm{L}(\mathbb{G}_i) | i \in I\} \in \mathbb{S}$ a *state* of $\mathcal{D}$. Given $S, S' \in \mathbb{S}$, $S \leqslant S' \iff \{s_i \leqslant s_i' | i \in I\}$ and $S < S' \iff S \leqslant S' \wedge S \neq S'$.

Definition 2.3 A list $T_k = (S_1, \cdots, S_k)$ is a *k-trajectory* of $\mathcal{D}$ if $(\forall i, j : 1 \leq i, j \leq k) i < j \Rightarrow S_i < S_j$.

In SPPS the state $S_0 = \{s_M = \epsilon, s_{R_1} = \epsilon, s_{R_2} = \epsilon\}$ means that no component in SPPS has as yet fired any string. The state $S_1 = \{s_M = (normal_1), s_{R_1} = \epsilon, s_{R_2} = \epsilon\}$ means Motor fires a string $normal_1$ and the other two are still idle. The state $S_2 = \{s_M = (normal_1)(\omega = nominal), s_{R_1} = (\omega = nominal), s_{R_2} = \epsilon\}$ means that Motor fires a string $(normal_1)(\omega = nominal)$, Roll_1 fires a string $\omega = nominal$ and Roll_2 is still idle. By Def. 2.3 (S_0, S_1, S_2) forms a 3-trajectory of SPPS. We write $S \in T$ to mean that the state S appears in T. In a work cycle, at each time instant there is a trajectory describing the transition behavior of $\mathcal{D}$ between the starting time instant of the work cycle and the current time instant. Let $\mathcal{T}_\mathcal{D}$ be the set

of all possible trajectories of $\mathcal{D}$ at the end of the work cycle. Next we describe how to use $\mathcal{D}$ to fulfil diagnostic tasks.

2.2 Local computation and communication

One of the main objectives of fault diagnosis is to appraise the fault status of the associated component. An appraisal must be based on an estimate of the transition behavior of this component just before the time when a decision is made. First we give a formal description of "estimate". Let a language $\mathrm{L} := \{s_i \in \Sigma^*_{obs,i} | i \in I\}$ represent our observation about $\mathcal{D}$. $s_i \neq \epsilon$ means there is observation in $\mathbb{G}_i$; otherwise there is no local observation in $\mathbb{G}_i$.

Definition 2.4 Let $\mathcal{D}$ be the diagnostic system. Given a language $\mathrm{L} = \{s_i \in \Sigma^*_{obs,i} | i \in I\}$, and collections of languages $\mathcal{E} = \{\mathrm{E}_i \subseteq \mathrm{L}(\mathbb{G}_i) | i \in I\}$ and $\tilde{\mathcal{E}} = \{\tilde{\mathrm{E}}_i \subseteq \mathrm{L}(\mathbb{G}_i) | i \in I\}$, $\tilde{\mathcal{E}}$ is an *estimate about $\mathcal{D}$ with respect to $\mathcal{E}$ and L* if

1. $(\forall i \in I)\tilde{\mathrm{E}}_i \subseteq \{ss' \in \mathrm{L}(\mathbb{G}_i) | s \in \mathrm{E}_i \wedge \mathrm{P}_{i,obs}(s') = s_i\}$
2. $(\forall i, j \in I)\mathrm{P}_{ij}(\tilde{\mathrm{E}}_i) = \mathrm{P}_{ji}(\tilde{\mathrm{E}}_j)$

L is called a *constraint* and $\mathcal{E}$ is called *prior knowledge*.

By Def. 2.4 we see that for each $i \in I$: (1) $\tilde{\mathrm{E}}_i$ is "evolved" from prior knowledge E_i and extended with new information from the constraint L - the set of newly obtained observations (Condition 1); (2) All local estimates are consistent (Condition 2).

Let $\Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L})$ be the set of all estimates about $\mathcal{D}$ with respect to $\mathcal{E}$ and L. $\tilde{\mathcal{E}} = \{\tilde{\mathrm{E}}_i | i \in I\} \in \Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L})$ is the *supremal estimate in* $\Gamma(\mathcal{D}, \mathcal{E}, L)$ (written $\mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L})$) if

$$(\forall \tilde{\mathcal{E}}' := \{\tilde{\mathrm{E}}'_i | i \in I\} \in \Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L}))(\forall i \in I)\, \tilde{\mathrm{E}}'_i \subseteq \tilde{\mathrm{E}}_i$$

Note that if $\mathcal{E}_1 = \{\mathrm{E}^1_i | i \in I\}$, $\mathcal{E}_2 = \{\mathrm{E}^2_i | i \in I\}$ are in $\Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L})$, so is $\mathcal{E}_3 = \{\mathrm{E}^1_i \cup \mathrm{E}^2_i | i \in I\}$. Hence the supremal estimate exists. In SPPS suppose

$$\mathcal{E} = \{\mathrm{E}_M = \{\epsilon\}, \mathrm{E}_{R_1} = \{\epsilon\}, \mathrm{E}_{R_2} = \{\epsilon\}\}$$

and $\mathrm{L} = \{s_M = \epsilon, s_{R_1} = \epsilon, s_{R_2} = (leAtS = late)\}$. Then the result $\mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L})$ is displayed in Figure 3, where $\tilde{\mathrm{E}}_{Motor} := \{s_{11}, s_{12}\}$, $\tilde{\mathrm{E}}_{Roll_1} := \{s_{21}, s_{22}, s_{23}, s_{24}\}$ and $\tilde{\mathrm{E}}_{Roll_2} := \{s_{31}, s_{32}, s_{33}, s_{34}, s_{35}\}$.

Computation procedure for the supremal estimate $\tilde{\mathcal{E}}$:
(1) Initialization: for each $i \in I$,

$$\tilde{\mathrm{E}}_i := \{ss' \in \mathrm{L}(\mathbb{G}_i) | s \in \mathrm{E}_i \wedge \mathrm{P}_{i,obs}(s') = s_i\}$$

(2) Communication: To each $\mathbb{G}_i$ $(i \in I)$ continuously apply the following operations until **Termination-Condition** is satisfied.

- send message: for each $j \in I$ with $\Sigma_i \cap \Sigma_j \neq \varnothing$, send $\mathrm{P}_{ij}(\tilde{\mathrm{E}}_i)$ to $\mathbb{G}_j$ as a message.

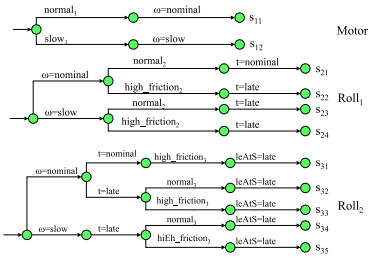

Figure 3. Supremal Estimate in SPPS

- local update: if a message $\mathrm{P}_{ji}(\tilde{\mathrm{E}}_j)$ from $\mathbb{G}_j$ is received then make the new assignment

$$\tilde{\mathrm{E}}_i := \mathrm{P}^{-1}_{ij}(\mathrm{P}_{ji}(\tilde{\mathrm{E}}_j)) \cap \tilde{\mathrm{E}}_i \tag{1}$$

The **Termination-Condition** (TC) is

$$(\forall i \in I)(\forall j \in I)\, \tilde{\mathrm{E}}_i = \mathrm{P}^{-1}_{ij}(\mathrm{P}_{ji}(\tilde{\mathrm{E}}_j)) \cap \tilde{\mathrm{E}}_i$$

Due to limited space, in this paper we cannot describe details of our communication protocol which is used to verify TC. The general idea is that each communication process is divided into several rounds, in each of which all communication occurs in one direction, either "upstream" or "downstream". When each local diagnoser sends out messages to other diagnosers, it puts a tag in the message indicating whether it has reached a fixed point or not. Other local diagnosers will use this tag to determine when to terminate the communication. Interested readers can contact the authors for more details about the communication protocols. Figure 4 demonstrates one round of communication in SPPS. In

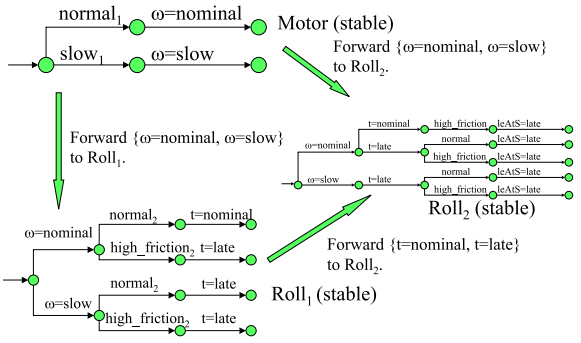

Figure 4. Communication in SPPS

the local update operation suppose $\tilde{\mathrm{E}}^1_i$ represents $\tilde{\mathrm{E}}_i$ on the right-hand side of (1) and $\tilde{\mathrm{E}}^2_i$ represents $\tilde{\mathrm{E}}_i$ on the left-hand side. If $\tilde{\mathrm{E}}^1_i \neq \tilde{\mathrm{E}}^2_i$ then the local update operation is called an *effective local update*.

Proposition 2.1 TC is satisfied in a finite number of effective local updates.

Proposition 2.2 When TC is satisfied, we have

$$\tilde{\mathcal{E}} = \{\tilde{\mathrm{E}}_i | i \in I\} = \mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}, \mathrm{L})$$

For each $i \in I$ let

$$R_i : 2^{\mathrm{L}(\mathbb{G}_i)} \to \mathrm{P}_{i,F}(\mathrm{L}(\mathbb{G}_i)) : W \mapsto R_i(W) := \mathrm{P}_{i,F}(W)$$

be the *local fault report map* on $\mathbb{G}_i$. Each $\mu \in R_i(W)$ is a *fault candidate*. In the following proposition, for each state $S_j \in \mathbb{S}$ we write s_i^j to mean that $s_i^j \in \mathrm{L}(\mathbb{G}_i) \cap S_j$.

Proposition 2.3 Let $i \in I$, fault event $\sigma_f \in \Sigma_{F,i}$, and trajectory $T \in \mathcal{T}_{\mathcal{D}}$. Suppose there exists a state $S_j \in T$ with $\sigma_f \in s_i^j$. For any state $S_r \in T$ with $S_j \leqslant S_r$, let the constraint be $\mathrm{L} := \{\mathrm{P}_{k,obs}(s_k^r) | k \in I\} \neq \{\epsilon\}$ and the prior knowledge be $\mathcal{E}_0 := \{\{\epsilon\}\}$. Then we have

$$(\exists \mathrm{E}_i \subseteq \mathrm{L}(\mathbb{G}_i))\, \mathrm{E}_i \in \mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}_0, \mathrm{L}) \wedge (\exists \mu \in R_i(\mathrm{E}_i))\, \sigma_f \in \mu$$

Prop. 2.3 tells us that in any trajectory, if a fault "occurs" then it can be "reported" at any later state in the trajectory which contains at least one new observation. Each fault report $R_i(\mathrm{E}_i)$ contains several fault candidates. It is quite possible that not all fault candidates contain the occurred fault. Hence in many cases we may not be able to determine the occurrence of the occurred fault without ambiguity. For example in SPPS when the computation and communication process terminates, Motor will report $\{normal_1, slow_1\}$, which means the fault status of the Motor is either $normal$ or $slow$. Both of them are consistent with the observation. Roll_1 will report $\{normal_2, high\_friction_2\}$ and Roll_2 will report $\{normal_3, high\_friction_3\}$. None of them can be used to determine the true fault uniquely. Such ambiguity is caused by the lack of global view in local diagnosis (and also the lack of sufficient local observation). In some sense local diagnosis is better served as a preprocessing procedure, which rules out a lot of impossible solutions before more advanced but quite resource-consuming diagnosis techniques are applied.

In practical situations observations are usually obtained cumulatively. For example a local component could receive an observation at time instant t_1, then receive another one at $t_2 > t_1$, and so on. The local component buffers these local observations according to the order in which they were received. This process is called *observation accumulation*. There are two ways to use these observations: (1) in one-stage diagnosis all previously used observations have to be remembered (Prop. 2.3 describes the diagnostic capability of this type of method); (2) in multi-stage diagnosis all previously used observations are erased from memory and only newly obtained observations are used to update estimates. In many practical situations a multi-stage diagnosis method is preferable to a one-stage method. So now we describe it in detail. Let $\mathcal{L} := \left\{\{s_i \in \Sigma_{obs,i}^* | i \in I\}\right\}$ be the set of all constraints and $\star : \mathcal{L} \times \mathcal{L} \to \mathcal{L}$ be the observation catenation map such that for each pair of constraints $\mathrm{L}_1 = \{s_{i,1} | i \in I\}$ and $\mathrm{L}_2 = \{s_{i,2} | i \in I\}$, $\mathrm{L}_1 \star \mathrm{L}_2 := \{s_{i,1} s_{i,2} | i \in I\}$.

Definition 2.5 Given constraints $\mathrm{L}_1 = \{s_{i,1} | i \in I\} \in \mathcal{L}$, $\mathrm{L}_2 = \{s_{i,2} | i \in I\} \in \mathcal{L}$, L_2 is *reasonably related to* L_1 if for each $\mathcal{E} = \{\mathrm{E}_i | i \in I\} \in \Gamma(\mathcal{D}, \{\{\epsilon\}\}, \mathrm{L}_1 \star \mathrm{L}_2)$, there exists

$$\mathcal{E}' = \{\mathrm{E}_i' \subseteq \mathrm{L}(\mathbb{G}_i) | (\forall s_i \in \mathrm{E}_i)(\exists s_i' \in \mathrm{E}_i')\, s_i' \leqslant s_i \wedge i \in I\}$$

such that $\mathcal{E}' \in \Gamma(\mathcal{D}, \{\{\epsilon\}\}, \mathrm{L}_1)$. A list of constraints $[\mathrm{L}_1, \cdots, \mathrm{L}_m]$ is an *observation chain* if

$$m \geq 2 \Rightarrow (\forall j : 2 \leq j \leq m)\ \mathrm{L}_j \text{ is reasonably related to } \mathrm{L}_1 \star \cdots \star \mathrm{L}_{j-1}.$$

In SPPS suppose we can also measure t, namely $t = nominal$, $t = late$ and $t = infinity$ are observable events in both Roll_1 and Roll_2. Then we can check that given two constraints

$$\begin{aligned} \mathrm{L}_1 &= \{s_M = \epsilon, s_{R_1} = \epsilon, s_{R_2} = (t = late)\} \\ \mathrm{L}_2 &= \{s_M = \epsilon, s_{R_1} = (t = late), s_{R_2} = (leAtS = late)\} \end{aligned}$$

L_2 is not reasonably related to L_1, because for each $\mathcal{E} \in \Gamma(\mathcal{D}, \{\{\epsilon\}\}, \mathrm{L}_1 \star \mathrm{L}_2)$, there doesn't exist the $\mathcal{E}'$ required in Def. 2.5. An intuitive reason is that, in L_1 $s_{R_1} = \epsilon$ means that the paper in Roll_1 hasn't been sent to Roll_2 yet. But on the other hand $s_{R_2} = (t = late)$ means Roll_2 has received a paper from Roll_1. Hence in the constraint L_1, the two observations $s_{R_1} = \epsilon$ and $s_{R_2} = (t = late)$ are not consistent. If let $\mathrm{L}_1 = \{s_M = \epsilon, s_{R_1} = (t = late), s_{R_2} = (t = late)\}$, then L_2 is reasonably related to L_1. The following proposition describes the relationship between one-stage diagnosis and multi-stage diagnosis.

Proposition 2.4 Given constraints $\mathrm{L}_1, \mathrm{L}_2 \in \mathcal{L}$ and prior knowledge $\mathcal{E}_0 := \{\{\epsilon\}\}$, if L_2 is reasonably related to L_1 then, $\mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}_0, \mathrm{L}_1 \star \mathrm{L}_2) = \mathrm{Sup}\Gamma(\mathcal{D}, \mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}_0, \mathrm{L}_1), \mathrm{L}_2)$.

A *local diagnoser* $\mathbb{D}_i$ consists of the local qualitative model $\mathbb{G}_i$, an estimate $\mathrm{E}_i \subseteq \mathrm{L}(\mathbb{G}_i)$, a local fault report map R_i, and a communication protocol C_i which is not defined in this paper, namely $\mathbb{D}_i = (\mathbb{G}_i, \mathrm{E}_i, R_i, \mathrm{C}_i)$. Suppose the duration of each work cycle is $(0, t] \subseteq \Re^+$, where $\Re^+$ denotes the non-negative real numbers. The work cycle is divided into m time subintervals $(t_0, t_1], (t_1, t_2], \cdots, (t_{m-1}, t_m]$ with $t_0 = 0$ and $t_m = t$. There is a global clock available to each local diagnoser for synchronizing their communication. In each time subinterval $(t_{j-1}, t_j]$ $(j = 1, \cdots, m)$, each local diagnoser $\mathbb{D}_i$ $(i \in I)$ buffers observable events, which are received in $(t_{j-1}, t_j]$. Suppose at t_j the observable events received by $\mathbb{D}_i$ forms a string $s_{i,t_j} \in \Sigma_{obs,i}^*$,

where the order of events in s_{i,t_j} is the temporal order of their being received. If at the current time subinterval there is no observation received then $s_{i,t_j} = \epsilon_i$. Let $\mathrm{L}_{t_j} := \{s_{i,t_j} \in \Sigma^*_{obs,i} | i \in I\}$.

Assumption 2: $[\mathrm{L}_{t_1}, \cdots, \mathrm{L}_{t_m}]$ is an observation chain.

Distributed Diagnosis Procedure:
1. **Initialization**: At t_0 set the initial estimate of each $\mathbb{D}_i$ ($i \in I$) to $\mathrm{E}_i^0 := \{\epsilon\}$.

2. **Diagnosis**: In each $(t_{j-1}, t_j]$ ($j = 1, \cdots, m$), if $\mathrm{L}_{t_j} = \{\epsilon\}$ then for each $i \in I$, set $\mathrm{E}_i^j := \mathrm{E}_i^{j-1}$ and go to the next time subinterval; otherwise do the following operations (2.1-2.2) at t_j.

2.1 Construct prior knowledge $\mathcal{E}_{t_{j-1}} := \{\mathrm{E}_i^{j-1} | i \in I\}$.

2.2 Compute $\mathcal{E}_{t_j} = \mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}_{t_{j-1}}, \mathrm{L}_{t_j})$. At this stage each local diagnoser $\mathbb{D}_i$ ($i \in I$) needs to use its communication protocol to support communication with other local diagnosers. $R_i(\mathrm{E}_i^j)$ is the local diagnosis in $\mathbb{D}_i$ at t_j.

The above multi-stage distributed diagnosis procedure essentially calculates

$$\begin{aligned} \mathcal{E}_{t_0} &:= \{\{\epsilon\}\} \\ (\forall j : 1 \le j \le m)\, \mathcal{E}_{t_j} &:= \mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}_{t_{j-1}}, \mathrm{L}_{t_j}) \end{aligned}$$

By Prop. 2.4 it is equivalent to a one-stage diagnosis procedure $(\forall j : 1 \le j \le m)\, \mathcal{E}_{t_j} = \mathrm{Sup}\Gamma(\mathcal{D}, \mathcal{E}_0, \mathrm{L}_{t_1} \star \cdots \star \mathrm{L}_{t_j})$. In reality if the trajectory of $\mathcal{D}$ at the end of the work cycle t_m is $\mathrm{T}_k = \{S_1, \cdots, S_k\} \in \mathcal{T}_{\mathcal{D}}$, then

$$\mathrm{L}_{t_1} \star \cdots \star \mathrm{L}_{t_m} = \{\mathrm{P}_{i,obs}(s_i^k) | i \in I\}$$

Hence by Prop. 2.3, if any fault occurs during the trajectory T_k then the distributed diagnosis procedure will "report" it in the same work cycle. Finally the relationship between local diagnosis and global diagnosis is described as follows.

Definition 2.6 Given a supremal estimate $\mathcal{E} = \{\mathrm{E}_i | i \in I\}$ of $\mathcal{D}$, a set of strings $Q = \{s_i \in \mathrm{E}_i \,|\, i \in I\}$ is a *global estimate* of $\mathcal{D}$ if $(\forall s_i, s_j \in Q)\, \mathrm{P}_{ij}(s_i) = \mathrm{P}_{ji}(s_j)$. The associated fault candidate set of Q, $\mathcal{F} = \{R_i(\{s_i\}) \,|\, i \in I\}$ is a *global diagnosis* of $\mathcal{D}$.

In words, a set of fault candidates is a global diagnosis if there exists a set of consistent strings, each of which is from a unique local diagnoser, such that each fault candidate is contained in exactly one string. A global diagnosis is a global perspective of the fault status of the whole system. Computing the set of all global estimates is NP-hard.

3 Test results

Figure 5 is a schematic depiction of a paper path model for the Xerox DC265ST printer, consisting of 24 components. There are 5 sensors labelled $\mathrm{S}_1, \cdots, \mathrm{S}_5$. Each sensor is used to record the leading-edge arrival time and the trailing-edge arrival time of each piece of paper. There are three motors in the system that transfer drives to rolls via gears, belts and clutches. Each box represents the model of a single paper path component's local behavior, including possible failures. Arrows indicate interaction between components.

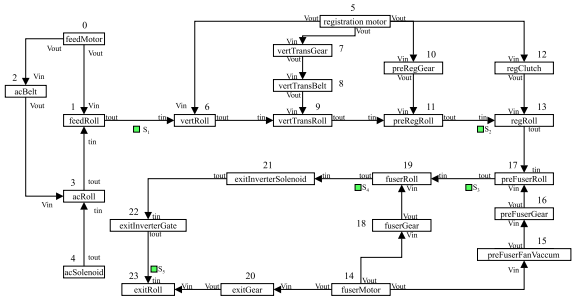

Figure 5. DC265ST Printer Model

Table 1 gives some possible fault scenarios in the paper path model for the Xerox DC265ST printer (displayed in Figure 5) and the corresponding local diagnoses, where "$\mathbf{N} : normal$", "$\mathbf{W} : wornout$" and "$\mathbf{N}, \mathbf{W} : 14$" means component 14 is either $normal$ or $wornout$. The CPU of the computer for the following computation was a PIII 750MHz and the computation time for the local diagnosis in each row was less than 10^{-5} sec.

Table 1

Sensor Readings		Local Diagnoses
Order	Detailed Readings	
1	leAtS1=normal	$\mathbf{N}$: 0,1,2,3,4
2	teAtS1=normal	$\mathbf{N}$: 5,6
3	leAtS2=normal	$\mathbf{N}$: 5,6,7,8,9,10
4	teAtS2=normal	$\mathbf{N}$: 5,12,13
5	leAtS3=late	$\mathbf{N}$: 5,12,13 $\mathbf{N}, \mathbf{W}$: 14,15,16,17
6	teAtS3=late	$\mathbf{N}, \mathbf{W}$: 14,18,19
7	leAtS4=late	$\mathbf{N}, \mathbf{W}$: 14,18,19

Table 1 can be interpreted as follows: after the first observation $leAtS1 = normal$ is obtained from sensor S_1, local diagnosers $\mathbb{G}_0, \cdots, \mathbb{G}_4$ report their local diagnoses as *normal*. After the second observation $teAtS1 = normal$ is obtained, local diagnosers $\mathbb{G}_5$ and $\mathbb{G}_6$ report their local diagnoses as *normal* and so on. From Table 1 we can conclude that based on seven sensor readings, components

$0, \cdots, 13$ are $normal$ and components $14, \cdots, 19$ are either $normal$ or $wornout$.

Table 2 shows the results of comparison between our method and L2, a centralized method developed by NASA Ames Research Center [4]. Although both methods produce global diagnoses, our method will find all consistent fault candidates, each of which will cover the fault status of all participating local components. L2 is designed to find all minimum-fault candidates that can explain the current observation discrepancy. Hence in the same fault scenario L2 will have a smaller search space than does our method. It is still a good benchmark to evaluate the performance of our method compared with a centralized method.

Table 2

No.	Sensor Readings	NPLD	NGD	CT (s)	
				Distr.	Centr.
1	LS1=l, TS1=l	9	21/6	0	0.02
2	LS1=l, TS1=l, LS2=l, TS2=l	14	49/12	0	0.18
3	LS1=l, TS1=l, LS2=l, TS2=l, LS3=l, TS3=l	20	343/84	0.05	13.28
4	LS1=l, TS1=l, LS2=l, TS2=l, LS3=l, TS3=l, LS4=l, TS4=l	21	343/84	0.06	19.63
5	LS1=l, TS1=l, LS2=l, TS2=l, LS3=l, TS3=l, LS4=l, TS4=l, LS5=l, TS5=l	24	637/108	0.22	27.08

In Table 2 "LS1=l" means the leading-edge arrival time at sensor S_1 is $late$, "TS1=l" means the trailing-edge arrives $late$ at sensor S_1, "NPLD" is an abbreviation for the Number of Participating Local Diagnosers, "CT" is the Computation Time and "Distr.", "Centr." mean $distributed$ and $centralized$ respectively. "NGD" is the Number of Global Diagnoses. In the column of NGD the number 21/6 means that the distributed method generates 21 global diagnoses and L2 generates 6 global diagnoses. The difference results because L2 only searches minimum-fault global diagnoses but our method searches all global diagnoses. We have checked that all fault candidates found by L2 are contained in the set of fault candidates generated by our method. Table 2 indicates that as far as the global diagnosis is concerned, our proposed automaton-based method is much more efficient for this problem than a centralized method.

4 Conclusion

In this paper we present a methodology for on-line distributed fault diagnosis. We use automata to model local diagnosers and use an on-line efficient local computation and communication approach to produce the local diagnosis in each local diagnoser. Although the local space complexity is increased due to the pre-compilation of the local diagnosers, the total spatial complexity of the whole system is still manageable due to the purely distributed architecture. The novel local computation and communication approach proposed is highly scalable and also robust to failures of local diagnosers. Hence it is suitable for fault diagnosis for large-scale systems. There are still several problems to be further investigated, e.g. how to incorporate probability into a transition model, how to efficiently determine the fault probability distribution when the communication terminates, how to determine the most likely fault in an efficient way, and how to find an optimal distributed structure which has the best tradeoff between spatial complexity and temporal complexity. These problems will be addressed in future papers.

Acknowledgement: The original formulation and implementation of the approach were developed while the first author was an intern at the Palo Alto Research Center. Financial support from the Defense Advanced Research Projects Agency (DARPA) under contract F33615-99-C3611, the Natural Sciences and Engineering Research Council (NSERC), under RG7399, and an Operating Grant from Honeywell, is gratefully acknowledged. L2 was made publicly available by NASA Ames Research Center.

References

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

[2] J. de Kleer and B. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[3] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic System: Theory and Applications*, 10(1/2):33–86, January 2000.

[4] J. Kurien and P. P. Nayak. Back to the future with consistency based trajectory tracking. In *Proceedings of AAAI 2000*, 2000.

[5] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[6] M. Sampath, S. Lafortune, and D. Teneketzis. Active diagnosis of discrete-event systems. In *Proc. 36th Conf. Decision Contr.*, pages 2976–2983, San Diego, USA, December 1997.

[7] 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.

[8] R. Su and W. Wonham. Decentralized fault diagnosis for discrete-event systems. In *Proc. 2000 CISS*, pages TP1:1–6, Princeton, New Jersey, March 2000.

[9] W. M. Wonham. *Notes on Control of Discrete-Event Systems: ECE 1636F/1637S 2002-2003*. Systems Control Group, Dept. of ECE, University of Toronto. URL: www.control.utoronto.ca/people/profs/wonham/wonham.html.

[10] S. H. Zad, R. Kwong, and W. Wonham. Fault diagnosis in discrete-event systems. In *Proc. 37th Conf. Decision Contr.*, pages 3769–3774, Tampa, Florida, USA, December 1998.