Distributed Diagnosability Properties of Discrete Event Systems[1]

Gregory Provan
Rockwell Scientific Company,
1049 Camino Dos Rios, Thousand Oaks, CA 91360
gprovan@rwsc.com

Abstract

We present a distributed diagnostics and control architecture for embedded distributed diagnostics. Using this architecture, we (1) propose a distributed diagnostics framework for discrete-event systems based on causal networks; (2) prove system diagnosability results based on component diagnosability; and (3) discuss the tradeoffs involving system diagnosability and communication requirements.

1 Introduction

The task of diagnosing Discrete Event Systems (DESs)[1] has received much attention in the literature. The two dominant formalisms for such analyses are the construction of a diagnoser from a Finite State Machine (FSM) system model [12, 13, 15], and the use of Petri net models [4]. Given the prevalence of many applications best modeled and diagnosed as distributed systems, attention has now shifted towards distributed approaches, such as [3, 5, 14].

The article extends the causal network [2] approach to diagnosing DESs [8, 9] to a decentralized approach. Causal networks represent a promising new representation for DES control and diagnostics, and offer some advantages over other approaches. In comparison to distributed approaches based on constructing a diagnoser from an FSM [3, 14], our approach provides two major benefits: (1) more efficient diagnostic inference, since the complexity of constructing the diagnoser and testing the diagnosability is exponential in the number of states and doubly exponential in the number of failure types [12], whereas the causal network approach has complexity exponential in the graph-width of the underlying graph [2], but not necessarily exponential in either the number of states or of failure types; (2) simpler and more compact modeling, especially for distributed models.

In this article we address the conditions necessary to ensure equivalence between diagnosability properties of centralized and decentralized approaches (based purely on local sensor data). Typically, distributed diagnosabilty is incomplete with respect to centralized diagnosability, i.e., some sub-systems are not diagnosable given purely local sensor data, but require data from other sub-systems. When discrepancies exist between distributed and centralized diagnosability, approaches like message-passing must be employed to enable complete sub-system diagnosabilty within a distributed model. Our contributions are as follows: (1) we propose a distributed diagnostics framework for discrete-event systems based on causal networks; (2) we prove system diagnosability results based on component diagnosability; and (3) we discuss different approaches for guaranteeing diagnosability for distributed systems, e.g., message-passing, local model strengthening and local observability, as well as the tradeoffs of each approach.

This document is organized as follows. Section 2 compares and contrasts our approach with related work in the literature. Section 3 introduces our modeling formalism, and specifies our notion of centralized and distributed models. Section 4 describes how we diagnose distributed models, and the diagnosability properties entailed. Section 5 identifies issues with diagnostic incompleteness of distributed models, and outlines methods of circumventing such problems based on message-passing and adding equations to local models. We summarize our conclusions in Section 6.

2 Related Work

In past work we have used the causal network formalism for modeling, diagnosing and reconfiguring discrete-event systems [8]. We model the underlying system (plant) behavior as a causal network (plant model), which specifies transitions among a set of discrete variables. We have described in [8] how we can use this approach for component-based systems modeling,

[1]Research supported in part by The Office of Naval Research under contract number N00014-98-3-0012.

where large systems are composed from instances taken from a library of component classes. We extended the causal network model-based diagnostics approach [2] to incorporate simulation, control/reconfiguration and diagnosis using a single underlying representation [9].

Causal networks and FSMs have many similarities, but also some differences in terms of representational capabilities and semantics; see [10] for a more complete explanation. For example, they represent different model properties in an explicit fashion. An FSM represents states and state transitions explicitly. However, the system models and the diagnoser representations for system diagnosis can be quite large, and it is difficult to explicitly encode the physical topology of component connectivity. In contrast, the causal network has a compact representation, and can encode system structure explicitly. This approach has an implicit notion of transition, and uses logical equations, which specify constraints on allowable transitions rather than explicity enumerating all transitions. It is this implicit representation of states and transitions that, to some extent, is responsible for the compactness of the represention.

The causal network approach provides a clear mechanism for mapping into models the physical system schematics, and the hierarchical decomposition of a system into sub-systems. Such a hierarchical decomposition is natural within engineering systems, which are sub-divided into topologically distinct sub-systems. In this article, we define a *block* to correspond to a subsystem; as a consequence of this definition, the components local to a block, such as sensors, actuators, faultable components, are clearly defined. Given such an engineering-based system decomposition, one important issue is the diagnosability of sub-systems based purely on local sensor data, versus system diagnosability given global sensor data (such as is present given a centralized model). Our causal network approach provides a clear method for identifying local observability criteria, as well as messages that must be passed between blocks, to guarantee diagnosability. In contrast, using an FSM leads to more complicated criteria for decentralized diagnosability [3, 14].

In terms of the computational complexity for diagnostic inference, causal networks make a different space-time tradeoff to the automata-based *Diagnoser-Approach* of [13]. Whereas the Diagnoser-Approach pre-constructs from an FSM a diagnoser, which is a space-intensive model of all possible system faults, and uses relatively simple algorithms to enumerate diagnoses, the causal network approach uses diagnosis algorithms operating on the relatively compact plant model itself, but at the expense of relatively more computationally intensive diagnosis algorithms.

This article shows, similar to [14], the properties necessary for centralized and distributed diagnosability to be equivalent, and the message-passing necessary for this to hold. On top of this, we show how adding sensors or adding equations to local models can reduce this message-passing traffic. Our future work includes a more thorough comparison of our diagnosability results with those of [3, 5, 14], in order to determine relative strengths and weaknesses of the different approaches.

3 Centralized and Decentralized Causal Network Models

This section summarizes our modeling and inference approach to diagnostics and control reconfiguration. We refer the reader to [8, 9] for a full description of causal network modeling for DES applications. We first introduce the model-based formalism, and then extend these notions to capture a distributed model-based formalism. We adopt and extend the model-based representation for diagnosis with behavioral modes of [2].

3.1 Centralized Models

For our discrete event applications, we use the causal network approach for both plant and control modeling. We model the underlying system (plant) behavior in terms of a causal network plant model Φ. The plant model, which can be viewed as an actuator-to-sensor map, describes the *physics* of the system, such as distances, speeds, and their relationships. Note that the plant model Φ simulates the system behavior under both normal and (multiple) abnormal operating modes of the components. We base our model on a set $\mathcal{V}$ of variables and an instantiation θ of those variables.

Definition 1 *A system description (plant model) is a tuple* $\Phi = (\mathcal{V}, \mathcal{G}, \Sigma, \breve{\theta})$*, where* $\mathcal{V}$ *is a set of variables,* $\mathcal{G}$ *is a directed graph called a causal structure whose nodes are members in* $\mathcal{V}$ *and whose directed arcs represent causal relations between pairs of nodes,* Σ *is a set of propositional sentences (called the System Axioms) constructed from members in* $\mathcal{V}$ *based on the topological structure of* $\mathcal{G}$*, and* $\breve{\theta}$ *is the initial state.*

The set of variables consists of the following exclusive subsets: $\mathcal{V}^{\mathcal{A}}$, a set of variables (called *assumables*) representing the failure modes of the components, $\mathcal{V}^{obs}$ (the set of observables) and $\mathcal{V}^{unobs}$ (the set of unobservables). For control purposes, we usually partition $\mathcal{V}^{obs}$ into the subset of sensors, $\mathcal{V}^{sen}$, and the subset of actuators, $\mathcal{V}^{act}$.

The equations in Σ define a set of constraints over the possible values of each variable. The literature contains many specifications of equations, such as using First-Order logic [11], constraints, or multivalued

propositional sentences [2]. In this paper we define the equations using a set of multi-valued propositional sentences, although any of these specifications will work.

We make the following assumptions for the plant model Φ.

1. Φ will not deadlock in any state.
2. Φ has no cycles.[1]
3. Faults persist once they occur.

We can capture structural properties of the model using the directed graph $\mathcal{G}$.[2] For example, if an actuator determines if a motor is on or not, we say that the actuator causally influences the motor. More generally, A may causally influence B if A is a predecessor of B in $\mathcal{G}$. We use $B \propto A$ to denote the causal influence of the value of B by the value of A.[3] $\mathcal{L}(\mathcal{V})$ denotes the language generated by the symbols $\mathcal{V}$ in Φ.

We specify failure-mode instantiations and decompose the possible states into normal states and faulty states as follows:

Definition 2 (Mode-Instantiation) *$\theta^{\mathcal{V}^{\mathcal{A}}}$ is an instantiation of behavior modes to mode-set $\mathcal{V}^{\mathcal{A}}$. Further, we partition $\theta^{\mathcal{V}^{\mathcal{A}}}$ such that $\theta^{\mathcal{V}^{\mathcal{A}}} = \theta^{\hat{\mathcal{A}}} \cup \theta^{\bar{\mathcal{A}}}$, where $\theta^{\bar{\mathcal{A}}}$ denotes normal system behaviour, i.e. all modes are normal, and $\theta^{\hat{\mathcal{A}}}$ denotes a system fault, which may consist of simultaneous faults in multiple components.*

We define the notion of diagnosis as follows:

Definition 3 (Diagnosis) *Given a system description Σ and an instantiation $\theta^{\mathcal{V}^{obs}}$ of $\mathcal{V}^{obs}$, a diagnosis is an instantiation of behavior modes $\theta^{\mathcal{V}^{\mathcal{A}}} = \theta^{\hat{\mathcal{A}}} \cup \theta^{\bar{\mathcal{A}}}$ such that $\Sigma \cup \theta^{\mathcal{V}^{obs}} \cup \theta^{\mathcal{V}^{\mathcal{A}}} \not\models \perp$.*

Definition 4 (Diagnosis Set) *Given a model Φ, D is defined as the corresponding diagnosis set, the set of unique diagnoses possible given every instantiation of observations in $\mathcal{V}_{obs}$.*

3.2 Decentralized (Distributed) models

This section describes our distributed formalism, which applies to collections of interconnected components, which we call blocks. We characterize block i, Φ_i, using the tuple $(X_i, \ell_i, \Theta_i, \Sigma_i, \breve{\theta}_i)$. Here, ℓ_i is the set of inputs (which contains control variables), Θ_i is the output set, Σ_i is the system description for block i, and $\breve{\theta}_i$ is the initial state. In other words, the inputs are the only nodes with causal arcs into the block, and the outputs are the only nodes with causal arcs out of the block. Typically, we have causal dependence of block outputs on inputs, *i.e.*$\Theta_i \propto \ell_i$.[4] We say that block X is a neighbor of block Y if the inputs of X are connected to the outputs of Y, or vice versa.

We assume that the set of equations Σ is *complete* given the initial state $\breve{\theta}$, meaning that the values for all variables are derivable from $\breve{\theta}$. Our notion of completeness translates to a block being complete given knowledge of the values of the block's inputs and observables.

Lemma 1 *A block $(X_i, \ell_i, \Theta_i, \Sigma_i, \breve{\theta}_i)$ is deductively complete if $\breve{\theta}_i = \ell_i \cup \mathcal{V}_i^{obs}$.*

We can derive this result based on the relation between observability and independence in a directed graph, such as $\mathcal{G}$. We use an example to demonstrate this property: given block k with two inputs (I_1, I_2) and one output O_k, we may have equations denoting fault conditions of the form:

$$[I_1 = \alpha] \wedge [I_2 = \beta] \wedge [\mathcal{V}_k^{\mathcal{A}} = fault] \supset [O_k = \gamma].$$

For every fault assignment to $\mathcal{V}_k^{\mathcal{A}}$ one can determine the input/output variables that must be known to determine that fault assignment. We can use a topological property known as D-separation to identify which predecessor variable in graph $\mathcal{G}$ can influence a node in $\mathcal{G}$. A node N is D-separated from it causal predecessors P in $\mathcal{G}$ by its direct parents in $\mathcal{G}$, $pa(N)$. Hence, if the values of $pa(N)$ are known, then node N is independent of the values of $P \setminus pa(N)$. For example, if $pa(N)$ consist of observables, then when $pa(N)$ are observed N is independent of the rest of the nodes in $\mathcal{G}$, i.e., the rest of the system.

We define a decomposition function to formalize how a centralized model is split into a collection of interconnected blocks.

Definition 5 (Decomposition Function) *a decomposition function is a mapping $\psi(\Phi) = \Phi^{\delta}$ that decomposes a centralized model Φ into a distributed model $\Phi^{\delta} = \{\Phi_1, ..., \Phi_m\}$. A distributed model is induced by a decomposition function ψ into a decomposition Π over the system variables $\mathcal{V}$, i.e. a collection $\{X_1, ..., X_m\}$ of nonempty subsets of $\mathcal{V}$ such that (1) $\forall i = 1, ..., m$, $X_i \subseteq 2^{\mathcal{V}}$; (2) $\mathcal{V} = \cup_i (X_i | X_i \in \Pi)$. When $\xi_{ij} = X_i \cap X_j \neq \emptyset$, we call ξ_{ij} the separating set, or common set, of variables between Φ_i and Φ_j.*

[1] This assumption is for the untimed model only; in the timed model [7], we assume that no unobservable cycle is present in Φ, as in [13, 14].

[2] In other model specifications, e.g. [11], these structural relations are captured using logical sentences.

[3] This notion of causal influence does not guarantee that A influences B, but that A *may* influence B.

[4] The causal function $\propto$ can be be generalized to include propositions, relations, probabilistic functions, qualitative differential equations, etc. We don't address such a generalization here.

4 Diagnosing Distributed Systems

This section defines a notion of diagnosability for distributed systems. We first specify the notion of diagnoses induced by a decomposition. Using our formalism, we do not need to alter the notion of diagnosis; rather, we decompose the model and define diagnoses in the standard way.

Definition 6 (Diagnostic mapping) *: Given a model Φ with corresponding diagnosis set D, and a distributed model Φ^δ such that $\psi(\Phi) = \Phi^\delta$, $\exists$ a corresponding distributed diagnosis set D^δ induced by the mapping ψ. Given diagnoses D_i^δ of block i, we have*

$$D^\delta = \bigcup_{i=1}^{m} D_i^\delta.$$

We are interested in characterizing the diagnosis set resulting from the decomposition function ψ, and we use the following notion of diagnosability to do so.

Definition 7 (Diagnosability) $\exists$ *some* $O \subseteq \theta^{\mathcal{V}_{obs}}$ *such that* $\Sigma \cup O \cup \theta_i^{\hat{\mathcal{A}}} \not\models \perp$, *for every* $\theta_i^{\hat{\mathcal{A}}} \in \theta^{\hat{\mathcal{A}}}$.

This means that a system is *diagnosable* if every fault mode in the system can be isolated with an appropriate set of observables. If Φ does not contain appropriate sensors/actuators such that any fault mode cannot be so isolated, then it is termed non-diagnosable, and the system as a whole is non-diagnosable (or partially diagnosable).

Note that in this article, for clarity of exposition we restrict ourselves to next-step control actions and next-step transitions. There is a direct extension to timed systems (see [7]), where we define arbitrary-step transitions. In this temporal model, the notion of diagnosability is extended such that the observation $O \in \mathcal{V}_{obs}$ in the next-step model to a sequence of observations $O_t, O_{t+1}, ..., O_{t+k}$. In this case, the notion of diagnosability has a clear relation to that defined by Sampath et al. for the diagnoser approach [13]. Whereas Sampath et al. construct a diagnoser within which a trace of transitions contains observable events that enable the failure of a particular type within a finite delay, in our timed representation we record a sequence of observable variables such that instantiating those observables in the timed model allows us to compute the particular failure mode.

To define distributed diagnosability, we need a notion of *local observables*:

Definition 8 *Given block k with variable set X_k, we define observables local to block k, $\mathcal{V}_k^{obs}$, as those contained within that block, i.e.*$\mathcal{V}_k^{obs} = \{O | O \subseteq X_k \cap \mathcal{V}^{obs}\}$.

A strong requirement for diagnosability for every block is deductive completeness:

Lemma 2 *A block $(X_i, \ell_i, \Theta_i, \Sigma_i, \breve{\theta}_i)$ is diagnosable if it is deductively complete, i.e., if $\breve{\theta}_i = \ell_i \cup \theta_i^{\mathcal{V}^{obs}}$.*

This makes the assumption that every block knows its inputs, and that all sensors and actuators are observable. The following section examines methods for guaranteeing this.

We now distinguish two classes of object diagnosability which do not require message-passing, strong and weak. These classes are based on whether diagnosability can be determined locally or not.

Definition 9 (Strong Object Diagnosability) *Given block k, for every $\theta_i^{\hat{\mathcal{A}}} \in \theta^{\hat{\mathcal{A}}}$, $\exists$ some $O \subseteq \theta_k^{\mathcal{V}_{obs}}$ such that $\Sigma \cup O \cup \theta_i^{\hat{\mathcal{A}}} \not\models \perp$.*

Strong Object Diagnosability captures the notion that a distributed object can compute all internal failure modes based only on local observable data (e.g. sensor and actuator data). In contrast, under *Weak Object Diagnosability* a distributed object cannot compute all internal failure modes based only on local data (e.g. sensor data), but can do so only with access to observable data available from neighbouring objects.

Definition 10 (Weak Object Diagnosability) *Given block k, $\exists$ some $\theta_i^{\hat{\mathcal{A}}} \in \theta^{\hat{\mathcal{A}}}$, and some $O \subseteq \theta_k^{\mathcal{V}^{obs}}$ such that $\Sigma \cup O \cup \theta_i^{\hat{\mathcal{A}}} \models \perp$ but $\Sigma \cup O' \cup \theta_i^{\hat{\mathcal{A}}} \not\models \perp$ for some $O \subset O' \subset \{\theta^{\mathcal{V}^{obs}} \cup \theta_k^{\mathcal{V}^{obs}}\}$.*

We can also extend these notions to those of the distributed system (DS):

Definition 11 (Strong DS Diagnosability) $\forall X_i \in \Pi$, $\forall \theta_i^{\hat{\mathcal{A}}} \in \theta^{\hat{\mathcal{A}}}$, $\exists$ *some* $O \subseteq \theta_k^{\mathcal{V}^{obs}}$ *such that* $\Sigma \cup O \cup \theta_i^{\hat{\mathcal{A}}} \not\models \perp$.

In other words, a distributed system is strongly diagnosable if, for every object in the distributed system, we can compute all internal failure modes based only on local data (e.g. sensor data). The system is *Weak DS Diagnosable* otherwise.

We can prove some important properties using these definitions. First, if we decompose a centralized system into a set of distributed components, the distributed system is diagnosable *only if* the centralized system is diagnosable.

Lemma 3 *: Given a distributed model Φ^δ and its corresponding centralized version Φ, Φ^δ is DS Diagnosable only if Φ is diagnosable.*

A more important property is that a distributed model has diagnosability properties no stronger than those of the corresponding centralized model.

Lemma 4 *: Given a distributed model Φ^δ and its corresponding centralized version Φ, $\nexists\ \theta_i^{\hat{A}} \in \theta^{\hat{A}}$ such that $\theta_i^{\hat{A}}$ is diagnosable in Φ^δ but undiagnosable in Φ.*

5 Centralized and Decentralized Diagnosability

This section examines how we can either satisfy or relax the distributed diagnosability requirements of the previous section, i.e., knowing all inputs and observables within each block. We will examine message-passing and "strengthening" the equations in each block.

In the following, we assume diagnosability of the centralized model and explore notions of distributed system diagnosability. Since the information processed in each local block is (potentially) incomplete, the diagnosability of each local block is often incomplete, and the system is not diagnosable. There are three ways in which we can guarantee system diagnosability: (1) message-passing between blocks; (2) adding extra equations to particular blocks together with message-passing; or (3) adding extra sensors to each block. The second option will provide less inter-block message traffic than the first, at the cost of more processing within the blocks (given the extra equations). The third option removes the need for message-passing, but at the cost of additional sensors. We now examine each approach in turn.

5.1 Message-Passing Approach

This section examines the use of message-passing to achieve distributed system diagnosability. The questions that need to be addressed include: (1) what information needs to be exchanged; (2) between which blocks does information need to be exchanged; and (3) the timing and method of information exchange. In this article we focus on the first two questions.

If we assume that all local sensors and actuators are observable, we can guarantee Strong DS-Diagnosability if we pass messages consisting of the set of variables common between all blocks. This guarantees that every block will receive data for its inputs.

Lemma 5 *A system consisting of n blocks is Strong DS-Diagnosable if the message-passing data $\mathcal{M} \subseteq \mathcal{V}$ is such that $\mathcal{M} = \Xi \setminus \mathcal{V}^{obs}$, and $\Xi = \bigcup_{i,j}\{\xi_{ij} | i,j = 1,...,n,\ i \neq j\}$ is the set of common variables in the system.*

This result is easy to prove since our definition of decentralized model provides a clear specification of system decomposition as well as the data shared between blocks. Note that if an input is observable, then that variable does not have to be included in the set of data passed between blocks. Also, the information that must be exchanged between any pair of blocks is defined by the block connectivity: if Block Φ_i has some input set ℓ_i, it must receive data from all parent blocks in $\mathcal{G}$. In other words, the set of variables common between two blocks identifies which data must be passed between the blocks, and the direction of data transfer.

Definition 12 *The information that must be sent from sub-model Φ_i to Φ_j is ξ_{ij}, where some outputs of Φ_i are inputs for Φ_j.*

One of the potential drawbacks to this message-passing approach is that $|\mathcal{M}|$ can be quite large, leading to heavy message-passing traffic.

5.2 Combined Equation Strengthening and Message-Passing Approach

If guaranteeing diagnosability in a distributed model would lead to heavy message-passing traffic, then we can reduce that traffic by adding extra equations to the local models.

How can one determine what extra equations need to be added to distributed models, and what data needs to be passed to distributed models? In the case of extra equations, the language generated in a distributed model can be is typically a strict subset of the language of the diagnosable centralized model, given our scheme of using only *local* observables in the distributed model. In other words, incompleteness occurs if $\mathcal{L}(\Phi) \supset \bigcup_i \mathcal{L}(\Phi_i)$. Hence, enumerating the sentences in $\Sigma' = \mathcal{L}(\Phi) \setminus \bigcup_i \mathcal{L}(\Phi_i)$ defines the necessary extra sentences Σ' for the entire model.

We can use results from [6] to identify the equations needed to strengthen a *local* model. These equations are determined from the interconnection topology of the collection of blocks in the distributed system. In other words, the extra equations for block Φ_i need to contain only variables from blocks directly connected to Φ_i. The nature of these equations is model-dependent, although at worst these equations would consist of the equations from the sub-system consisting of Φ_i and its neighbors.

The data required for message-passing is bounded by the observable data contained in Σ'.

Lemma 6 *Given an incomplete distributed version Φ^δ of a diagnosable centralized model Φ, we can extend Φ^δ to be Weak-DS diagnosable by generating sentences*

given by $\Sigma' = \mathcal{L}(\Phi) \setminus \bigcup_i \mathcal{L}(\Phi_i)$*, and passing observables* O' *given by* $O' = \{O | O \subseteq \mathcal{L}(\Phi) \setminus \bigcup_i \mathcal{L}(\Phi_i)\}$.

The benefit of this approach is that it reduces message-passing traffic. The drawbacks include: (1) in the worst case, the complexity of diagnosing local models can approach that of the centralized model; and (2) it is hard to define local equations in a domain-independent manner.

5.3 Improved Observability Approach

It is possible to use observability criteria to determine what must be known to guarantee diagnosability. A system is strongly DS diagnosable if every block has observable inputs.

Lemma 7
Given a distributed system $\Phi = \{\Phi_1, ..., \Phi_m\}$*, if* $\forall \Phi_i$*, we have* $\ell_i \subseteq \theta_i^{\mathcal{V}^{obs}}$*, then* $\forall \Phi_i \in \Phi$*,* $\forall \theta_k^{\hat{A}} \in \theta^{\hat{A}}$*,* $\exists$ *some* $O \in \theta_i^{\mathcal{V}^{obs}}$ *such that* $\Sigma \cup O \cup \theta_k^{\hat{A}} \not\models \perp$.

This can be guaranteed by placing sensors at all inputs to every block. This will eliminate all message-passing traffic, but at the expense of many sensors and/or actuators. In the real world, it is unlikely that the necessary number of sensors will be made available. However, we have identified three complementary approaches for guaranteeing distributed system diagnosability: message-passing, local model strengthening, and increased local observability. Our future work includes examining the tradeoffs among these approaches, both theoretically and empirically.

6 Summary and Conclusions

This document has described some fundamental issues in diagnosing distributed systems. In particular, we identified criteria for observability and message-passing amongst distributed models that will guarantee complete diagnosability of a distributed system. We examined three ways in which we can guarantee complete system diagnosability for a distributed system: (1) message-passing between blocks; (2) adding extra equations to particular blocks together with message-passing; or (3) adding extra sensors to each block. We discussed the strong and weak points of each approach.

We describe in [6] a mechanism for computing distributed diagnoses using system topology and observability properties that avoids the message-passing overhead of passing data for variables, but instead passes diagnoses. The tradeoff in this approach is that we need to pre-compile diagnoses for groups of blocks. Hence, it decreases the message-passing overhead at the expense of increasing the memory requirements at each distributed block.

References

[1] C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer, 1999.

[2] Adnan Darwiche. Model-based diagnosis using structured system descriptions. *Journal of Artificial Intelligence Research*, 8:165–222, 1998.

[3] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Journal of Discrete Event Dynamical Systems: Theory and Application*, 1- (1-2):33–86, 2000.

[4] L. E. Holloway and S. Chand. Distributed fault monitoring in manufacturing systems using concurrent discrete event observations. *Integrated Computer-Aided Engineering*, 3(4):244–254, 1996.

[5] S. Jiang and R. Kumar. Decentralized control of discrete event systems with specializations to local control and concurrent systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 30(5), Oct. 2000.

[6] G. Provan. A Model-Based Diagnosis Framework for Distributed Embedded Systems. In *Proc. KR '2002*. Morgan Kaufmann, 2002.

[7] G. Provan and Y.-L. Chen. Modeling, diagnosis, and control of timed discrete event systems using temporal causal networks. In *Proc. 1998 Intl. Workshop on Discrete Event Systems (WODES'98)*, pages 152–154, Cagliari, Italy, August 1998.

[8] G. Provan and Y.-L. Chen. Component-based modeling and diagnosis of process-control systems. In *Proc. 1999 IEEE Intl. Symposium on Computer-Aided Control System Design*, pages 194–199, Kohala Coast, HI, August 1999.

[9] G. Provan and Y.-L. Chen. Model-based diagnosis and control reconfiguration for discrete event systems: An integrated approach. In *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, AZ, December 1999.

[10] G. Provan and Y.-L. Chen. A Comparison of Finite State Machine and Causal Network Approaches to the Modeling and Diagnosis of Discrete Event Systems. In *Proc. American Control Conference*, June 2000.

[11] R. Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32:57–96, 1987.

[12] V. Chandra S. Jiang, Z. Huang and R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46(8):1318–1321, August 2001.

[13] M. Sampath, R. Sangupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosibility of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, September 1995.

[14] E. Su and W. M. Wonham. Decentralized diagnosis for discrete event systems. In *Proc. Conf. on Info. Science and Systems*, Princeton, NJ, March 2000.

[15] S. Zad, R. Kwong, and W. Wonham. Fault diagnosis in discrete-event systems: Framework and model reduction. In *37th IEEE Conf. on Decision and Control*, pages 3769–3774, Tampa, FL, 1998.