The Effect of Observations on the Complexity of Model–Based Diagnosis

Adnan Darwiche
Department of Mathematics
American University of Beirut
PO Box 11 - 236, Beirut, Lebanon
darwiche@aub.edu.lb

Gregory Provan
Rockwell Science Center
1049 Camino Dos Rios
Thousand Oaks, CA 91360
provan@risc.rockwell.com

Abstract

This paper shows how to efficiently diagnose systems by making use of observations. In particular, we present two theorems concerning the effect of observations on the complexity of Model–Based Diagnosis. The first theorem shows how the presence of certain observations allows us to decompose a diagnostic reasoning task into independent reasoning tasks on subsystems. The second theorem shows how the absence of certain observations allows us to ignore parts of a system during diagnostic reasoning. Another main contribution of this paper is an application of these theorems to diagnosing discrete–event systems. In particular, we identify observability and modularity characteristics of discrete–event systems that make them amenable to the presented theorems and, hence, to any diagnostic approach that employs these theorems effectively. This also explains why a particular approach that we have presented elsewhere has proven effective for diagnosing these systems.

Introduction

This paper describes results that have evolved out of two lines of research, one theoretical and the other practical. In the first line of research, (Darwiche 1995b; 1995a) described a computational approach for model–based diagnosis (MBD) using structured system descriptions. In the second line, (Darwiche & Provan 1996) applied this approach to the diagnosis of discrete–event systems to find out that the diagnosis of such systems is not as costly as the literature on MBD may suggest. One goal of this paper is to formally state the aspect of discrete–event systems and the property of our approach that are both responsible for the surprising computational efficiency of diagnosing discrete-event systems.

Specifically, we will provide two general theorems that state conditions which facilitate the computation of model–based diagnoses. We will then show that discrete–event systems satisfy the conditions of these theorems and that the structure–based approach (described in (Darwiche & Provan 1996; Darwiche 1995b; 1995a)) implicitly applies these theorems. Both theorems concern the effect of system observations on the complexity of model–based diagnosis. The first theorem states conditions under which the *presence* of observations allows one to split the computation of model–based diagnoses into smaller computations that can be performed independently. The second theorem states conditions under which the *absence* of system observations allows one to ignore some parts of a system description when performing such computations.

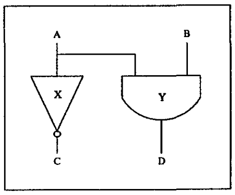
We also present in this paper an explanation of why a certain model–based approach (Darwiche & Provan 1996) is effective for diagnosing discrete–event systems. The value of such explanation is two-fold. First, it formalizes an aspect of diagnostic applications that makes them amenable to model–based techniques, thereby allowing us to identify a class of efficiently–diagnosable problems. Second, it lays the ground for utilizing this aspect computationally by other diagnostic approaches.

The rest of this paper is structured as follows. The following section reviews basic notions of model–based diagnosis that we need to state our formal results. The next section states two theorems that describe the effect of observations on the computation of model–based diagnoses. The section before last shows how to apply these theorems to the class of discrete–event systems. We finally close with some concluding remarks. Proofs of thereoms are omitted for space limitation, but can be found in a longer version of this paper, available from the authors.

Model–Based Diagnosis

In model-based diagnosis, we use the term *system description* to denote a system model (de Kleer, Mackworth, & Reiter 1992). Traditionally, a system description consists of a set of logical sentences Δ called a *database* and a set of distinguished symbols $\mathbf{A} = \{ok(X), ok(Y), \ldots\}$ called *assumables.* Assumables represent the health of components and are initially assumed to be true (see Figure 1 for an example).

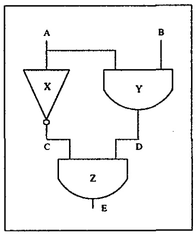
A diagnosis problem emerges when assumables can no longer be justified. Specifically, given some literals



$$\mathbf{P} = \{A, B, C, D\}$$
$$\mathbf{A} = \{ok(X), ok(Y)\}$$

$$\Delta = \left\{ \begin{array}{lcr} A \wedge ok(X) & \supset & \neg C, \\ \neg A \wedge ok(X) & \supset & C, \\ A \wedge B \wedge ok(Y) & \supset & D, \\ \neg(A \wedge B) \wedge ok(Y) & \supset & \neg D \end{array} \right\}$$

Figure 1: A system description.



$$\mathbf{P} = \{A, B, C, D, E\}$$
$$\mathbf{A} = \{ok(X), ok(Y), ok(Z)\}$$

$$\Delta = \left\{ \begin{array}{lcr} A \wedge ok(X) & \supset & \neg C, \\ \neg A \wedge ok(X) & \supset & C, \\ A \wedge B \wedge ok(Y) & \supset & D, \\ \neg(A \wedge B) \wedge ok(Y) & \supset & \neg D, \\ C \wedge D \wedge ok(Z) & \supset & E, \\ \neg(C \wedge D) \wedge ok(Z) & \supset & \neg E \end{array} \right\}$$

Figure 2: A system description.

ϕ that represent an observed system behavior, the system is considered faulty if ϕ is inconsistent with $\Delta \cup \mathbf{A}$. In this case, one needs to relax some of the assumables (that is, replace instances of $ok(.)$ with instances of $\neg ok(.)$) in order to restore consistency. A particular relaxation of these assumables is called a *diagnosis.* In Figure 1, a system observation $\{C, D\}$ would indicate a failure. Moreover, there are three diagnoses in this case: $ok(X) \wedge \neg ok(Y), \neg ok(X) \wedge ok(Y)$ and $\neg ok(X) \wedge \neg ok(Y)$.

We have the following formal definition of a system description, which we adopt in the rest of this paper.

Definition 1 *Let* $\mathbf{P}$ *be a set of atomic propositions. A* $\mathbf{P}$*-literal is a positive or negative literal whose atom belongs to* $\mathbf{P}$. *Two literals are distinct if they do not refer to the same atom. A* $\mathbf{P}$*-clause is a disjunction of distinct* $\mathbf{P}$*-literals. A* $\mathbf{P}$*-instantiation is a conjunction of distinct* $\mathbf{P}$*-literals. A full* $\mathbf{P}$*-instantiation is a* $\mathbf{P}$*-instantiation containing exactly one literal for each atom in* $\mathbf{P}$.

Definition 2 (System Description) *A system description is a triple* $(\mathbf{P}, \mathbf{A}, \Delta)$, *where* $\mathbf{P}$ *is a set of atomic propositions, called non-assumables,* $\mathbf{A}$ *is a set of atomic propositions, called assumables, and* Δ *is a set of propositional sentences constructed from atoms in* $\mathbf{P}$ *and* $\mathbf{A}$.

Given the notion of a system description, we can define two key terms in model–based diagnosis:

Definition 3 (Observation) *Given a system description* $(\mathbf{P}, \mathbf{A}, \Delta)$, *a system observation is a* $\mathbf{P}$*–instantiation.*

Definition 4 (Diagnosis) *Given a system description* $(\mathbf{P}, \mathbf{A}, \Delta)$ *and a system observation* ϕ, *a diagnosis is a full* $\mathbf{A}$*-instantiation that is consistent with* $\Delta \cup \phi$.

The standard notion for characterizing diagnoses is minimal conflicts:

Definition 5 (Conflict) *Given a system description* $(\mathbf{P}, \mathbf{A}, \Delta)$, *a conflict of system observation* ϕ *is an* $\mathbf{A}$*–clause* β *such that* $\Delta \cup \phi \models \beta$. *A conflict is minimal if it is not subsumed by any other conflict. The set of minimal conflicts for system description* $(\mathbf{P}, \mathbf{A}, \Delta)$ *and observation* ϕ *will be denoted by* $MinConflicts^{\Delta}_{\mathbf{A}}(\phi)$.

When clear from the context, we drop the superscript Δ, the subscript $\mathbf{A}$, or both, from the notation $MinConflicts^{\Delta}_{\mathbf{A}}$.

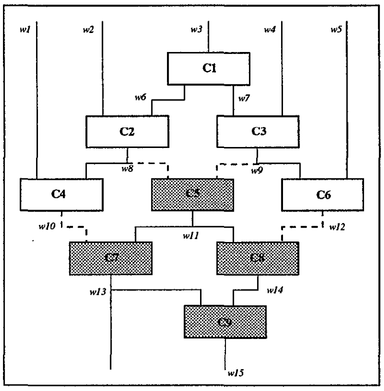
Minimal conflicts are the basis for characterizing and computing diagnoses because an instantiation of the assumables is a diagnosis precisely when it is consistent with the set of all minimal conflicts (de Kleer, Mackworth, & Reiter 1992).

For example, if we observe that both C and D are true in Figure 1, we conclude that one of the gates must be malfunctioning: under normal conditions, C being true implies that A is false, which further implies that D is false. Therefore, the clause $\neg ok(X) \vee \neg ok(Y)$ is implied by the system observation $\phi = \{C, D\}$ and system description Δ. Moreover, all other $\mathbf{A}$–clauses implied by the system description and observation are subsumed by $\neg ok(X) \vee \neg ok(Y)$. This means that $\neg ok(X) \vee \neg ok(Y)$ is the only minimal conflict:

$$MinConflicts^{\Delta}_{\mathbf{A}}(\phi) = \{\neg ok(X) \vee \neg ok(Y)\}.$$

The Effect of Observations on MBD

We now present two theorems that facilitate the computation of conflicts under the presence/absence of certain system observations. We first state the theorems, explain what each means with respect to a concrete example, and then explain them with respect to the prototypical system in Figure 3.



$$\mathbf{P} = \{w_1, \ldots, w_{15}\}$$
$$\mathbf{A} = \{ok(C_1), \ldots, ok(C_9)\}$$

Figure 3: A schematic diagram of a system.

Theorem 1 (Decomposition) *Let $(\mathbf{P}, \mathbf{A}, \Delta_1)$ and $(\mathbf{P}, \mathbf{A}, \Delta_2)$ be two system descriptions and let $\mathbf{S}$ be all atoms that are common between Δ_1 and Δ_2. If ϕ is a system observation that contains a literal for each atom in $\mathbf{S}$, then*

$$\begin{aligned} &MinConflicts_{\mathbf{A}}^{\Delta_1 \cup \Delta_2}(\phi) \\ &= MinConflicts_{\mathbf{A}}^{\Delta_1}(\phi_{\Delta_1}) \cup MinConflicts_{\mathbf{A}}^{\Delta_2}(\phi_{\Delta_2}). \end{aligned}$$

Here, ϕ_{Δ_1} is the subset of observation ϕ with atoms occurring in Δ_1, and ϕ_{Δ_2} is the subset of ϕ with atoms occurring in Δ_2.

This theorem is stating conditions under which one can decompose the computation of certain conflicts into independent computations that can be performed in parallel. Specifically, if two sub–systems interact through a set of variables, and the state of these variables is known, then one can process the sub–systems independently and then combine the results. Consider Figure 1 for an example. Let

$$\Delta_1 = \left\{ \begin{array}{c} A \wedge ok(X) \supset \neg C, \\ \neg A \wedge ok(X) \supset C \end{array} \right\};$$

$$\Delta_2 = \left\{ \begin{array}{c} A \wedge B \wedge ok(Y) \supset D, \\ \neg(A \wedge B) \wedge ok(Y) \supset \neg D \end{array} \right\}.$$

Only one atom, A, is common between Δ_1 and Δ_2. Therefore, if $\phi = \{\neg A, \neg C, D\}$, then we can decompose the computation of conflicts as follows:

$$\begin{aligned} &MinConflicts^{\Delta_1 \cup \Delta_2}(\phi) \\ &= MinConflicts^{\Delta_1}(\phi_{\Delta_1}) \cup MinConflicts^{\Delta_2}(\phi_{\Delta_2}), \end{aligned}$$

where $\phi_{\Delta_1} = \{\neg A, \neg C\}$, $\phi_{\Delta_2} = \{\neg A, D\}$, and

$$\begin{aligned} MinConflicts^{\Delta_1 \cup \Delta_2}(\phi) &= \{\neg ok(X), \neg ok(Y)\} \\ MinConflicts^{\Delta_1}(\phi_{\Delta_1}) &= \{\neg ok(X)\} \\ MinConflicts^{\Delta_2}(\phi_{\Delta_2}) &= \{\neg ok(Y)\}. \end{aligned}$$

For another more general example, consider the system in Figure 3. Suppose that Δ_1 contains the description of components C_1, C_2, C_3, C_4 and C_6, and that Δ_2 contains the description of remaining components. Note that w_8, w_9, w_{10} and w_{12} are all the atoms common between Δ_1 and Δ_2. According to Theorem 1, computing the conflicts of any system observation ϕ that fixes the state of these atoms can be done through two smaller and independent computations, one involving Δ_1 and the other involving Δ_2.

We now turn to the second theorem, which states conditions under which one can prune parts of the system description when computing conflicts.

Theorem 2 (Pruning) *Let $(\mathbf{P}, \mathbf{A}, \Delta_1)$ and $(\mathbf{P}, \mathbf{A}, \Delta_2)$ be two system descriptions. If*

1. Δ_1 and Δ_2 share no assumables; and

2. every instantiation of $\mathbf{S} \cup \mathbf{A}$ is consistent with Δ_2, where $\mathbf{S}$ are the common atoms between Δ_1 and Δ_2,

then for any system observation ϕ whose atoms appear in Δ_1, we have

$$MinConflicts_{\mathbf{A}}^{\Delta_1 \cup \Delta_2}(\phi) = MinConflicts_{\mathbf{A}}^{\Delta_1}(\phi).$$

This theorem allows us to ignore some parts of the system description under certain conditions. As we shall see next, these conditions are not difficult to establish in general. Consider Figure 2 and let

$$\Delta_1 = \left\{ \begin{array}{c} A \wedge ok(X) \supset \neg C, \\ \neg A \wedge ok(X) \supset C, \\ A \wedge B \wedge ok(Y) \supset D, \\ \neg(A \wedge B) \wedge ok(Y) \supset \neg D \end{array} \right\};$$

$$\Delta_2 = \left\{ \begin{array}{c} C \wedge D \wedge ok(Z) \supset E, \\ \neg(C \wedge D) \wedge ok(Z) \supset \neg E \end{array} \right\}.$$

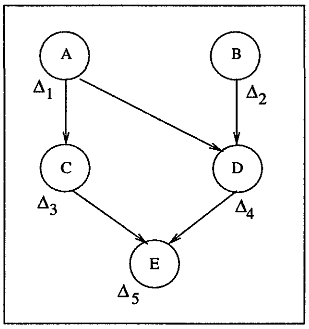
Then Δ_1 and Δ_2 share no assumables in this case, which establishes the first condition of Theorem 2. Moreover, the common atoms between Δ_1 and Δ_2 are $\mathbf{S} = \{C, D\}$, and one can show that every instantiation of $\mathbf{S} \cup \mathbf{A} = \{C, D, ok(X), ok(Y), ok(Z)\}$ is consistent with Δ_2, therefore establishing the second condition of Theorem 2. Given these two conditions, the conflicts of any observation whose atoms appear in Δ_1 can be computed while ignoring Δ_2. For example, if $\phi = \{\neg A, \neg C, D\}$, then

$$\begin{aligned} MinConflicts^{\Delta_1 \cup \Delta_2}(\phi) &= MinConflicts^{\Delta_1}(\phi) \\ &= \{\neg ok(X), \neg ok(Y)\}. \end{aligned}$$

Similarly, if $\phi = \{C, D\}$, then

$$\begin{aligned} MinConflicts^{\Delta_1 \cup \Delta_2}(\phi) &= MinConflicts^{\Delta_1}(\phi) \\ &= \{\neg ok(X) \vee \neg ok(Y)\}. \end{aligned}$$

To provide more intuition about the conditions of Theorem 2, consider the system in Figure 3. Suppose that Δ_1 contains the description of components C_1, C_2, C_3, C_4 and C_6, and that Δ_2 contains the description of remaining components.



$$\begin{aligned}
\Delta_1 &= \Delta_2 = \{\} \\
\Delta_3 &= \left\{ \begin{array}{lcr} A \wedge ok(X) & \supset & \neg C, \\ \neg A \wedge ok(X) & \supset & C \end{array} \right\} \\
\Delta_4 &= \left\{ \begin{array}{lcr} A \wedge B \wedge ok(Y) & \supset & D, \\ \neg(A \wedge B) \wedge ok(Y) & \supset & \neg D \end{array} \right\} \\
\Delta_5 &= \left\{ \begin{array}{lcr} C \wedge D \wedge ok(Z) & \supset & E, \\ \neg(C \wedge D) \wedge ok(Z) & \supset & \neg E \end{array} \right\}
\end{aligned}$$

Figure 4: A structured system description.

- The first condition of Theorem 2 says that Δ_1 and Δ_2 should not share assumables, which can be enforced by using different assumables to represent the health of different components.
- The set $\mathbf{S}$ in this case contains w_8, w_9, w_{10} and w_{12}, and the second condition of Theorem 2 says that any instantiation of $\mathbf{S} \cup \mathbf{A}$ should be consistent with Δ_2. This condition therefore requires the system description Δ_2 not to eliminate any possible state of the atoms in $\mathbf{S}$. This should be self–imposed since the atoms in $\mathbf{S}$ are inputs to the sub–system described by Δ_2. Moreover, the system description Δ_2 is not supposed to constrain the state of inputs to the system; it should merely contrain its outputs given its inputs.

The decomposition and pruning theorems are simple to apply if one uses a ***structured system description*** which was introduced in (Darwiche 1995b; 1995a) and used for diagnosing discrete–event systems in (Darwiche & Provan 1996). A structured system description — a symbolic causal network in (Darwiche 1995b) — is a traditional system description constrained by a system structure in the form of a directed acyclic graph. Figure 4 contains a structured system description for the system of Figure 2. Roughly speaking, one can construct a structured system description as follows. First, include a node in the graph for each port (input/output) of a system component. Second, connect input ports of each component to its output port by a directed arc. Finally, associate with each component's output port a set of propositional sentences (component description) describing the value of that port in terms of its input ports and assumables. Component descriptions must satisfy some local conditions that are described elsewhere (Darwiche 1995b; 1995a). Note that this approach covers many key aspects of real-world systems. For example, this approach allows models to explicitly contain cycles, such as those induced by feedback; such cycles "unravel" when we temporally unfold the structured system description, as described in (Darwiche & Provan 1996). In addition, we can model multiple behavior modes, such as a sensor being OK, stuck-on and stuck-off (Darwiche 1995b; 1995a).

Given a structured system description, Theorems 1 and 2 can be interpreted/applied as follows:

Theorem 1. For each literal l in the observation ϕ, delete arcs that are outgoing from node P in the graph, where P is the atom of literal l. Then add literal l to each database Δ_Q that is associated with a child Q of node P. This process may result in a number of disconnected graphs, each of which can be processed independently to compute minimal conflicts. The results of these independent computations should be disjoined to obtain the minimal conflicts of the whole system.

Theorem 2. For each leaf node P in the graph that is not mentioned in the system observation ϕ, delete P and its associated database Δ_P from the structured system description. This can be applied recursively to delete further nodes. The minimal conflicts of the pruned system description will be the same as the minimal conflicts of the original system description.

These simple graph–based operations are justified by Theorems 1 and 2 and by properties of structured system descriptions. Figure 5 contains an example application of these theorems to the structured system description in Figure 4 when the system observation ϕ is $\{\neg A, \neg C, D\}$. Using Theorem 2 and the fact that E is not part of the observation, we prune node E and its associated database Δ_5, leading to Figure 5(a). Using Theorem 1, we delete the arc from node A to node D, adding $\neg A$ to Δ_4, which leads to Figure 5(b). We now have two independent sub–systems. There is only one minimal conflict, $\neg ok(X)$, with respect to the first sub–system, and another minimal conflict, $\neg ok(Y)$, with respect to the second sub–system. Disjoining these conflicts we obtain $\{\neg ok(X), \neg ok(Y)\}$, which are the minimal conflicts with respect to the full system.

Diagnosing Discrete–Event Systems

A discrete–event system is a continuous- or discrete–time system in which control actions are issued at discrete points in time. Discrete–event systems can model a broad class of real–world systems, such as manufacturing processes, communication networks, computer networks, and many industrial processes (Sampath *et al.* 1995). A factory assembly line for filling bottles, with controls such as "put bottle on conveyor belt"

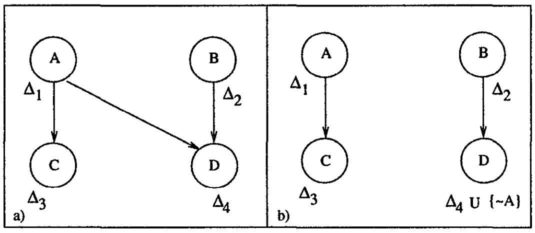


Figure 5: Pruning & Decomposition theorems.

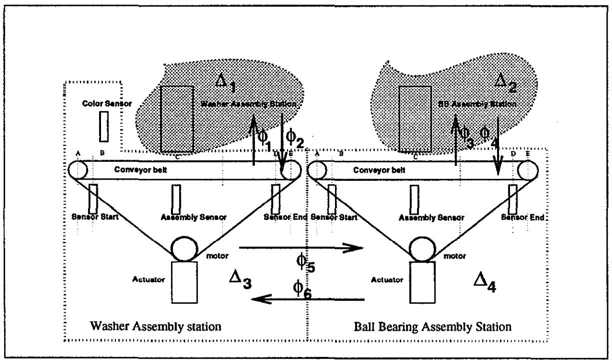


Figure 6: Diagram of a factory. This system consists of a pair of conveyor belts that move a block first to a washer assembly station, at which a robot picks up a washer and places it on the block, then to a ball-bearing assembly station, at which a ball-bearing is released into the washer, and finally to the end of the conveyor.

and "start bottle-filling-machine," is an example of a discrete-event system. Figure 6 shows an example of a discrete-event system that performs different operations on a block as it moves across a conveyor system. A diagnostic reasoner for this system would need to model the system so it can detect and then isolate sensor, actuator and motor failures.

To diagnose such systems, we developed a model-based diagnostic reasoner using structured system descriptions (Darwiche & Provan 1996). But contrary to what one would expect (given the usual computational difficulty with model-based diagnosis), we were able to obtain reasonable response times on systems of realistic complexity. Through further analysis of these results, we can partially explain them in terms of two points. First, the structure-based approach we use does apply Theorems 1 and 2 implicitly. Second, discrete-event systems have two characteristics that make them amenable to these theorems:

1. *Modularity:* Discrete-event systems are typically constructed from standard building-blocks (e.g., conveyors, welders, and drill-presses) which are interfaced using sensors and actuators. These building blocks are effectively sub-systems that communicate by reading sensors and setting actuators. For example, when the block arrives at the washer assembly station in Figure 6, the washer assembly sensor triggers, which stops the conveyor belt and starts the process described by Δ_1. When the process is complete, a sensor in the washer assembly station triggers, announcing that the process is complete and restarting the conveyor process described by Δ_2.
2. *Observability:* Discrete-event systems are highly observable by design and the controller has access to the state of sensors and actuators at each control cycle. Therefore, up to the point where an abnormality is present, the history of sensor and actuator readings can be made available.

We now show how these characteristics enable the application of Theorems 1 and 2 with respect to Figure 6 and then discuss their implications.

One can view the factory in Figure 6 as composed of four sub-systems: two conveyors, washer assembly and ball-bearing assembly, which are described separately using Δ_1, Δ_2, Δ_3 and Δ_4 respectively. Given that these sub-systems interact only through sensors and actuators (Property 1 above), and given that any system observation ϕ will include these sensors and actuators (Property 2 above), one can use Theorem 1 to compute the minimal conflicts of the whole system, $\mathit{MinConflicts}^{\Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4}(\phi)$, in terms of the minimal conflicts of sub-systems, $\mathit{MinConflicts}^{\Delta_1}(\phi_{\Delta_1})$, $\ldots$, $\mathit{MinConflicts}^{\Delta_4}(\phi_{\Delta_4})$.

Theorem 1 sanctions the independent processing of discrete-event sub-systems since the interaction between these sub-systems is restricted to sensors and actuators. Minimal conflicts and, hence, diagnoses, of the whole system can then be obtained by simply combining the minimal conflicts computed with respect to these sub-systems. This shows that the diagnosis of discrete-event systems scales up as long as the newly added sub-systems are interfaced to the current system using only sensors and actuators (which is common practice). This is also a key explanation of why the diagnosis of such systems is more manageable than one would expect, as observed in (Darwiche & Provan 1996).

One important point to note, however, is that, since discrete-event systems are dynamic, their descriptions contain *temporal* propositional sentences. That is, if atom A stands for an actuator, then we would have atoms $A_1, A_2, \ldots, A_t$ in the system description, standing for the actuator at time points $1, 2, \ldots, t$. Suppose now that we model the system over some t time steps, and then encounter a problem at time $t = 5$. In this case, we do not have the state of sensors and actuators beyond time 5 and, therefore, cannot apply Theorem 1 directly. This is the place where Theorem 2 plays a key role. In particular, the successive application of this theorem allows us to prune all sentences that describe the system behavior beyond time 5, eliminating any mention of sensors and actuators after this time and allowing the direct application of Theorem 1 as described above.

We close this section by stressing two points. First, the detailed application of our theorems to discrete–event systems is best illustrated in the context of structured system descriptions, which we omit here for space limitations, but can be found in (Darwiche & Provan 1996). Second, Theorems 1 and 2 do not fully explain the reason for our success in efficiently diagnosing discrete–event systems: the structure–based approach used in (Darwiche & Provan 1996) does have further merits that contribute to its effectiveness with discrete–event systems. Discussing these other merits is outside the scope of this paper but can be found in (Darwiche & Provan 1996) and (Darwiche 1995a).

Related Work

Within MBD, several researchers have investigated the role of hierarchical decomposition (Genesereth 1984; Hamscher 1990), system independence (Tsybenko 1995; Freitag & Friedrich) and localization (Xiang, Poole, & Beddoes 1993) in improving diagnostic efficiency. A novel contribution of this paper is a formalizion of how we can improve diagnostic efficiency by decomposing the system using observations. Along the lines of system decomposition for diagnostic reasoning, the work of Tsybenko (1995) is of particular relevance. Our framework extends Tsybenko's analysis to provide a formal description of independence that provides a coherent description of component independence and independence induced by observations.

Exploiting observations to efficiently diagnose discrete–event systems is a novel contribution, as few researchers have applied model–based approaches to the diagnosis of such systems. A notable exception is the work of Sampath et al. (1995), who diagnose discrete–event systems formalized using finite automata. Although Sampath et al. adopt a compositional approach to system descriptions, their framework does not have a formal notion of independence, and hence cannot employ component independence and independence induced by observations for the purposes of system decomposition.

Conclusions

This paper has shown how to use observations to efficiently diagnose complex systems that have certain properties. In particular, we presented two theorems concerning the effect of observations on the complexity of MBD: the first theorem shows how the presence of certain observations allows us to decompose diagnostic reasoning into reasoning on subsystems, and the second theorem shows how the absence of certain observations allows us to ignore parts of a system during diagnostic reasoning.

The second main contribution of this paper is an application of these theorems to the diagnosis of discrete–event systems. Discrete–event systems are structured and highly observable, two features that makes them amenable to the presented theorems. These properties of discrete–event systems, and the presented theorems, partially explain why a particular approach that we proposed elsewhere has proven effective against this class of systems.

References

Darwiche, A., and Provan, G. 1996. Exploiting system structure in model–based diagnosis of discrete–event systems. In *Proceedings of the Seventh International Workshop on Principles of Diagnosis*, 95–105.

Darwiche, A. 1995a. Model–based diagnosis using structured system descriptions. Technical Report 96-69, Rockwell Science Center, Thousand Oaks, Ca.

Darwiche, A. 1995b. Model-based diagnosis using causal networks. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 211–217.

de Kleer, J.; Mackworth, A. K.; and Reiter, R. 1992. Characterizing diagnoses and systems. *Artificial Intelligence* 56(2-3):197–222.

Freitag, H., and Friedrich, G. Focusing on Independent Diagnosis Problems. In *Proc. Conf. on Principles of Knowledge Representation and Reasoning.* Morgan-Kaufmann Publishers.

Genesereth, M. R. 1984. The use of design descriptions in automated diagnosis. *Artificial Intelligence* 24:411–436.

Hamscher, W. 1990. Diagnosing Devices with Hierarchical Structure and Known Component Failure Modes. In *Proc. Conf. on AI Applications*, 48–54.

Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; and Teneketzis, D. 1995. Diagnosability of Discrete-Event Systems. *IEEE Trans. on Automatic Control* 40(9):1555–1575.

Tsybenko, Y. 1995. Decomposition into Independent Diagnosis Subproblems. In *Proc. Workshop on Principles of Qualitative Reasoning*, 173–180.

Xiang, Y.; Poole, D.; and Beddoes, M. 1993. Multiply-sectioned Bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence* 9(2):171–220.