Diagnosis with Incomplete Models: Diagnosing Hidden Interaction Faults

Lukas Kuhn [1], **and Johan de Kleer** [1]

[1] *Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304, USA*
lukas.kuhn@parc.com
dekleer@parc.edu

ABSTRACT

This paper extends model-based diagnosis (MBD) (de Kleer and Williams, 1987; Reiter, 1987) to systems with hidden interaction faults. An interaction fault is present if an interaction among a set of components leads to an observable failure, even though each individual component individually meets the specifications. A naive approach to address interaction faults is to simply account for all possible interaction faults in the system model. However, the naive approach presumes that all possible faults, both component and interaction faults, are known and addressed in the model. This assumption is violated by most real world systems, such as shorts in circuits (Davis, 1984) or unmodeled connections (de Kleer, 2007). That leads to incomplete system models, hence possibly hidden interaction faults. The problem of hidden interactions has been known for a long time (Davis, 1984), but until now no general solution has been proposed. Instead of pushing for complete models (Preist and Welham, 1990) or relying on additional structural information (Davis, 1984; Bottcher, 1995; de Kleer, 2007) we approach the challenge differently. We allow system models to be incomplete and introduce a general, domain independent extension to model-based diagnosis to account for resulting hidden interaction faults. This extends model-based diagnosis to systems with incomplete models, in particular to models with incomplete structural information. In the paper, we demonstrate the proposed diagnosis framework on a logic circuit with a hidden interaction fault.

1 INTRODUCTION

Model-based diagnosis assumes that all necessary information, regarding all possible failure causes, is available in the system model. In our experience, this generally accepted assumption does not hold in practice. In reality, systems fail for all kind of reasons, some of which designers might not be able to predict at the time the system model is built. This leads to incomplete models and to possible hidden interaction faults. For example, during the landing maneuver of the Mars Polar Lander an interaction between a touch sensor and the deployment of one of the Lander's legs most likely caused the mission to fail (Young *et al.*, 2000). The deployment of the leg caused the touch sensor to produce a noise spike which was incorrectly classified as an indication of touch down. As a consequence, the lander shut off its thrust about 40 meters above the touch-down surface. This is a classic example of a failure caused by a hidden interaction. If the engineers could have predicted this interaction, the failure could have been avoided. The classification algorithm could have requested either additional information from the altitude sensor onboard the lander or a time persistent signal from the touch sensor. Building adequate models for increasingly complex systems, especially for embedded systems, is very difficult; building complete models is practically impossible. In practice most models are incomplete, especially when all possible interactions are not known at the time the system is built.

Unlike a behavior model, which might describe the behavior only partially, e.g. weak fault model (describes only nominal behavior), the structural model is usually assumed to be complete (Davis, 1984; Preist and Welham, 1990; Bottcher, 1995; de Kleer, 2007). An incomplete system topology, e.g. a model that doesn't capture all connections, causes standard diagnosis frameworks to result in an irresolvable contradiction.

Instead of pushing for complete models, we approach the challenge differently. We allow models to be incomplete and introduce a diagnosis framework that works with incomplete models, extending model-based diagnosis to systems with hidden interactions. We account for interaction faults without explicitly modeling them. The resulting approach enables diagnosis for systems with multiple, interaction faults.

The paper is organized as follows: We introduce a logic circuit, *SMALLY*, which serves as our example system to review standard model-based diagnosis and illustrate its limitations. Then, we define interaction

faults formally and introduce a general extension to model-based diagnosis to account for hidden interaction faults.

2 RELATED WORK

The general mechanisms of inferring health states from observations have a long history in artificial intelligence and engineering including logic based frameworks (Reiter, 1992), continuous non-linear systems (Rauch, 1995), xerographic systems (Zhong and Li, 2000), and hybrid logical probabilistic diagnosis (Poole, 1991).

The process of diagnosis can be viewed as the interaction between observations and predictions. Observations capture the actual system behavior, whereas predictions are deduced from the system model. Model-based diagnosis presumes a system failure to be present if predictions and observations differ from each other.

Model-based approaches (de Kleer and Williams, 1987; Reiter, 1987) predict component interaction only where these are explicitly provided in the system description. The problem of faults caused by hidden interactions has been known since (Davis, 1984). In (Davis, 1984) bridge faults between adjacent components are introduces, but the suggested solution requires explicit knowledge about which unintended connections potentially result from adjacent components. In (Preist and Welham, 1990) a solution, similar to the naive approach, is proposed which explicitly models all possible unintended interactions. We support the argument that a complete model is preferable over a incomplete model, but note that a complete model might not always be available. In (Bottcher, 1995) the work of (Davis, 1984) is generalized by introducing a notion of neighbors that requires information about spatial proximity among components. We are not aware of a diagnosis framework that accounts for hidden interaction faults by a general, domain independent extension without relying on additional domain dependent knowledge. All approaches listed in the related work section, (Davis, 1984; Preist and Welham, 1990; Bottcher, 1995; de Kleer, 2007) assume that additional knowledge regarding potential hidden interaction is available. In this paper, we introduce an approach to diagnosing hidden interactions, that does not rely on any kind of additional information such as knowledge about potential unintended connections or spatial proximity among components.

3 REVIEW OF MODEL-BASED DIAGNOSIS

Consider the logic circuit, *SMALLY*, illustrated in Figure 1. The system can be described by a component set and a system description. The system description specifies the behavior of the individual components and how the components interact with each other. In order to perform diagnosis we include observations in the system definition. Formally, we define a system by:

Definition 1. *A observable system is a triple* $(SD, COMPS, OBS)$ *where*

- SD*, system description, is a set of first-order sentences,*

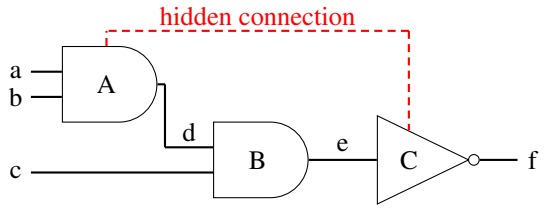

Figure 1: Example circut, SMALLY, with two and-gates and one inverter-gate.

- $COMPS$*, components, is a set of constants,*
- OBS*, observations, is a set of first-order sentences.*

Typically, the system description SD organizes the knowledge by maintaining a component library CL and a system topology ST. Generally, we can not assume that a system description SD is organized in any particular way, but we can assume that the SD intends to capture the behavior and the structure of the system. Note, that we say 'intends', as a system description might be incomplete.

Our example *SMALLY* contains two and-gates A, B and one inverter C. In Figure 1 solid lines illustrate connections captured by the topological model and dashed lines connections not captured by the topological model. The circuit *SMALLY* has a hidden connection between component A and C, as indicated by the dashed line. The connection is hidden to the diagnosis framework, as it doesn't appear in the model. The model has no knowledge of the behavior of the hidden connection. We assume that the actual behavior of the system will always deviate from the nominal behavior iff both component A and C are used together. A real world scenario could be that both components are late in propagating their signal, yet both are within specification. As a result, both tested individually will lead to no observable failure. If and only if we test both together and the delay accumulates can a failure be observed. In our example we assume that there are no intermittent faults. This is not a general restriction to our framework but makes the example more comprehensible.

In our example the set of components $COMPS$ consists of the three gates shown in Figure 1, thus $COMPS = \{A, B, C\}$. The system topology ST, is shown in Equation 1. Note, that the connection between component A and component C is not mentioned in the system topology, as we assume the connection is hidden. Thus the topological model is incomplete.

$$\begin{aligned} ST \quad = \{&And(A) \wedge And(B) \wedge Inv(C), \\ &a \equiv in(A,1) \wedge b \equiv in(A,2) \wedge out(A) \equiv d, \\ &d \equiv in(B,1) \wedge c \equiv in(B,2) \wedge out(B) \equiv e, \\ &e \equiv in(C,1) \wedge out(C) \equiv f\} \end{aligned} \tag{1}$$

To indicate the health state of an component we define the concept of an AB-literals and use them to formalize the component behavior in the component library CL.

Definition 2. *Let an AB-literal indicate the health of a component $x \in COMPS$. An AB-literal can be either $AB(x)$ or $\neg AB(x)$, where $AB(x)$ represents that component x is ABnormal (faulted) and $\neg AB(x)$ indicates that x is not ABnormal, thus normal.*

The component library CL describes the behavior of the individual components.

$$\begin{aligned} CL \quad = \{&And(x) \rightarrow [\neg AB(x) \rightarrow [in(x,1) \wedge in(x,2) \equiv out(x)]], \\ &Inv(x) \rightarrow [\neg AB(x) \rightarrow [in(x,1) \equiv \neg out(x)]]\} \end{aligned} \tag{2}$$

The system description SD is the union of the component library CL and the system topology ST, as shown in Equation 3.

$$SD \quad = \quad CL \cup ST \tag{3}$$

A complete assignment over all components or respectively over all corresponding health AB-literals, to either abnormal or not abnormal, is called a health assignment. A special case is the assignment that assigns not abnormal to all AB-literals, denoted $\neg AB^*$. The $\neg AB^*$ is defined in Definition 3.

Definition 3. *The $\neg AB^*$ assigns not abnormal to all AB-literals. Formally,*

$$\neg AB^* \quad = \quad \{\bigwedge_{c \in COMPS} \neg AB(c)\}. \tag{4}$$

In the absense of failures, the $\neg AB^*$ together with the system description SD and the actual observations should be consistent, as defined in Definition 4.

Definition 4. *A set of observations OBS is consistent with the system description iff the following sentence is satisfiable:*

$$SD \cup OBS \cup \neg AB^* \tag{5}$$

Let's assume we collect observation obs_1 which involves both components, A and C, for example we observe a, b, c, and f,

$$obs_1 \quad = \quad [a \equiv 1 \wedge b \equiv 1 \wedge c \equiv 1] \rightarrow f \equiv 1. \tag{6}$$

Given observation obs_1 and the system description SD, we can evaluate if the predicted behavior is consistent with what we observed. In our example the predicted behavior is not consistent with the actual observation obs_1. The system description SD together with $\neg AB^*$ imply that $a \equiv b \equiv c \equiv 1$ that $d \equiv 1$, $e \equiv 1$, and $f \equiv 0$. The predicted value for f is therefore 0, but the actually observed value is 1. The difference is called a discrepancy. Based on system description SD and observation obs_1 we can infer a conflict, according to Definition 6.

Definition 5. *An AB-clause is a disjunction of AB-literal containing no complementary pair of AB-literals.*

Definition 6. *A conflict of $(SD, COMPS, OBS)$ is an AB-clause entailed by $SD \cup OBS$.*

The resulting conflict in our example is

$$SD \cup \{obs_1\} \vdash AB(A) \vee AB(B) \vee AB(C). \tag{7}$$

The diagnosis task is to find health assignments that make SD and OBS consistent. Formally, a diagnosis is defined by Definitions 7 and 8.

Definition 7. *Given two sets of components, C_{AB} and $C_{\neg AB}$, we define $D(C_{AB}, C_{\neg AB})$ to be the conjunction:*

$$\left[\bigwedge_{c \in C_{AB}} AB(c)\right] \wedge \left[\bigwedge_{c \in C_{\neg AB}} \neg AB(c)\right] \tag{8}$$

where $AB(x)$ corresponces to the AB-literal of x.

Definition 8. *A diagnosis Δ for $(SD, COMPS, OBS)$ is a subset of the component set, formally $\Delta \subseteq COMPS$, such that the following set of sentences is satisfiable*

$$SD \cup OBS \cup \{D(\Delta, COMPS - \Delta)\} \tag{9}$$

Definition 9. *The cardinality of a diagnosis Δ, denoted $|\Delta|$, is to the number of elements in Δ.*

The list in Equation 10 shows all valid diagnoses based on observation obs_1 ordered by cardinality. The cardinality of a diagnosis is defined in Definition 9.

$$\begin{aligned} &\textit{single fault diagnoses:} \\ &\Delta_1 = \{A\}, \quad \Delta_2 = \{B\}, \quad \Delta_3 = \{C\}, \\ &\textit{double fault diagnoses:} \\ &\Delta_4 = \{A,B\}, \quad \Delta_5 = \{A,C\}, \quad \Delta_6 = \{B,C\}, \\ &\textit{triple fault diagnoses:} \\ &\Delta_7 = \{A,B,C\} \end{aligned} \tag{10}$$

We can reduce the set of diagnoses by a more constrained definition of diagnosis, coined minimal cardinality diagnosis, defined in Definition 10.

Definition 10. *A diagnosis Δ for $(SD, COMPS, OBS)$ is a minimal cardinality diagnosis if and only if there exists no other diagnosis Δ' such that $|\Delta'| < |\Delta|$.*

The list in Equation 11 shows the set of minimal cardinality diagnoses. The minimal cardinality among all diagnoses is 1 yet we can't conclude that there is only one failure in the system. The only conclusion we can draw is that there is at least one failure in the system.

$$\begin{aligned} &\textit{single fault diagnoses:} \\ &\Delta_1 = \{A\}, \quad \Delta_2 = \{B\}, \quad \Delta_3 = \{C\} \end{aligned} \tag{11}$$

Let's say we collect another observation obs_2:

$$obs_2 \quad = \quad [a \equiv 1 \wedge b \equiv 1] \rightarrow d \equiv 1. \tag{12}$$

Based on the two observations collected, obs_1 Equation 6 and obs_2 Equation 12, we can deduce that a fault in component A individually can not explain the discrepancy. Recall, we assumed non-intermittent faults. This reduces the set of minimal cardinality diagnoses to the list:

$$\begin{aligned} &\textit{single fault diagnoses:} \\ &\Delta_2 = \{B\}, \quad \Delta_3 = \{C\}, \end{aligned} \tag{13}$$

To illustrate the limitations of prior diagnosis frameworks, we assume that we collect another two observations obs_3 and obs_4, shown in Equation 15.

$$\begin{aligned} obs_3 \quad &= \quad [d \equiv 1 \wedge c \equiv 1] \rightarrow e \equiv 1 \\ obs_4 \quad &= \quad [e \equiv 1] \rightarrow f \equiv 0 \end{aligned} \tag{14}$$

Based on the observations obs_1, obs_2, and obs_3, we can conclude that neither component A individually

nor component B individually can explain the discrepancy. Once we expand our reasoning to include all available observations, obs_1, obs_2, obs_3, and obs_4, the diagnosis framework results with an irresolvable contradiction. But why is that? Let's take a closer look at our observations, obs_1, obs_2, obs_3, and obs_4. We can re-write the observations as shown in Equation 16.

$$\begin{aligned} obs_2 &= [a \equiv 1 \wedge b \equiv 1] \rightarrow d \equiv 1 \\ obs_3 &= [d \equiv 1 \wedge c \equiv 1] \rightarrow e \equiv 1 \\ [obs_2 \wedge obs_3] &\rightarrow [a \equiv 1 \wedge b \equiv 1 \wedge c \equiv 1] \rightarrow e \equiv 1 \\ obs_4 &= [e \equiv 1] \rightarrow f \equiv 0 \\ [obs_2 \wedge obs_3 \wedge obs_4] &\rightarrow [a \equiv 1 \wedge b \equiv 1 \wedge c \equiv 1] \rightarrow f \equiv 0 \\ obs_1 &= [a \equiv 1 \wedge b \equiv 1 \wedge c \equiv 1] \rightarrow f \equiv 1 \end{aligned} \quad (15)$$

We can now see that observation obs_1 is in an irresolvable contradiction to the observations obs_2, obs_3, and obs_4. This inconsistency is independent of the health assignment we choose. As there exists no health assignment that makes the observations consistent, we can infer that there exists no diagnosis for the system. Generally, we can define the existence of a diagnosis as:

Definition 11. *A diagnosis exists for* $(SD, COMPS, OBS)$ *iff* $SD \cup OBS$ *is consistent.*

Standard model-based diagnosis will terminate with an irresolvable contradiction, if no diagnosis can be found. Formally, from Definition 11 it follows that there exists no diagnosis for $(SD, COMPS, OBS)$ iff $SD \cup OBS$ is inconsistent. Hence, in our example, no diagnosis can be found, and standard model-based diagnosis terminates with an error.

4 LIMITATIONS OF STANDARD MODEL-BASED DIAGNOSIS

In the previous section we illustrated the limitations of standard model-based diagnosis. The limitations are caused by the assumption that an accurate and complete model is available. In a real world scenario this is rather impractical. Building models is a time demanding, expensive process. Therefore, most system models are limited to enable detection or isolation of a small pre-defined set of failures. Typically, a system model captures only the knowledge required to diagnosis this pre-defined set of failures and abstracts all other information away.

For example, the automobile industry adopted on-board diagnosis for cars, but only targeted to specific subsystems and failure modes. Some cars have the capability to diagnosis if a head light bulb is out, but fail if the connecting cable is broken. A broken cable occurs so infrequently, that most diagnosis designer neglect that a cable might break and abstract it away (de Kleer, 2007). If the cable does break, the initial diagnosis might suggest that one of the two connected components is faulted. Once the two components are individually tested without noticeable abnormality the diagnosis framework either incorrectly concludes an intermittent fault in one of the two components (if the framework is aware of this fault type) or terminates with an irresolvable error. The irresolvable error results from the unawareness of the connection. The connection is hidden to the diagnosis framework and all other components are exonerated as fault candidates. The diagnoser results with an empty list of diagnosis candidates, yet has observed a discrepancy. This results in an irresolvable contradiction.

Another reason for incomplete system models is due to model recycling, the act of reusing an already existing model. Building models is an expensive and time demanding task, which makes model recycling attractive. Typically, there are two kinds of sources for reusable models: Either there exists a similar system, similar enough to adapted its model or there exists a model for the target system which was originally built for a different task, e.g. system based on a planning (Kuhn *et al.*, 2008) or scheduling (Muscettola *et al.*, 1998) model.

The set of failures desired to be diagnosable as well as the intended repairs influence the scope and abstraction level of the resulting system model. For example, a vendor that only performs repair by exchanging entire subsystems might neglect fault isolation on the component level as it is not necessary for the repair. This leads to abstract system models targeted towards a specific diagnosis task. In our car example, diagnosis is targeted to find the most common failures (e.g. broken light bulb), but results with an irresolvable contradiction if a component outside of the model scope causes the abnormality, e.g. a broken cable. Such models violate the no-function-in-structure principle.

5 MODEL-BASED DIAGNOSIS WITH INTERACTION FAULTS

We propose a diagnosis framework that is able to diagnose component faults as well as hidden interaction faults. Our approach builds on the assumption, that there is no additional knowledge besides what's already captured by the model. Hence all available knowledge is already built into the model, yet the model might still be incomplete. This leads to a new fault type: faults caused by hidden interactions, coined interaction faults. An interaction fault is present, if an interaction among a set of components leads to an observable failure, even though each individual component individually meets the specification. Hidden interactions can lead to interaction faults. We distinguish between the following two kinds of hidden interactions:

- A *hidden component interaction* is present if a set of known components interact through a hidden component. The component is hidden in the sense that it doesn't appear in the model. A common example for hidden components are connections in circuits (de Kleer, 2007). Most system models abstract connections away. Typically, a fault in a connection initially results in the belief that one of the two connected components is faulted. Once the two components are individually tested without noticeable abnormality the diagnosis framework either incorrectly concludes an intermittent fault (if the framework is aware of this fault type) or terminates with an irresolvable contradiction.
- A *hidden behavior interaction* is present iff the interaction between a set of components leads to unpredicted behavior. Consider a food processing line for candy bars. There are multiple components wrapping and boxing candy bars. It may be that component A leaves a tiny rip which is of no consequence for the consumer, but boxing component B has a small protrusion such that the rip sometimes catches and destroys the candy bar. We call such faults hidden behavior interaction faults: A and B are perfectly operational individually but will not work correctly if A and B operate together. Such faults also occur in circuits: Two gates A and B may not work well together as the accumulated delay

leads to a failure. Testing both components individually might convey that both are late, yet within specification. (Some call this bad design, but most complex systems have design errors.)

In the following section we introduce a diagnosis framework for systems containing interaction faults. First, we define such systems and discuss an extension to standard model-based diagnosis such that the diagnosis framework accounts for both individual component faults as well as hidden interaction faults. The extension is generally applicable without additional system knowledge. Definition 12 defines a model-based diagnosis system with hidden interactions:

Definition 12. *A model-based diagnosis system with hidden interactions is represented as a quadruple* $(SD+, COMPS, OBS+, SCOPE)$ *where:*

- $SD+$*, extended system description, is a set of first-order sentences,*
- $COMPS$*, components, is a set of constants,*
- $OBS+$*, extended observations, is a set of first-order sentences,*
- $SCOPE$*, scope of an observation, is a function mapping an observation onto a subset of* $COMPS$*.*

The extended system description $SD+$ extends the standard system description SD, defined in Definition 1, to account for potential hidden interactions. Similar to before, the $SD+$ contains knowledge about the behavior and the structure of the system, but admits the possibility of a hidden interaction. Related approaches (Davis, 1984; Bottcher, 1995; de Kleer, 2007) have suggested to extend the SD by explicitly modeling potential hidden interactions by using additional knowledge about the system. The extension we suggest is domain independent, hence any system is extended with the same extension. This enables our approach to be widely applicable, even if potential hidden interactions are not known at design time. Before we introduce the extension, we need to modify the definition of AB-literals. We denote the modified literals as AB^i-literals:

Definition 13. *Let* $Pow(COMPS)$ *be the set of all subsets of* $COMPS$ *such that*

$$Pow(COMPS) = \{pc \mid pc \subseteq COMPS\}, \tag{16}$$

let $pc \in Pow(COMPS)$ *indicate an interaction among all components in* pc *and let all* $AB^i(x) = AB(x)$*. An* AB^i*-literal indicates the health of* $pc \in Pow(COMPS)$ *and can be either* $AB(pc)$ *or* $\neg AB(pc)$*, where* $AB(pc)$ *represents that* pc *is ABnormal (faulted) and* $\neg AB(pc)$ *indicates that* pc *is not ABnormal, thus behaving normal.*

Based on the standard system description SD, we can construct the extended system description $SD+$ simply by first adding the model extension ME shown in Equation 17 and secondly by replacing all AB-literals with the corresponding single component AB^i-literals.

$$ME = \bigcup_{pc \in Pow(COMPS)} AB^i(pc) \rightarrow \left[\bigwedge_{pc' \subset pc} \neg AB^i(pc') \right] \tag{17}$$

The extension semantically adds two aspects to a system description. First, it introduces AB^i-literals for pc's with higher cardinality and second it introduces the relation among individual AB^i-literals. AB^i-literals for higher cardinality pc's account for unmodeled interactions which might occur between components $c \in pc$. This guides the diagnosis framework to detect and isolate abnormalities even if they are caused by hidden interactions. In case we diagnosis some pc to be abnormal, sentence 17 enforces that all subsumed pc''s are diagnosed to be not abnormal. The second aspect is important as it only makes sense to hypothesize about an interaction fault if all hypotheses of subsumed individual component faults as well as interaction faults are exonerated.

We can formalize the extended system description for our example *SMALLY* as shown in Equation 18. The hidden interaction is indicated as a dashed connection in Figure 1.

$$\begin{aligned}
SD &= CL \cup ST \cup ME \text{ where} \\
CL &= \{And(x) \rightarrow \\
&\quad [\neg AB(\{x\}) \rightarrow [in(x,1) \wedge in(x,2) \equiv out(x)]] \\
&\quad Inv(x) \rightarrow [\neg AB(\{x\}) \rightarrow [in(x,1) \equiv \neg out(x)]]\} \\
ST &= \{And(A) \wedge And(B) \wedge Inv(C), \\
&\quad a \equiv in(A,1) \wedge b \equiv in(A,2) \wedge out(A) \equiv d, \\
&\quad d \equiv in(B,1) \wedge c \equiv in(B,2) \wedge out(B) \equiv e, \\
&\quad e \equiv in(C,1) \wedge out(C) \equiv f\} \\
ME &= \{AB^i(\{A,B\}) \rightarrow [\neg AB(\{A\}) \wedge \neg AB(\{B\})], \\
&\quad AB^i(\{A,C\}) \rightarrow [\neg AB(\{A\}) \wedge \neg AB(\{C\})], \\
&\quad AB^i(\{B,C\}) \rightarrow [\neg AB(\{B\}) \wedge \neg AB(\{C\})], \\
&\quad AB^i(\{A,B,C\}) \rightarrow \\
&\quad [\neg AB(\{A\}) \wedge \neg AB(\{B\}) \wedge \neg AB(\{C\}) \wedge \\
&\quad \neg AB(\{A,B\}) \wedge \neg AB(\{A,C\}) \wedge \neg AB(\{B,C\})]\}
\end{aligned} \tag{18}$$

The extended $SD+$ doesn't define any relations between possible observations and interaction faults. A good technician can infer interaction faults from observations. Let's say a technician tests two components individually and observes no abnormality, but if both components are tested together the observations indicate an abnormality. The technician would draw the conclusion that there might exist a hidden interaction. To enable a diagnosis framework to perform the same kind of inference, we define two things: First, we define the scope of an observation, basically what is being tested together. Second, we incorporate the scope into the observations, to indicate which observation is relevant to which interaction fault. Given an observation, the concept of an observation scope defines the set of components that has potentially impacted this observation. Once the function $SCOPE$ is defined, we can extend a set of observations OBS to a set of extended observations $OBS+$ according to:

$$OBS+ = \left\{ [obs] \vee \bigvee_{pc \subseteq Scope(obs)} AB^i(pc) \mid |pc| > 1, obs \in OBS \right\} \tag{19}$$

Generally, the function $SCOPE$ can be extracted from the system description without relying on additional information. An observation is a set of measurement points. The system structure combined with the component behavior provides information in order to determine which set of components has potential impact on which set of measurement points. By backward reasoning, we can extract the scope of an observation. In our example, we informally define the scope of an observation as the components that had potentially impacted the resulting measurement point. For example an observation measuring at point a and e scopes

over component A and B, an observation measuring at point a, b, and f scopes over the components A, B, C and an observation measuring only a and b scopes over no component as the signal neither travels from a to b nor from b to a. The scope for the observations in our example are illustrated in Listing 21.

$$\begin{aligned} SCOPE(obs_1) &= \{A, B, C\} \\ SCOPE(obs_2) &= \{A\} \\ SCOPE(obs_3) &= \{B\} \\ SCOPE(obs_4) &= \{C\} \end{aligned} \tag{20}$$

Given the observation scopes we can construct the extended observation based the extension rule shown in Equation 19. The resulting extended observations are shown in Listing 22.

$$\begin{aligned} obs_1^i &= [obs_1] \vee AB^i(\{A, B\}) \vee AB^i(\{A, C\}) \vee \\ &\quad AB^i(\{B, C\}) \vee AB^i(\{A, B, C\}) \\ obs_2^i &= obs_2 \\ obs_3^i &= obs_3 \\ obs_4^i &= obs_4 \end{aligned} \tag{21}$$

We have extended the system definition and introduced AB^i-literals with the intent to diagnose hidden interaction faults. Definition 7 defines a diagnosis as set of components assigned to be abnormal such that the resulting assignment over all components makes the system description consistent with the observations. In this definition an assignment is limited to determine which individual component is considered to be abnormal or not abnormal. In order to account not only for individual components but also for hidden interactions we expand the assignment to be over all AB^i-literals. Individual components are captured by AB^i-literals with cardinality 1 and hidden interactions are addressed by AB^i-literals with higher cardinality. A diagnosis is an assignment of abnormal or not abnormal to all elements of $Pow(COMPS)$ describing one possible health state of the system. Formally, a diagnosis for systems with hidden interactions is defined in Definition 15.

Definition 14. *Given two sets C_{AB}, $C_{\neg AB} \subseteq Pow(COMPS)$, we define $D^i(C_{AB}, C_{\neg AB})$ to be the conjunction:*

$$\left[\bigwedge_{pc \in C_{AB}} AB_i(pc)\right] \wedge \left[\bigwedge_{pc \in C_{\neg AB}} \neg AB_i(pc)\right] \tag{22}$$

where $AB^i(x)$ corresponds to the AB^i-literal of x.

Definition 15. *A diagnosis Δ^i for $(SD+, COMPS, OBS+, SCOPE)$ is a subset of $Pow(COMPS)$, such that the following set of sentences is satisfiable*

$$SD \cup OBS \cup \left\{ D^i(\Delta, Pow(COMPS) - \Delta) \right\} \tag{23}$$

The Listing 24 shows all valid diagnoses for $SMALLY$ given that we observed only observation obs_1^i.

single fault diagnoses:
$\Delta_1^i = \{\{A\}\}$, $\Delta_2^i = \{\{B\}\}$,
$\Delta_3^i = \{\{C\}\}$, $\Delta_4^i = \{\{A, B\}\}$,
$\Delta_5^i = \{\{A, C\}\}$, $\Delta_6^i = \{\{B, C\}\}$,
$\Delta_7^i = \{\{A, B, C\}\}$
double fault diagnoses:
$\Delta_8^i = \{\{A\}, \{B\}\}$, $\Delta_9^i = \{\{A\}, \{C\}\}$,
$\Delta_{10}^i = \{\{B\}, \{C\}\}$, $\Delta_{11}^i = \{\{A\}, \{B, C\}\}$,
$\Delta_{12}^i = \{\{A, B\}, \{C\}\}$, $\Delta_{13}^i = \{\{B\}, \{A, C\}\}$
triple fault diagnoses:
$\Delta_{14}^i = \{\{A\}, \{B\}, \{C\}\}$ (24)

Similar to Definition 10, we can reduce the set of diagnoses by adapting the concept of minimal cardinality diagnoses, as illustrated in Definition 16.

Definition 16. *A diagnosis Δ_x^i for $(SD+, COMPS, OBS+, SCOPE)$ is minimal in cardinality if and only if there exists no other diagnosis Δ_y^i such that $|\Delta_y^i| < |\Delta_x^i|$.*

The minimal cardinality diagnoses resulting from observation obs_1^i are all single fault diagnoses in Listing 16. We can further reduce the set by an even more strict definition of minimality, coined a minimal cardinality, minimal interaction diagnosis.

Definition 17. *A minimal cardinality diagnosis Δ_x^i for $(SD+, COMPS, OBS+, SCOPE)$ is also a minimal cardinality, minimal interaction diagnosis if and only if there exists no other diagnosis Δ_y^i such that an element in $|\Delta_y^i|$ is a strict subset of any element in $|\Delta_x^i|$.*

The resulting set of minimal cardinality, minimal interaction diagnoses, given that we observed observation obs^i, is illustrated in Listing 25.

minimal cardinality, minimal interaction diagnoses:
$\Delta_1^i = \{\{A\}\}$, $\Delta_2^i = \{\{B\}\}$, $\Delta_3^i = \{\{C\}\}$ (25)

Listing 25 shows the set of minimal cardinality, minimal interaction diagnoses assuming only observation obs_1^i is available. Let's say we include observation obs_2^i such that the resulting set of diagnoses has to be consistent with both observations, obs_1^i and obs_2^i. We can deduce that a fault in component A individually can not explain the discrepancy. Therefore the resulting set of minimal cardinality, minimal interaction diagnoses reduces to the once shown in Listing 26

minimal cardinality, minimal interaction diagnoses:
$\Delta_2^i = \{\{B\}\}$, $\Delta_3^i = \{\{C\}\}$ (26)

Let's say we continue and include observations obs_3^i. From all three observations, we deduce that a single fault in component A as well as a single fault in component B can not explain the discrepancy. The only explanation for the discrepancy is that there exists either a single fault in component C, an interaction fault or a multiple fault. As we are interested in the set of minimal cardinality, minimal interaction diagnoses the multiple faults will only be considered if all single faults are exonerated. This leaves us with the hypotheses of a single fault in component C or some kind of interaction fault. Let's look at the remaining set of diagnoses, shown in Listing 27, in more detail.

minimal cardinality diagnoses:

$$\begin{array}{llll} \Delta_3^i = & \{\{C\}\}, & \Delta_4^i = & \{\{A,B\}\}, \\ \Delta_5^i = & \{\{A,C\}\}, & \Delta_6^i = & \{\{B,C\}\}, \\ \Delta_7^i = & \{\{A,B,C\}\} & & \end{array} \tag{27}$$

The diagnoses in Listing 27 are all minimal cardinality diagnoses according to Definition 16, yet not all of them are also minimal cardinality, minimal interaction diagnosis. According to Definition 16 and the fact that diagnosis Δ_3^i is a valid diagnosis we can conclude that diagnoses Δ_5^i, Δ_6^i, and Δ_7^i are not considered to be minimal cardinality, minimal interaction diagnoses. All three contain at least one element x, such that $\{C\} \subset x$. The resulting set of minimal cardinality, minimal interaction diagnoses is shown in Listing 28.

minimal cardinality, minimal interaction diagnoses:

$$\begin{array}{llll} \Delta_3^i = & \{\{C\}\}, & \Delta_4^i = & \{\{A,B\}\} \end{array} \tag{28}$$

Considering all four observations $obs_1^i, obs_2^i, obs_3^i, obs_4^i$, we deduce that a single fault in component C can't explain the observations either. Diagnosis Δ_3^i is not longer a valid diagnosis. At this point the standard model-based diagnosis framework terminates with an irresolvable contradiction. The proposed framework generates the set of minimal cardinality, minimal interaction diagnosis shown in Listing 29.

minimal cardinality, minimal interaction diagnoses:

$$\begin{array}{llllll} \Delta_4^i = & \{\{A,B\}\}, & \Delta_5^i = & \{\{A,C\}\}, & \Delta_6^i = & \{\{B,C\}\} \end{array} \tag{29}$$

Our framework he does not result with an irresolvable contradiction if and only if at least one of the AB^i-literals shown in Listing 30 is assigned to $abnormal$. By assigning one of the interaction AB^i-literals to $abnormal$ observation obs_1^i evaluates independently of the assignment to all other AB^i-literals without conflict.

$$AB^i(\{A,B\}) \vee AB^i(\{A,C\}) \vee AB^i(\{B,C\}) \vee AB^i(\{A,B,C\}) \tag{30}$$

Definition 18. *A system is diagnosed to contain multiple faults iff a minimal cardinality diagnosis Δ^i contains more than one element.*

Definition 19. *A system is diagnosed to contain an interaction fault iff a minimal cardinality, minimal interaction diagnosis Δ^i contains an element $x \in \Delta$ with more than one component.*

6 CONCLUSION

This paper has proposed a fundamentally new approach to address the very real issue that all system models are incomplete. Ensuring complete models is impossible. Through introducing interaction literals most kinds of unintended interactions can be accommodated within the model-based diagnosis framework. One of the main motivations behind this work arose from developing diagnostic algorithms for Xerographic equipment. Interaction faults are surprisingly common and are difficult for technicians to diagnose. They are also difficult to self-diagnose (more and more equipment includes self-diagnosis).

This paper has lays out our fundamental approach to hidden interaction faults. As with all model-based frameworks, it is computationally explosive if directly implemented as described in the definitions of this paper. In our implementation we employ a notion of diagnostic foci which introduces both $AB(x)$ and $AB^i(x)$ literals only when needed, i.e., extends ME only when needed. A direct translation of Equation 17 to all interaction faults of cardinality n leads to potentially $|COMPS|^n$ clauses. The implementation is a subject of another paper.

REFERENCES

(Bottcher, 1995) Claudia Bottcher. No faults in structure?: how to diagnose hidden interactions. In *IJCAI'95: Proceedings of the 14th international joint conference on Artificial intelligence*, pages 1728–1734, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

(Davis, 1984) Randall Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24(1):347–410, 1984.

(de Kleer and Williams, 1987) J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(32):97–130, 1987.

(de Kleer, 2007) J. de Kleer. Modeling when connections are the problem. In *Proc 20th IJCAI*, pages 311–317, Hyderabad, India, 2007.

(Kuhn *et al.*, 2008) Lukas Kuhn, Bob Price, Johan de Kleer, Minh Do, and Rong Zhou. Pervasive diagnosis: Integration of active diagnosis into production plans. In *proceedings of AAAI*, Chicago, Illinois, USA, 2008.

(Muscettola *et al.*, 1998) Nicola Muscettola, P. Pandurang Nayak, Barney Pell, and Brian C. Williams. Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1-2):5–47, 1998.

(Poole, 1991) D. Poole. Representing diagnostic knowledge for probabilistic horn abduction. In *International Joint Conference on Artificial Intelligence (IJCAI91)*, pages 1129–1135, 1991.

(Preist and Welham, 1990) C. Preist and B. Welham. Modelling bridge faults for diagnosis in electronic circuits. In *Proceedings of the First International Workshop on Principles of Diagnosis*, Stanford, 1990.

(Rauch, 1995) Herbert E. Rauch. Autonomous control reconfiguration. *IEEE Control Systems Magazine*, 15(6):37–48, 1995.

(Reiter, 1987) R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.

(Reiter, 1992) Raymond Reiter. A theory of diagnosis from first principles. In *Readings in Model-Based Diagnosis*, pages 29–48, 1992.

(Young *et al.*, 2000) Thomas Young, James Arnold, Thomas Brackey, Michael Carr, Douglas Dwoyer, Ronald Fogleman, Ralph Jacobson, Herbert Kottler, Peter Lyman, and Joanne Maguire. Mars program independent assessment team report. Technical report, Report to the NASA Administrator and to Congress, 2000.

(Zhong and Li, 2000) C. Zhong and P. Li. Bayesian belief network modeling and diagnosis of xerographic systems. In *Proceedings of the ASME Symposium on Controls and Imaging - IMECE*, Orlando, Florida, 2000.