

+

Studies in Logic, Language and Information

Managing editor: Robin Cooper, University of Edinburgh

Executive editor: Maarten de Rijke, CWI, Amsterdam

Editorial board:

Peter Aczel, Manchester University

Nicholas Asher, The University of Texas, Austin

Jon Barwise, Indiana University, Bloomington

John Etchemendy, CSLI, Stanford

Dov Gabbay, Imperial College, London

Hans Kamp, Universität Stuttgart

Godehard Link, Universität München

Fernando Pereira, AT&T Bell Laboratories, Murray Hill

Dag Westerståhl, Stockholm University

The *Studies in Logic, Language and Information* book series is the official book series of the European Association for Logic, Language and Information (FoLLI).

The scope of the book series is the logical and computational foundations of natural, formal, and programming languages, as well as the different forms of human and mechanized inference and information processing. It covers the logical, linguistic, psychological and information-theoretic parts of the cognitive sciences as well as mathematical tools for them. The emphasis is on the theoretical and interdisciplinary aspects of these areas.

The series aims at the rapid dissemination of research monographs, lecture notes and edited volumes at an affordable price.



A Great Collection

edited by
U. Gnowho
and U. Gnowho-Else



CSLI was founded early in 1983 by researchers from Stanford University, SRI International, and Xerox PARC to further research and development of integrated theories of language, information, and computation. CSLI headquarters and the publication offices are located at the Stanford site.

CSLI/SRI International	CSLI/Stanford	CSLI/Xerox PARC
333 Ravenswood Avenue	Ventura Hall	3333 Coyote Hill Road
Menlo Park, CA 94025	Stanford, CA 94305	Palo Alto, CA 94304

Copyright © 1995
Center for the Study of Language and Information
Leland Stanford Junior University

Printed in the United States

01 00 99 98 97 96 95 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

U. GnoWho

A Great Title / U. GnoWho

p. cm. — (Studies in Logic, Language and Information ; no. ??)

Includes bibliographical references and index.

ISBN 1-234567-89-0 (cloth) ISBN 1-234567-89-0 (pbk.)

1. Logic. 2. Language. 3. Information. I. Title. II. Series: Studies in Logic, Language and Information ; no. ??, etc.

BC5.S57 199?

160

90-82189
CIP



Contents

Contributors	vii
Preface	ix
1 The Consistency-based Approach to Automated Diagnosis of Devices	1
OSKAR DRESSLER, PETER STRUSS	



Contributors

O. DRESSLER is a managing director of OCC'M Software, a company offering software and consultancy in the area of knowledge-based systems. His research interests include non-monotonic reasoning, truth-maintenance systems, and model-based systems.

Current address: Sebastian-Bauer-Str. 35, D-81737 Munich, Germany.
E-mail: dressler@informatik.tu-muenchen.de

P. STRUSS is a private lecturer at the Computer Science Institute of the University of Technology in Munich and a managing director of OCC'M software. His main research interests are in model-based systems and qualitative reasoning. *Current address:* Computer Science Institute, University of Technology Munich, Orleansstr. 34, D-81667 Munich, Germany.
E-mail: struss@informatik.tu-muenchen.de

Preface

This is the preface for *A Great Title*.

On behalf of the organizing committee we want to thank the above-mentioned institutions for their support. We are indebted to Dikran Karagueuzian of CSLI Publications for his help during the preparation of this volume.

U. GnoWho, Amsterdam
U. GnoWho-Else, Stanford

The Consistency-based Approach to Automated Diagnosis of Devices

OSKAR DRESSLER, PETER STRUSS

ABSTRACT. This chapter surveys theories that provide principled approaches to automating the task of diagnosing broken artifacts and presents systems that implement these approaches. The key idea of model-based diagnosis is to explicitly represent the knowledge about a device as a model of the device structure and of the behavior of its constituents and to organize diagnosis as an inference process based on this model and the observed behavior. This approach created the demand for and the possibility of developing a rigorous theoretical foundation for automated diagnosis. In particular, this comprises a formal characterization of the goal and of the inferences that achieve the goal, given model-based predictions and the actual observations of the artifact's behavior. We argue that diagnosis is becoming a major field of application and an important touchstone for the utility of logical theories and AI.

1 Introduction

In this chapter we present the foundations of an area in automated problem solving which we feel to be exciting and important for several reasons. In a little more than ten years, work on automated diagnosis based on models has managed both to establish a strong theoretical basis and to create a technology mature enough to tackle real industrial applications. It is now well-positioned in the fringe where rigorous theories and requirements from real-world applications meet. This does not only allow us to build application systems with formally stated preconditions and provable capabilities and properties. It also provides challenges for theoretical work and hard criteria for evaluating its results and helps to focus it. In fact, we believe that model-based diagnosis can really become one of the rare success stories in artificial intelligence where a logical theory is the rigorous foundation for

A Great Collection
U. Gnowho and U. Gnowho-Else, eds.
Copyright © 1996, CSLI Publications.

automated reasoning systems that solve tough real world problems, such as localizing an unforeseen fault in an unexperienced device. The potential impact misdiagnoses and wrong situation assessments have on the safety of people and on the environment stresses the importance of such powerful diagnostic systems.

This development was possible after a critical assessment of the nature and limitations of “First Generation Diagnostic Expert Systems”, which were predominant in the seventies, with a major focus on medical diagnosis. These systems (with the blood infection diagnosis system MYCIN as the most famous instance) captured diagnostic skills by sets of more or less direct associations between observable symptoms and diseases as their potential causes (“IF symptom S THEN disease D with certainty C”). Being grounded in experience gained in previous cases, diagnosis was treated as collecting empirical evidence for the presence of certain malfunctions rather than a strict deductive process, providing no demand for a rigorous logical treatment. The necessity to state diagnostic knowledge in terms of explicit symptom-fault associations inherently limited the scope of applicability. Only the identification of previously encountered faults was possible based on previously observed symptoms of systems that are well experienced to allow for the enumeration of the relevant associations. Because these associations tend to be quite specific for narrow types of systems, building such systems was a matter of time-consuming single-piece production.

All this turned out to be too restrictive when confronted with requirements in diagnosis of technical devices. Industrial application of automated diagnosis has to cover the detection and localization of new kinds of (combinations of) faults, exhibited by newly designed and constructed systems and the interpretation of symptoms never observed before. And adaptation of a diagnostic system to a new variation of a device has to be easy, systematic, and reliable. It is pretty obvious that human capabilities to satisfy these requirements are not simply based on empirically derived symptom-fault associations. They stem from knowledge about the physical and technological principles underlying the (intended or deviating) functioning of an artifact which allows one to systematically deduce fault hypotheses from available observations even if the artifact is novel.

The key idea of model-based diagnosis is to explicitly represent this knowledge as a model of the device structure and of the behavior of its constituents and to organize diagnosis as an inference process based on this model and the observed behavior. This approach created the demand for and the possibility of developing a rigorous theoretical foundation for automated diagnosis. In particular, this comprises a formal characterization of the goal and of the inferences that achieve the goal, given model-based predictions and the actual observations of the artifact’s behavior. This is the major accomplishment of the theoretical work in the field and also is

one focus of the following presentation. It forms a necessary, but by no means sufficient condition for the effective and efficient computation of diagnoses for any interesting application. Controlling the inference process in an appropriate way is necessary to make it feasible and will be discussed, as well. In our view, this is one of the current challenges in the field, which must be tackled based on practical experience from complex real examples and must aim at clean extensions of the theory (rather than replacing it by informal heuristics).

Of course, the content and form of the system model is fundamental to the approach. Problems involved in modeling physical systems were mainly neglected in the first phase, often by treating examples of digital circuits at the level of logical gates such that the required behavior models could easily be accommodated by the logical formalism of the diagnostic theory. For the majority of applications, however, especially those involving dynamic systems, it is necessary to exploit other (mathematical) modeling formalisms in the logical framework. Due to limited space, we confine the discussion of these issues to the aspect of using multiple models in the diagnostic process and refer to the literature on qualitative modeling ([23], [15]) for a broader coverage. Work on appropriate modeling formalisms is crucial for further progress in the area. It will lead to more powerful theories and an extended scope of industrial applications, provided the field does not back off from the challenges of real problems by seeking hide either in elaboration of abstract theories or in unsystematic heuristic special-purpose systems.

In the following section, we present the theoretical foundations for the consistency-based approach to model-based diagnosis and a diagnostic system that localizes faults by exploiting models of correct behavior only. Section 3 extends the theory to include models of faulty behavior, as well, thus enabling the identification of particular faults. Techniques for controlling the diagnostic process and their theoretical foundations are discussed in section 4. Finally, open problems and suggested future work are pointed out.

2 Consistency-based Diagnosis

A human expert, such as a car mechanic, uses prior experience with faulty devices. Coming across a similar set of symptoms will trigger the memory of a previous diagnostic case. If such prior knowledge is available, it can be used as a shortcut. However, the mechanic is also capable of reasoning about the expected behavior of the car's subsystems at a much deeper level of physical laws, known as first principles, and 'pre-compiled' behavior of designed artifacts, second principles. Any deviation from the expected behavior he considers to be caused by a fault.

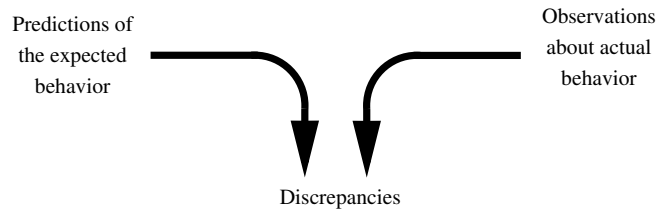


FIGURE 1 Discrepancy detection

The consistency-based approach to diagnosis takes exactly this view. A model of the device is used to predict its expected behavior which can be compared with the artifact's actual behavior (Figure 1). If discrepancies are detected, there is a diagnosis problem. The device was correctly designed and built, but now it exhibits indications of malfunctioning. The diagnostic task is to determine what is wrong in order to enable the re-establishment of the intended functionality. Possible answers may be that certain device components (or sets of them) are not working properly or that the device was affected in an even more serious way by changing its structure. In contrast to a broken wire, a bridge fault between two adjacent wires cannot be identified as a malfunctioning component or component set. This sounds difficult - and it is. Despite some attempts ([4], [18], [1]), the diagnosis of structural faults that establish new interaction paths between components is an **open problem**. In the absence of such faults, possible answers, called diagnoses, can be stated in terms of broken components.

In this section we shall show how the model, i.e. the system description of the respective artifact, and the concept of a diagnosis can be stated in first order logic, and we shall present a system that implements the derivation of diagnoses from the system description and a set of observations.

2.1 System Description

Deviations from **normal** operation are faults. We assume that they are caused by malfunctioning components. Two important ideas start from this basic insight.

- First, we only need a model of the correctly functioning device, which is easier to supply than anticipating all the different forms of malfunction.
- Second, the correct system behavior should be modelled component-wise such that deviations can be traced back to individual components.

A system is considered to be composed of a set of components, *COMPS*. We use first order logic for describing the behavior of components and aggregates of components. Object constants and predicates allow us to name individual components and specify the interactions between them. The components interface with their environment through **ports** which can be designated by functions. Unary predicates are used to identify the various component types. A binary predicate, *Value*, is used to assign values to parameters and variables. Explicitly identifying ports via equalities allows specifying the device structure.

Example 1:

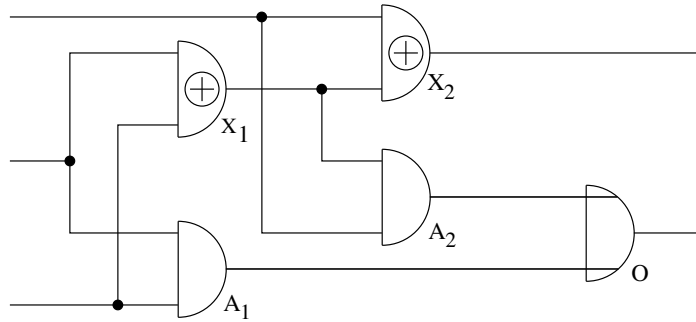


FIGURE 2 A full adder

The device in Figure 2 is a full adder. There are five gates and the connecting wires. In this example, there is a one-to-one correspondence between ports and variables (the input and output signals), and the components do not have internal parameters or state variables. 1 and 0 designate high and low signals on the ports. A formal description (which identifies ports and signals on them) may look as follows.

Domain Axioms (for ports in digital circuits)

$$\begin{aligned} \forall x \text{ COMPONENT}(x) \Rightarrow \\ [Value(Input_1(x), 1) \oplus Value(Input_1(x), 0) \\ \wedge Value(Input_2(x), 1) \oplus Value(Input_2(x), 0) \\ \wedge Value(Output(x), 1) \oplus Value(Output(x), 0)], \end{aligned}$$

where “ \oplus ” denotes “exclusive or”.

Behavior (of component types)

$$\begin{aligned}
& \forall x \text{ ANDGATE}(x) \wedge \text{ok}(x) \wedge \text{Value}(\text{Input}_1(x), 1) \\
& \quad \wedge \text{Value}(\text{Input}_2(x), 1) \Rightarrow \text{Value}(\text{Output}(x), 1) \\
& \forall x \text{ ANDGATE}(x) \wedge \text{ok}(x) \wedge \text{Value}(\text{Input}_1(x), 0) \\
& \quad \Rightarrow \text{Value}(\text{Output}(x), 0) \\
& \forall x \text{ ANDGATE}(x) \wedge \text{ok}(x) \wedge \text{Value}(\text{Input}_2(x), 0) \\
& \quad \Rightarrow \text{Value}(\text{Output}(x), 0) \\
& \forall x \text{ ORGATE}(x) \wedge \text{ok}(x) \wedge \text{Value}(\text{Input}_1(x), 0) \\
& \quad \wedge \text{Value}(\text{Input}_2(x), 0) \Rightarrow \text{Value}(\text{Output}(x), 0) \\
& \quad \text{etc.}
\end{aligned}$$

Parts

$$\begin{aligned}
& \text{ANDGATE}(A_1), \text{ANDGATE}(A_2), \text{ORGATE}(O), \\
& \text{XORGATE}(X_1), \text{XORGATE}(X_2)
\end{aligned}$$

Structure

$$\begin{aligned}
\text{Input}_1(X_1) &= \text{Input}_1(A_1), \text{Input}_2(X_1) = \text{Input}_2(A_1), \\
\text{Input}_1(X_2) &= \text{Input}_2(A_2), \text{Input}_2(X_2) = \text{Input}_1(A_2), \\
\text{Output}(X_1) &= \text{Input}_2(X_2), \text{Output}(A_1) = \text{Input}_2(O), \\
\text{Output}(A_2) &= \text{Input}_1(O)
\end{aligned}$$

Because we are describing the correct behavior of components, we have additionally introduced the predicate ‘ok’. The behavioral description of a component C contains $\text{ok}(C)$ as an explicit precondition. This will allow us to conclude $\neg\text{ok}(C)$, i.e. *abnormal*(C), when observations contradict the values derived from the model.

All of the above sentences are part of the system description, SD .

Please note, that there is no canonical representation of a device. When giving the above description we have already made some choices. For example, we did not represent the wires between the gates and, hence, neither their behavior nor the *ok*-predicate for them. Moreover, we did not represent time explicitly, e.g. by introducing an argument ‘time’ for the value-predicate. As we shall see in detail later, the model chosen to represent a device and its behavior is crucial for what kind of diagnoses can be expected. We shall, for example, not be able to diagnose a broken wire when no wire occurs in the conceptualization.

2.2 Diagnoses

In the absence of structural faults diagnosis means, at least, localizing malfunctioning components. From the perspective of the task of **fault localization**, we only have to distinguish between two modes of behavior: the

correct one and an arbitrary deviation from the correct behavior. Hence, in this section, we consider only two modes for each component C , denoted by $ok(C)$ and $\neg ok(C)$. Following [10] we define diagnoses as complete mode assignments to every component in the device. $COMPS$ generally denotes the set of all components in the device. This set is partitioned into the subsets of correct and faulty components:

Definition 1 (Mode Assignment):

Let $FAULTY \subseteq COMPS$, $OK \subseteq COMPS$ and $FAULTY \cap OK = \emptyset$.

$$D(FAULTY, OK) \equiv \bigwedge_{x \in FAULTY} \neg ok(x) \wedge \bigwedge_{x \in OK} ok(x)$$

is called a mode assignment w.r.t. $COMPS$. It is complete when $FAULTY \cup OK = COMPS$.

For the sake of brevity, we shall often write complete mode assignments as

$$D(FAULTY, \cdot) \equiv D(FAULTY, COMPS \setminus FAULTY)$$

Often we consider $D(FAULTY, OK)$ as the set

$$\{ok(x) \mid x \in OK\} \cup \{\neg ok(x) \mid x \in FAULTY\}.$$

A diagnosis problem manifests itself when assuming all components are working correctly (i.e. $FAULTY = \emptyset$) contradicts a set of observations, OBS :

$$SD \cup OBS \cup D(\emptyset, COMPS) \vdash \perp$$

A diagnosis assigns modes such that the inconsistencies are removed.

Definition 2 (Diagnosis):

Let $FAULTY \subseteq COMPS$.

A diagnosis for a system description SD , observations OBS and components $COMPS$ is a complete mode assignment $D(FAULTY, \cdot)$ such that

$$SD \cup OBS \cup D(FAULTY, \cdot)$$

is satisfiable.

Diagnoses can be numerous as we see in the following example.

Example 1 (continued):

Supplying the full adder with inputs and observing outputs as in Figure 3 clearly states a diagnosis problem, because the system description SD together with the observations

$$\begin{aligned} OBS = \{ & Value(Input_1(X_2), 1), Value(Input_1(X_1), 1), \\ & Value(Input_2(A_1), 0), Value(Output(X_2), 1), \\ & Value(Output(O), 1)\} \end{aligned}$$

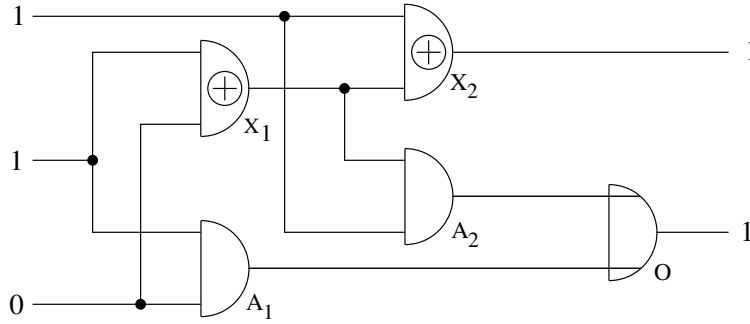


FIGURE 3 Full adder with observations

and the mode assignment $D(\emptyset, COMPS)$ is not satisfiable.

All the complete mode assignments below are diagnoses in the sense of Definition 2:

$$\begin{aligned}
 &D(\{X_2\}, \cdot), D(\{X_1, A_1\}, \cdot), D(\{X_1, A_2\}, \cdot), D(\{O, X_1\}, \cdot), \\
 &D(\{X_1, X_2, A_1\}, \cdot), D(\{X_1, A_1, A_2\}, \cdot), D(\{X_1, A_1, O\}, \cdot), \\
 &D(\{O, X_1, A_2\}, \cdot), D(\{X_1, X_2, A_2\}, \cdot), D(\{O, X_1, X_2\}, \cdot), \\
 &D(\{O, X_2, A_1, A_2\}, \cdot), D(\{O, X_1, A_1, A_2\}, \cdot), \\
 &D(\{O, X_1, X_2, A_2\}, \cdot), D(\{O, X_1, X_2, A_1\}, \cdot), \\
 &D(\{A_1, X_1, X_2, A_2\}, \cdot), D(\{A_1, X_1, X_2, A_2, O\}, \cdot)
 \end{aligned}$$

There are 16 diagnoses for this simple example! The search space of potential diagnoses is extremely large. There are $2^{|COMPS|}$ potential diagnoses. Obviously, a parsimonious representation is needed.

One may be surprised by the relatively large number of diagnoses, although for the malfunctioning device only one of the diagnoses is the actual one. A diagnosis system, however, is normally supplied only with partial information (namely inputs and outputs of the whole adder). Often, many variables describing the behavior of physical systems cannot be measured, and only for a small fraction of the measurable ones observations are actually available. As a consequence, a diagnosis program will always have to deal with sets of diagnoses rather than a unique diagnosis. Although the set of diagnoses may be huge, the system should not miss a potential diagnosis. In contrast to conventional diagnostic expert systems based on symptom-fault associations supplied by human experts, model-based diagnosis systems greatly reduce the possibility of missing a diagnosis. This can

only occur when a wrong observation is added or if the device model does not cover the intended behavior completely. When an incomplete inference engine is used, some discrepancies may not be detected. In this case, the set of diagnoses is larger than necessary but still contains all possible diagnoses.

Not all of the above diagnoses are equally plausible. While

$$D(\{X_2\}, \cdot)$$

only hypothesizes a fault in a single component,

$$D(\{A_1, X_1, X_2, A_2, O\}, \cdot)$$

as an extreme case claims that all of them are broken. Intuitively, the latter is less likely than the former, because one may assume that the device has been working correctly before or has at least been designed correctly. Therefore, a mode assignment implying a minimal deviation from the normal state is arguably a better diagnosis than one that implies everything is broken. This motivates the following definition.

Definition 3 (Minimal Diagnosis):

A diagnosis $D(FAULTY, \cdot)$ is minimal iff for no proper subset $FAULTY' \subset FAULTY$, $D(FAULTY', \cdot)$ is a diagnosis.

Example 1 (continued):

In the above example,

$$D(\{X_2\}, \cdot), D(\{X_1, A_1\}, \cdot), D(\{X_1, A_2\}, \cdot), \text{ and } D(\{O, X_1\}, \cdot)$$

are minimal diagnoses.

The following lemma states that minimal diagnoses in a way represent all diagnoses. One has to be aware, however, that even this set of minimal diagnoses can grow exponentially with respect to the number of components.

Lemma 1 (Representation Lemma):

Let $D(FAULTY, \cdot)$ be a diagnosis. Then there exists a minimal diagnosis $D(FAULTY_{min}, \cdot)$ such that $FAULTY_{min} \subseteq FAULTY$.

This is nicely illustrated in a lattice where each node denotes a set $FAULTY$ of malfunctioning components characterizing the diagnosis $D(FAULTY, \cdot)$. The partial order among the lattice elements is defined by set inclusion. Figure 4 shows the lattice for a device with four components A, B, C , and D and three minimal diagnoses $D(\{B\}, \cdot)$, $D(\{A, D\}, \cdot)$, and $D(\{A, C\}, \cdot)$. All diagnoses in the shaded area are covered by minimal diagnoses. In contrast, the minimum cardinality diagnosis $D(\{B\}, \cdot)$ alone, for instance, does not cover the diagnoses $D(\{A, C, D\}, \cdot)$ and $D(\{A, C\}, \cdot)$.

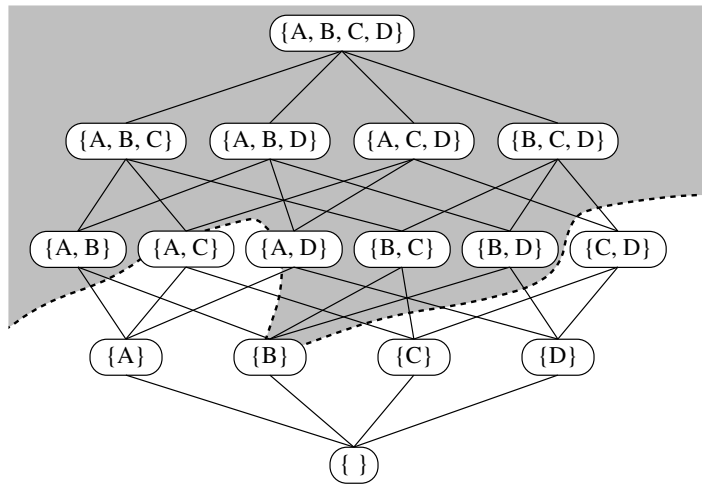


FIGURE 4 Diagnosis-lattice with minimal diagnoses

One might conjecture that the converse of the representation lemma also holds, namely that all supersets of minimal diagnoses are also diagnoses. The following intuitive counterexample shows that this is not generally true.

Example 2:

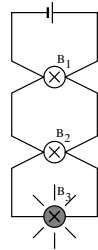


FIGURE 5 Three light bulbs

A battery is connected in parallel by six wires to three lightbulbs (Figure 5). Suppose only the last bulb in the row, B_3 , is lit. Clearly, $D(\{B_1, B_2\}, \cdot)$ is a minimal diagnosis but $D(\{B_1, B_2, B_3\}, \cdot)$ is not. After all, B_3 is lit, and, hence, not faulted.

2.3 The General Diagnostic Engine

In [9] de Kleer and Williams described a diagnostic framework that computes minimal diagnoses in the sense introduced above and subsequently influenced almost all work in the field of model-based diagnosis. Their general diagnostic engine, GDE

- computes candidates for diagnosis from **minimal conflicts**, i.e. minimal sets of component mode assumptions derived from detected inconsistencies,
- handles multiple faults, in contrast to previous systems,
- exploits an assumption-based truth maintenance system (ATMS, [5]) to identify conflicting assumption sets, and
- uses it as a basis for determining optimal probing points.

In GDE, four major phases are organized in a cycle (see Figure 6):

- behavior prediction,
- conflict detection,
- diagnoses generation and ranking, and
- discrimination between diagnoses by additional measurements

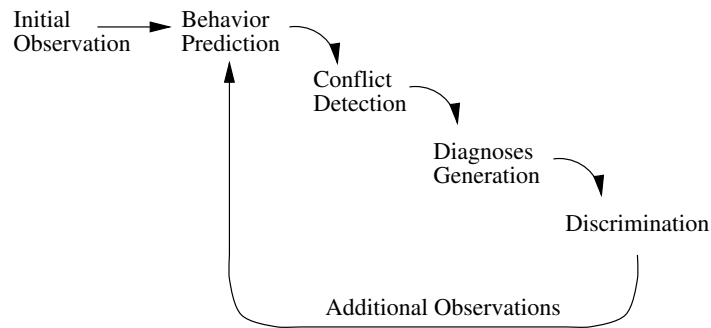


FIGURE 6 The diagnostic cycle

We shall briefly discuss these phases and their relation to the concepts introduced before.

2.3.1 Behavior Prediction

During behavior prediction, the system description (SD) and observations (OBS) are used to derive conclusions about variable values. The process is centered around the device components. Whenever sufficient informa-

tion about variables at a component is available, conclusions about other variables of the **same** component are drawn. Each of the conclusions is derived from variable values **and** assumptions about the correct behavior.

Example 1 (continued):

From input values for a gate the output is computed like, for instance,

$$\begin{aligned} & Value(Input_1(X_1), 1) \wedge Value(Input_2(X_1), 1) \wedge ok(X_1) \\ & \Rightarrow Value(Output(X_1), 0). \end{aligned}$$

GDE's predictive engine makes use of an ATMS ([5]) that serves as a repository for inferences. It caches propositional horn clauses as so-called **justifications**

$$\alpha_1 \wedge \dots \wedge \alpha_n \Rightarrow \beta.$$

A distinguished subset $ASSM$ of the set of propositional atoms, $PROP$, is called **assumptions**: $ASSM \subseteq PROP$. Component modes like e.g. $ok(X_1)$ are treated as assumptions. The set of atoms derivable from a set of assumptions (a so-called **environment**) E is called **context** of E and denoted by $cxt(E)$. All environments which allow for the derivation of the constant \perp are considered inconsistent.

Reasoning in **multiple contexts** can be characterized as considering all consistent contexts $cxt(E)$ of all subsets $E \subseteq ASSM$. All propositions are labeled with the complete set of minimal (w.r.t. set inclusion) consistent environments from which they are derivable. I.e. for a proposition p its **label** is defined as

$$\begin{aligned} Label(p) = & \{E \subseteq ASSM \mid (E \text{ is consistent} \wedge \\ & \wedge p \in cxt(E) \wedge \forall E' \subset E \ p \notin cxt(E'))\} \end{aligned}$$

Justifications are used to record the inferences performed by a problem solver, in our case a predictive engine. The label of a proposition is computed by propagating and combining environments in the network of justifications using basic set operations. By caching inferences as justifications, an inference is done once for the first context. It automatically holds in each context that is characterized by a superset of its environments. Whenever the antecedents hold in another context this causes just an update of the label of the consequent proposition. Thus, expensive recomputation is avoided.

However, labels the ATMS computes can grow big and hamper larger applications. Focusing on **interesting** contexts ([12]) avoids this problem while maintaining the essential properties of assumption-based truth maintenance.

2.3.2 Conflict Detection

Whenever a variable v takes on contradictory values, say α and β , in the same context, the supporting assumption set must be identified as a **nogood**, i.e. an inconsistent environment. This is achieved by creating justifications of the form

$$Value(v, \alpha) \wedge Value(v, \beta) \Rightarrow \perp$$

The ATMS computes and maintains a database of minimal nogoods. Therefore, the identification of minimal conflicts is trivial for the diagnostic engine; they are simply determined by the minimal nogoods, as shown in the following example.

Example 1 (continued):

For the full adder in Figure 3, the ATMS records the justification

$$Value(Output(X_2), 1) \wedge Value(Output(X_2), 0) \Rightarrow \perp .$$

$Value(Output(X_2), 1)$ was given as an observation, hence it does not depend on any assumption and, hence, holds for the empty environment. The ATMS records it as a fact:

$$Label(Value(Output(X_2), 1)) = \{\emptyset\}.$$

Please note, that this label implies that $Value(Output(X_2), 1)$ holds universally, i.e. in all logical contexts. In contrast, the label

$$Label(Value(Output(X_2), 0)) = \emptyset$$

means that there is no environment in which it holds.

For $Value(Output(X_2), 0)$, the label

$$\{\{ok(X_1), ok(X_2)\}, \{ok(A_1), ok(O), ok(A_2), ok(X_2)\}\}$$

is computed because

$$\begin{aligned} & Value(Input_1(X_2), 1) \wedge Value(Input_2(X_2), 1) \wedge ok(X_2) \\ & \Rightarrow Value(Output(X_2), 0), \end{aligned}$$

and the labels of the antecedent nodes are

$$Label(Value(Input_1(X_2), 1)) = \{\emptyset\},$$

$$Label(Value(Input_2(X_2), 1)) = \{\{ok(X_1)\}, \{ok(A_1), ok(O), ok(A_2)\}\},$$

$$\text{and } Label(ok(X_2)) = \{\{ok(X_2)\}\}.$$

$ok(X_2)$ only depends on the assumption that it holds, while the label of $Value(Input_2(X_2), 1)$ reflects the two different paths on which it can be derived. Because of the observation $Value(Output(X_2), 1)$, any assumption set supporting $Value(Output(X_2), 0)$ is a conflict. Consequently, the ATMS detects two minimal nogoods,

$$\{\{ok(X_1), ok(X_2)\}, \{ok(A_1), ok(O), ok(A_2), ok(X_2)\}\},$$

Each of these nogoods states that (at least) one of the components included must be faulty:

$$\neg ok(X_1) \vee \neg ok(X_2) \text{ and } \neg ok(A_1) \vee \neg ok(O) \vee \neg ok(A_2) \vee \neg ok(X_2).$$

These disjunctive clauses are called **conflicts**. As for diagnoses, we shall sometimes consider conflicts as sets of incorrectness assumptions. A conflict is **minimal** if it does not have a conflict as a proper subset.

2.3.3 Candidate Generation and Candidate Ranking

In a sense, minimal conflicts contain the essence of the detected discrepancies between model and artifact behavior. The diagnostic information contained in a single conflict is that at least one of the involved components is misbehaving. A diagnosis must thus hypothesize a fault for at least one of them. Hence, a minimal diagnosis $D(FAULTY, \cdot)$ is obtained from the minimal conflicts $CONFL_1, \dots, CONFL_n$ for any set $FAULTY \subseteq COMPS$ which is

a) a **hitting set** of the respective set of suspect components

$$COMPS_i := \{C_j \in COMPS \mid \neg ok(C_j) \in CONFL_i\},$$

i.e. $\forall i \text{ } FAULTY \cap COMPS_i \neq \emptyset$

and b) minimal, i.e.

$$\forall FAULTY' \subset FAULTY \Rightarrow \exists COMPS_i \text{ } FAULTY' \cap COMPS_i = \emptyset$$

Example 1 (continued):

The minimal hitting sets for the minimal conflicts we obtain for the full adder are

$$\{\neg ok(X_2)\}, \{\neg ok(X_1), \neg ok(A_1)\}, \{\neg ok(X_1), \neg ok(O)\},$$

$$\{\neg ok(X_1), \neg ok(A_2)\}.$$

This also illustrates that besides the single-fault diagnosis that X_2 misbehaves multiple fault candidates are generated.

GDE computes hitting sets from minimal conflicts by simultaneously maintaining the set of minimal hitting sets encountered so far. Unfortunately, in the worst case both the set of minimal conflicts and the set of minimal diagnoses grow exponentially in the number of device components. This fact is illustrated by an algorithm for constructing minimal diagnoses which we present in the following.

The elements of a conflict disjunctively support the respective conflict node (Figure 7),

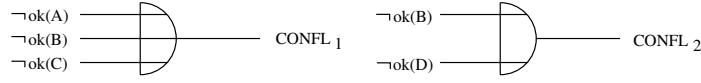


FIGURE 7 Conflict nodes $CONFL_1$ and $CONFL_2$ constructed from conflicts $\{\neg ok(A), \neg ok(B), \neg ok(C)\}$ and $\{\neg ok(B), \neg ok(D)\}$

and the conflict nodes conjunctively support the resulting candidate node Γ_1 (Figure 8).

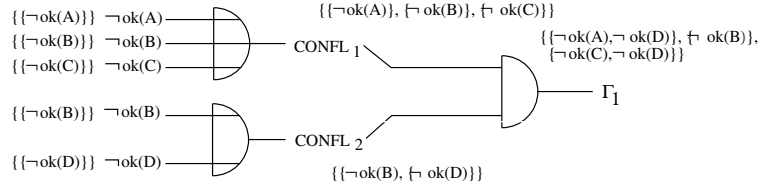


FIGURE 8 Candidate node Γ_1 formed from conflict nodes $CONFL_1$ and $CONFL_2$

Now the resulting AND-OR-tree is labeled. Each conflict element is labeled with the set containing itself, e.g. $\neg ok(A)$ is labeled with $\{\{\neg ok(A)\}\}$. Conflict nodes are labeled with the set of its antecedents' labels, e.g.

$$Label(CONFL_1) = \{\{\neg ok(A)\}, \{\neg ok(B)\}, \{\neg ok(C)\}\}.$$

Candidate nodes are labeled with the minimal set covers of their antecedent nodes, e.g.

$$Label(\Gamma_1) = \{\{\neg ok(A), \neg ok(D)\}, \{\neg ok(B)\}, \{\neg ok(C), \neg ok(D)\}\}.$$

The AND-OR-tree construction and labelling can be implemented by means of the ATMS. Conflict nodes are (disjunctively) justified by multiple justifications of the form

$$\neg ok(\cdot) \Rightarrow \text{conflict node}$$

where $\neg ok(\cdot)$ is an element of a conflict.

Additional conflicts are handled as follows. First, a conflict node is formed as above. In a second step the new conflict node and the last

candidate node, in our case Γ_1 , conjunctively justify a new candidate node:

$$\text{candidate-node} \wedge \text{new-conflict-node} \Rightarrow \text{new-candidate-node}$$

Labelling proceeds as before. Computation of minimal diagnoses is then done by the ATMS. This is both an easy and efficient way to implement **incremental** diagnosis generation.

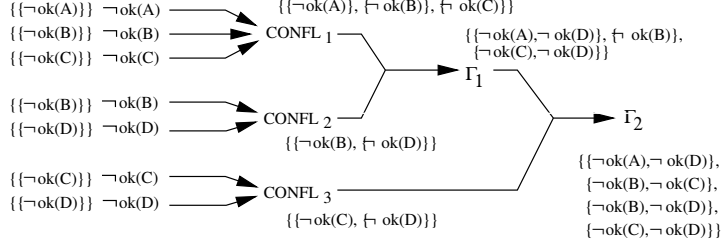


FIGURE 9 Candidate generation delegated to the ATMS

It is easily verified that the label of the last candidate node is exactly the set of the sets *FAULTY* that specify the minimal diagnoses $D(\text{FAULTY}, \cdot)$. Intuitively each candidate node selects one element from each antecedent conflict.

In order to see that there may be exponentially many minimal candidates, suppose we have $2n$ components C_1, \dots, C_{2n} and n pairwise conflicts $\{\neg ok(C_1), \neg ok(C_2)\}, \{\neg ok(C_3), \neg ok(C_4)\}, \dots, \{\neg ok(C_{2n-1}), \neg ok(C_{2n})\}$. Then each conflict node is labeled with a set containing exactly two elements: $\{\{\neg ok(C_{2i})\}, \{\neg ok(C_{2i+1})\}\}$. Minimization of set covers does not remove any of these elements since the elements of any two conflicts are different. Hence, the labels grow by a factor of 2 with each step towards the final candidate node. This has a label of size 2^n .

In practice, the potential exponential blow up of the minimal candidate sets hurts less than one would expect. The main reason seems to be that the number of components involved in conflicts is usually quite small. One possible explanation is that the devices to be diagnosed were designed by humans. Therefore, the complexity of the device is limited by a human's capability to evaluate its design.

2.3.4 Discriminating between Diagnoses by Additional Measurements

Normally, the diagnostic agent will have to discriminate between several diagnoses. For instance, the single conflict

$$\{\neg ok(A_1), \neg ok(O), \neg ok(A_2), \neg ok(X_2)\}$$

results in four minimal diagnoses:

$$D(\{A_1\}, \cdot), D(\{O\}, \cdot), D(\{A_2\}, \cdot) \text{ and } D(\{X_2\}, \cdot).$$

And every superset of any of them is also a candidate.

Likelihood of component failures often is a valuable source of discriminating information. If one can assume independence of faults in different components, then the probability $P(\delta)$ of a specific $\delta = D(FAULTY, \cdot)$ being the actual diagnosis is given by

$$P(\delta) = \prod_{C_i \in FAULTY} P(\neg ok(C_i)) \cdot \prod_{C_i \notin FAULTY} P(ok(C_i))$$

The diagnostic system can start with manufacturer-supplied probabilities for component failures. As new observations come in, however, it cannot stick to the original probability values. Some of the components might have become almost exonerated while others have become more and more suspect. The necessary updates are computed according to Bayes' rule. Depending on the outcome of the measurement of a variable X the posterior probability $P(\delta | X = v)$ can be computed [9]:

$$P(\delta | X = v) = \begin{cases} 0 & \text{if the candidate } \delta \text{ predicts } X \neq v \\ P(\delta) / P(X = v) & \text{if } \delta \text{ predicts } X = v \\ P(\delta) / m & \text{if } \delta \text{ is uncommitted w.r.t. variable } X \end{cases}$$

In the last case, the number of different values m that the variable X can take is used for a heuristic that assumes an even probability distribution among these values. The definition of "predicts" is somewhat complicated. A diagnosis δ **predicts** $X = v$ iff

- $SD \cup OBS \cup \delta$ is satisfiable, but
- $SD \cup OBS \cup \{X \neq v\} \cup \delta$ is not satisfiable.

We refrain from using the more natural formulation

$$SD \cup OBS \cup \delta \models X = v$$

for the second condition because it requires a logically complete predictive engine. The chosen definition allows one to encompass practically relevant predictive mechanisms which most often are logically incomplete.

It is important to note that a diagnosis can predict a value notwithstanding the fact that no particular fault is assumed or modelled. This is because a candidate implies (most) components being correct, and, hence, their models make predictions. Usually after taking a few measurements the probabilities for individual component failures differ radically from the initial ones. This indicates that the influence of initial probabilities is small. Therefore, one might start with equal failure probabilities for all components or with estimated values for a few component classes ([6]).

Based on the probabilistic ranking of diagnoses the system can decide what information would (on average) discriminate best between candidates.

In GDE, the next measurement is selected by choosing a variable that has minimum entropy under one-step look-ahead.

2.4 Characterization of Minimal Diagnoses

In a previous section we have shown how GDE computes minimal diagnoses. A key step is the identification of minimal conflicts. The concept of minimal conflicts proves to be useful not only for computing minimal diagnoses but also for characterizing them. A characterization of minimal diagnoses is desirable because it gives us a precise description of the expected outcome of **any** consistency-based diagnosis procedure, not just GDE. Having available such a yardstick is important. For example, almost all implemented systems use incomplete predictive engines. They may fail to detect discrepancies and hence compute suboptimal results, i.e. the set of diagnoses as represented by the computed “minimal” diagnoses may be too large.

Definition 4 (Conflict):

Let $\neg D(\textit{FAULTY}, \textit{OK})$ denote the (disjunctive) clause

$$\bigvee_{c \in \textit{FAULTY}} ok(c) \vee \bigvee_{c \in \textit{OK}} \neg ok(c)$$

$\neg D(\textit{FAULTY}, \textit{OK})$ is a **conflict** iff $SD \cup OBS \cup D(\textit{FAULTY}, \textit{OK})$ is not satisfiable, i.e.

$$SD \cup OBS \cup D(\textit{FAULTY}, \textit{OK}) \vdash \perp$$

A conflict can also be denoted by the set of literals in the disjunction, i.e.

$$\neg D(\textit{FAULTY}, \textit{OK}) = \{ok(c) | c \in \textit{FAULTY}\} \cup \{\neg ok(c) | c \in \textit{OK}\}.$$

A conflict C is minimal iff no other conflict C' is a subset of C .

The notation for a conflict was chosen to emphasize the relation to mode assignments. A conflict arises from the detection of an inconsistent mode assignment. But, to avoid confusion, please note that a conflict assigns “ ok ” to the components suspected to be faulty in the mode assignment and “ $\neg ok$ ” to the ones assumed correct:

If

$$SD \cup OBS \cup \left\{ \bigwedge_{c \in \textit{FAULTY}} \neg ok(c) \wedge \bigwedge_{c \in \textit{OK}} ok(c) \right\}$$

is not satisfiable, then

$$SD \cup OBS \vdash \bigvee_{c \in \textit{FAULTY}} ok(c) \vee \bigvee_{c \in \textit{OK}} \neg ok(c)$$

(and vice versa).

In GDE, there exist models of correct behavior only. As a result, discrepancies solely arise from “ok” modes. Consequently, conflicts in GDE have the form $\neg D(\emptyset, OK)$ or $\{\neg ok(c) | ok(c) \in OK\}$.

Definition 5 (Positive Conflict):

A positive conflict consists of negated ok-literals only.

From a logical point of view it is important that no diagnostic information is lost when diagnoses are computed from the conflicts ([8]). This is not obvious if we consider that, when stepping from detected discrepancies to the underlying conflicts, the specific form of the discrepancies (e.g. **how much** a predicted value deviates from an observed one) is lost. Only the existence of the discrepancy and its origin matters, not its specific kind. And it suffices to consider the minimal conflicts as the following theorem states:

Theorem 1 :

Let *MIN-CONFLICTS* be the set of minimal conflicts w.r.t. *SD*, *COMPS* and *OBS*, and let *FAULTY* \subseteq *COMPS*.

$D(FAULTY, \cdot)$ is a diagnosis iff

$$MIN-CONFLICTS \cup \{D(FAULTY, \cdot)\}$$

is satisfiable.

Both, diagnoses and conflicts are defined as (conjunctive and disjunctive, resp.) clauses. We obtain a specific relationship between them, if we take into account that, because of the finite number of components, $SD \cup OBS$ can be rewritten as a propositional theory. For such a background theory, concepts of “minimal” disjunctive and conjunctive clauses are captured by “prime implicates” and “prime implicants” of a theory.

A prime implicate is a “strongest” disjunctive clause implied by a theory:

Definition 6 (Prime Implicate):

Let *Th* be a set of propositional sentences.

A disjunctive clause *DC* is a prime implicate of *Th* iff

- (i) $Th \vdash DC$ and
- (ii) $\forall DC' \subset DC : Th \not\vdash DC'$.

As for node assignments and conflicts clauses are often written as sets. A prime implicant is a “weakest” conjunctive clause entailing *Th*.

Definition 7 (Prime Implicant):

Let Th be a set of propositional sentences.

A conjunctive clause CC is a prime implicant of Th iff

- (i) $CC \vdash Th$ and
- (ii) $\forall CC' \subset CC : CC' \not\vdash Th$

Conflicts are disjunctive clauses entailed by $SD \cup OBS$. It is easy to see that minimal conflicts are prime implicates of $SD \cup OBS$. The intuition behind the characterization of diagnoses is a little more complicated to grasp. If $D(FAULTY, OK)$ is satisfiable and entails the positive minimal conflicts PMC then $PMC \cup \{D(FAULTY, OK)\}$ is satisfiable. Since there are only positive conflicts, $MIN-CONFLICTS \cup \{D(FAULTY, OK)\}$ is also satisfiable. Hence, $D(FAULTY, OK)$ is a diagnosis. If we look for the weakest of such clauses we end up with the prime implicants of the positive minimal conflicts. They turn out to be exactly the minimal diagnoses.

Theorem 2 ([8]):

$D(FAULTY, \cdot)$ is a minimal diagnosis iff

$$\bigwedge_{c \in FAULTY} \neg ok(c)$$

is a prime implicant of the positive minimal conflicts.

This theorem provides a theoretical characterization of the minimal diagnoses, although not necessarily a tractable way of computing the “most interesting” diagnoses.

3 Diagnosis with Fault Modes

The big advantage of consistency-based diagnosis systems over previous mainly association-based approaches is that all deviations from normal behavior can be diagnosed without the necessity to specify the possible malfunctions. The basic principles ensure that a model of the correct behavior suffices. In its basic form, however, this generality does not cover all diagnostic inferences humans use to diagnose quickly. For instance, we do not suspect components when their faulty behaviors are not consistent with what is known about the situation. Even a human non-expert immediately knows that bulbs B_1 and B_2 must be broken in the situation of Example 2 (Figure 5). In contrast, a system like GDE does not rule out, for example, B_3 as a suspect. After all, it participates in producing discrepancies: if B_3 is lit, there is a voltage drop across it, hence also correct bulbs B_2 and B_1 should be lit, but they are not. It might exhibit the strange misbehavior that it is lit without a voltage drop. We can discard this as a physically possible fault, but GDE cannot, since it has no knowledge about possible

faults. If GDE logically negates the correct mode of a component, anything is possible, whilst if a component in reality is “physically negated” (i.e. breaks) it still does behave in one more or less deterministic, describable fashion. As a consequence, a consistency-based diagnosis algorithm based on models of correct behavior needs more probes to single out the final diagnosis. This is a severe limitation when not every point in a device is accessible for probing. Making use of knowledge about the behavior of faulty components can compensate for lacking observations. Hypothesizing a fault and then checking the consistency with what is known sometimes allows one to refute it. Of course, if no other diagnosis can be found an **unknown** fault must be present. If we assume, however, that we can enumerate all possible faults, the component can actually be exonerated.

In this section, we extend the models of components to include descriptions of faulty behaviors, as well. Nevertheless, the basic principle of the diagnostic procedure remains the same: the detection of inconsistent mode assignments which now may also involve fault modes. We shall first modify the fundamental concepts appropriately and then present two diagnostic approaches to exploiting fault models: a best-first search based on fault probabilities and a diagnostic algorithm using preferences among behavior modes and working hypotheses. The latter is grounded in a different logical foundation, namely default logic.

3.1 Multiple Behavior Modes

Rather than only the binary mode “ok” as in the basic framework we introduce a set of **different, mutually exclusive behavior modes**, $modes(C_i)$, for each component $C_i \in COMPS$, represented as propositional atoms. Accordingly, SD can contain models of several component faults ([17], [22], or [10]) which are associated with the respective modes. There is an unknown mode which has no model attached to it and, hence, can never be refuted.

Diagnosing a system means finding out what is wrong and which components work properly, which, in the extended framework, translates into appropriately assigning exactly one mode to each component.

Accordingly, we have to modify Definition 1 of a mode assignment:

Definition 8 (Mode Assignment):

A (complete) mode assignment, D , is a conjunctive clause

$$D = \bigwedge_{C_i \in COMPS} m_{j_i}(C_i), \text{ where } m_{j_i} \in modes(C_i).$$

Again, we shall also regard D as the set of involved modes:

$$D = \{m_{j_i}(C_i) \mid C_i \in COMPS\},$$

where $m_k(C_i) \in D \wedge m_l(C_i) \in D \Rightarrow k = l$.

The concept of a diagnosis remains unchanged, namely a mode assignment that does not contradict the observations.

Definition 9 (Diagnosis):

A diagnosis for a system description SD , observations OBS , and components $COMPS$ is a complete mode assignment D , such that

$$SD \cup OBS \cup D$$

is satisfiable.

We introduce different predicates for the various component modes. For example,

$$\begin{aligned} \forall c \text{ ANDGATE}(c) \wedge \text{stuck-at-0}(c) &\Rightarrow \text{Value}(\text{Output}(c), 0) \\ \forall c \text{ ANDGATE}(c) \wedge \text{stuck-at-1}(c) &\Rightarrow \text{Value}(\text{Output}(c), 1) \end{aligned}$$

describes the behaviors of an AND-gate when its output is stuck at zero or stuck at one. Because these behaviors are explicitly modeled, it is possible to refute the respective component modes.

This means there exist now models of faulty behaviors that can be checked against the observations and, possibly, be refuted. This makes the diagnostic system stronger in that it allows for fault **identification**, i.e. detecting **which misbehavior** is present, as opposed to fault **localization** in section 2, determining **where** a misbehavior is present, and the fundamental task of fault detection: noticing **that** a misbehavior is present. Sometimes, fault localization happens through fault identification. Under limited observability of a system, the only way to localize potentially faulty components may be to rule out that faults are present in the others. This exoneration of components, of course, relies on the assumption that all possible faults are known and refuted.

However, there is a price to be paid for this enhanced capability in terms of complexity. Many different faulty behaviors can be exhibited by each system constituent and, in a combinatorial way, by the entire system. Nevertheless, human experts often manage to navigate through this huge space of possibilities quite economically. This economy is essentially grounded on a **focusing** principle: “Do (or consider) only what appears necessary for the case at hand”. Applied to diagnosis, this means not to establish and check fault hypotheses that are not necessary to solve the current problem.

However, what is necessary is not obvious in advance, because the diagnostic problem is not. After all, identifying the case at hand is the **target**

of the diagnostic process. This is why the focus will change during this process. One starts with the most promising or most important focus and iteratively switches to “less preferable” ones if required. Thus, the focusing principle is complemented by a second one, the **preference** principle: “Determine and shift the focus according to the order of preferences”.

This leads to establishing and checking fault hypotheses only if the ones that are more plausible or more likely (or more dangerous) have already been ruled out. As a result, the system spends more work, the more unlikely or exceptional a diagnostic case is.

These principles underly, for instance, the SHERLOCK system ([10]), where preferences are induced by fault probabilities, the focusing strategy in GDE⁺ ([22], [12]), the exploitation of model abstraction and simplification in DP ([20]), and the revision process in MAGELLAN ([2]).

3.2 Diagnosis as Best-First Search

In [10] and [7], the SHERLOCK algorithm was described that

- assumes a priori component mode probabilities as part of the system description,
- enumerates and checks the consistency of diagnoses in the order of their likelihood until the so-called set of leading diagnoses covers a pre-defined percentage of the probability mass for diagnoses,
- takes most informative probes based on the leading diagnoses, and
- updates component mode probabilities when new evidence is gathered.

SHERLOCK makes use of a focused ATMS ([7], [12]) that allows checking of individual candidates for diagnosis without the cost of computing complete ATMS labels.

In SHERLOCK, the modes for each component are totally ordered by their probabilities. The ok mode is ranked highest and the unknown mode is ranked lowest. Based on mode probabilities, also diagnoses are ranked by their probabilities. Although later evidence changes the probability distribution over diagnoses the order of remaining consistent diagnoses is determined by the a priori probability distribution only. Diagnoses are considered better when they have higher probability.

Definition 10 (Leading Diagnoses):

Let k be a small natural number. A diagnosis D is called a **leading diagnosis** iff less than k diagnoses have higher probability than D .

Fault probabilities are only estimates. Therefore, for large component libraries one cannot guarantee a correct overall total ordering of the modes of all components. Consequently, a diagnosis engine like SHERLOCK may

terminate without having generated a diagnosis that puts the actually broken components into fault mode.

3.3 Diagnosis based on Defaults

An alternative to using probabilities of component modes is to specify an ordering on the modes of **one** component. One must, however, be aware that in doing so, global information about the order between component modes of **different** components is lost. By virtue of their numeric nature probabilities induce a total order on the diagnoses whereas an order defined locally for each component induces only a partial order.

3.3.1 Preferred diagnoses

In order to express differences between the modes such as the frequency of occurrence, likelihood, or criticality, we impose an order on them:

Definition 11 (Preference):

A preference is a partial order on the modes of each component:

$$\geq \subseteq \text{modes}(C_i) \times \text{modes}(C_i)$$

where the correct behavior mode is the most preferred and the unknown mode the least preferred one:

$$\forall m_j(C_i) \in \text{modes}(C_i) \setminus \{ok(C_i)\} : ok(C_i) > m_j(C_i),$$

$$\forall m_j(C_i) \in \text{modes}(C_i) \setminus \{unknown(C_i)\} : m_j(C_i) > unknown(C_i),$$

where ‘>’ is defined as: $x > y :\Leftrightarrow x \geq y \wedge \neg(y \geq x)$.

Preferences among modes induce a preference order on mode assignments: For $D = \{m_{j_i}(C_i)\}$ and $D' = \{m'_{j_i}(C_i)\}$

$$D \geq D' :\Leftrightarrow \forall i m_{j_i}(C_i) \geq m'_{j_i}(C_i).$$

The diagnoses of interest can now be defined as the most preferred mode assignments that accord with the system description and the observations.

Definition 12 (Preferred Diagnoses):

D is a preferred diagnosis iff no other diagnosis is strictly preferred over it: For all diagnoses $D' : D' \geq D \Rightarrow D' = D$

3.3.2 Characterization of Preferred Diagnoses in Default Logic

The intuition behind the preferences among modes is that one can assume a mode $m_j(C_i)$ to be the actual one in a particular context (of other components’ modes and observations),

- if all modes strictly preferred to $m_j(C_i)$ have been refuted in this context,
- provided there is no evidence against $m_j(C_i)$ in the context.

This can be conveniently expressed in default logic ([19]). A (normal) default is an inference rule

$$a : b / b$$

with the meaning “If a is established and it is consistent to assume b , then b ”. Hence, for each mode $m_j(C_i)$ a default def_{ij} is defined as:

$$\bigwedge_{m_k(C_i) \in pre_j(C_i)} \neg m_k(C_i) : m_j(C_i) / m_j(C_i)$$

where

$$pre_j(C_i) := \{m_k(C_i) \mid m_k(C_i) > m_j(C_i)\}$$

is the set of “predecessors” of $m_j(C_i)$. In particular, for each component C_i , the “cascade” of preference defaults starts with

$$: ok(C_i) / ok(C_i)$$

which basically means “Normally C_i works correctly”.

A default theory is a pair of a set of defaults, DEF , and a set of premises (formulas), P . While in classical monotonic logic, P has a uniquely determined deductive closure

$$Cn(P) := \{p \mid P \vdash p\},$$

a default theory (DEF, P) can have several **extensions**, which contain, besides the monotonically derivable formulas, also the consequences of maximal sets of applicable defaults. With the preference defaults defined above, preferred diagnoses can be characterized as extensions of the respective default theory:

Theorem 3 (Characterization of preferred diagnoses):

Let $DEF = \{def_{ij}\}$ be the set of preference defaults. A diagnosis D is a preferred diagnosis iff $Cn(SD \cup OBS \cup D)$ is an extension of the default theory $(DEF, SD \cup OBS)$.

3.3.3 Computing Preferred Diagnoses

In the Default-based Diagnosis Engine (DDE, [13], [14]), extensions of a default theory and, hence, preferred diagnoses are computed by means of the **non-monotonic assumption-based truth-maintenance system** (NM-ATMS, see [11]).

The basic ATMS ([5]) which was described as part of GDE stores inferences of a problem solver as **justifications** that record the dependency of conclusions on their antecedents. It uses this “frozen inference structure” to **label** each conclusion with all minimal, consistent sets of assumptions (i.e. ultimate premises) that allow deriving this conclusion.

The NM-ATMS provides a way to encode a normal default, say

$$\neg ok(C_i) : fault_1(C_i) / fault_1(C_i),$$

as a justification

$$\neg ok(C_i) \wedge out(\neg fault_1(C_i)) \Rightarrow fault_1(C_i),$$

where $out(\neg fault_1(C_i))$ is an assumption meaning that the negation of $fault_1$ has not been established (and, hence, assuming $fault_1$ causes no inconsistency).

Intuitively, each extension of the default theory decides upon each mode $m_j(C_i)$: it contains its negation or assumes it. If χ_{ij} denotes this choice, these disjunctions are encoded by two justifications

$$\neg m_j(C_i) \Rightarrow \chi_{ij} \quad \text{and} \quad out(\neg m_j(C_i)) \Rightarrow \chi_{ij}$$

for each component mode. They are all conjoined by the justification

$$\chi_{11} \wedge \dots \wedge \chi_{nm_n} \Rightarrow \phi.$$

In [11] it has been shown that the label of ϕ contains assumption sets that generate the extensions of the default theory. These are sets of mode assumptions corresponding to mode assignments that are consistent, hence diagnoses, and generated by the preference defaults and, hence, are preferred diagnoses.

Remark :

Note that extending the set of defaults by another preference default def_{kl} requires, besides the two justifications for χ_{kl} , one additional justification

$$\phi \wedge \chi_{kl} \Rightarrow \phi'$$

to obtain the extensions of the resulting default theory from the label of ϕ' .

As described in more detail in section 4.3, this is the basis for an algorithm that, instead of starting with the entire set of preference defaults $DEF = \{def_{ij}\}$, incrementally adds defaults as mode assignments are refuted in order to determine their successors in the preference order. DDE ([13]) exploits this to keep the default theory (and the set of instantiated models) as small as possible. It can be understood as focusing in a best-first search according to the preference order as opposed to SHERLOCK which is based on probabilities.

4 The Diagnostic Process

4.1 Diagnosis as Hypothetical Reasoning

The systems discussed in section 3 illustrate that, if we want to improve efficiency of diagnostic algorithms, we have to develop control principles that focus the activities of the consistency-based diagnosis engine: the predictive engine must be focused on a substructure of the whole model, diagnoses

generation must be confined to the most promising diagnoses, etc. If we still do not want to give up completeness, such focusing principles must not absolutely exclude any parts of the problem space from investigation. It must be possible for the system to regenerate and reconsider observations, models, and diagnoses that were neglected before, due to some control decision. In other words, any inferences dependent on such decisions have to be treated as defeasible inferences. Establishing reasonable working hypotheses in order to simplify problem solving is essential for human skills in reasoning about complex systems, e.g. in diagnosis. Still more important is the capability to notice when such a working hypothesis turns out to be wrong and to recover from this situation. For instance, one might start diagnosis by looking for a familiar fault that explains the observed misbehavior, realize later on that this leads to very unlikely coincidences of several faults, and, for that reason, then consider components damaged in a rare or unknown way. Other examples of diagnostic working hypotheses are the belief in the correctness of observations (e.g. supplied by sensors), the absence of structural damage, and the non-intermittent nature of the present fault.

To mimic reasoning with working hypotheses in a diagnostic system and to make them retractable, one has to make them explicit first. Hence, the basic framework of consistency-based diagnosis has to be extended in order to

- explicitly represent the various diagnostic assumptions, and to
- explicitly represent the knowledge required for reasoning about the presence of these assumptions as important constituents of any real diagnostic process.

For this purpose, yet another set of propositions, *WHYP*, is introduced, capturing the diagnostic assumptions, and the goals and concepts of consistency-based diagnosis are revised.

Definitions 2 and 9 of a diagnosis have to be modified in order to express that we are looking for a set of mode assignments to components **and a set of diagnostic hypotheses** that is consistent with the system description and the observations.

Definition 13 (Diagnosis under a Set of Hypotheses):

Let $WHYP' \subseteq WHYP$ be a set of diagnostic hypotheses. A mode assignment D is a diagnosis of $(SD, COMPS, OBS)$ under a working hypotheses $WHYP'$ iff

$$SD \cup OBS \cup WHYP' \cup D$$

is satisfiable.

The introduction of diagnostic assumptions is a major change. But from a formal point of view, Definition 13 does not look very different from the original one if we join assumptions about the component modes with the diagnostic assumptions. For the special case of only two modes, $ok(C)$ and $\neg ok(C)$, as discussed in section 2, this leads to the basic idea of DP (“Diagnosis as a Process”, [22]): exploit the consistency-based diagnosis techniques to determine valid sets of diagnostic assumptions, i.e. apply consistency-based diagnosis to $COMPS \cup WHYP$.

So, let for $whyp \in WHYP$, $ok(whyp) \Rightarrow whyp$ and $\neg ok(whyp) \Rightarrow \neg whyp$

Definition 14 (DP-Diagnosis):

Let $FAULTY \subseteq COMPS \cup WHYP$.

$D(FAULTY, (COMPS \cup WHYP) \setminus FAULTY)$ is a DP-diagnosis iff it is a diagnosis of the system $(SD, COMPS \cup WHYP, OBS)$.

Remark :

It is fairly easy to see that this provides a way to determine diagnoses under a set of hypotheses: Let $FAULTY_{COMPS} \subseteq COMPS$ and $FAULTY_{WHYP} \subseteq WHYP$. If

$$FAULTY = FAULTY_{WHYP} \cup FAULTY_{COMPS}$$

is a DP-diagnosis of $(SD, COMPS \cup WHYP, OBS)$ then $FAULTY_{COMPS}$ is a diagnosis of $(SD, COMPS, OBS)$ under $WHYP \setminus FAULTY_{WHYP}$.

Briefly speaking, the solution is to apply the GDE algorithm of section 2.3 to both mode assumptions and diagnostic assumptions.

In the default logic approach, a working hypothesis, $whyp$, can be encoded as a default in a natural way,

$$: whyp / whyp,$$

expressing that the diagnostic system is to “prefer” believing $whyp$, unless there is evidence to the contrary. The extensions to the respective default theory will maintain maximal (w.r.t. set inclusion) sets of working hypotheses.

There is a certain dilemma, because, on the one hand, diagnostic assumptions are introduced in order to simplify the diagnostic reasoning process by “concealing” exceptional cases, but, on the other hand, making them explicit may let the space of diagnoses grow considerably. What one would like to do is treating these assumptions as “working hypotheses” which are maintained as long as possible and considered as retractable (which means occurring in diagnoses) only if this resolves an inconsistency. In DP ([20]), this idea is captured by the concept of a **focus of suspicion**

which specifies the kind of candidates that are considered at a certain stage of the diagnostic process. A focus of suspicion (*FOS*) is simply a set of assumption sets, $FOS \subseteq 2^{COMPS \cup WHYP}$, and can be used, for instance, to exclude the consideration of candidates containing particular types of diagnostic assumptions, such as assumptions about the correctness of observations or exclusion of unknown faults. If the intersection of the focus of suspicion with the DP diagnoses becomes empty or contains only unlikely candidates, this demands for a change of the focus which becomes a major issue in control of the diagnostic process.

4.2 Diagnosis with Multiple Models

Consistency-based diagnosis uses models for determining behavior modes by ruling out (combinations of) behavior modes, if the respective models are contradicting the actual system's behavior. Obviously, a theory of model-based diagnosis has to be a theory of the content and diagnostic use of models in the first place.

However, as a more or less explicit research strategy, most of the approaches in the field focused on problems of the diagnostic procedure, presuming there exists a simple and unique way of modeling the device or avoiding to specify its nature. More structure has been imposed on the model by incorporating **fault models** (Section 3) and **hierarchy**. Like a hierarchical structure, the distinction between different views on component models ([4], [21], [16]) aims at exploiting different perspectives and at increasing efficiency of diagnosis of non-trivial artifacts.

Although considered as an essential ingredient to model-based reasoning from the very beginning ([3]), some of the most powerful means for this purpose, namely reasoning with **abstractions**, **simplifications** and **approximations**, have received more attention only recently.

Why is this a difficult task? Generally speaking, abstractions may turn out to be too coarse to achieve a satisfactory diagnosis, and simplifications may miss important features of a certain problem. Trivially, if a diagnostic system uses a model that is inadequate for a particular case at hand, the resulting diagnosis is likely to be wrong or at least useless; the model may fail to reveal an existing contradiction, or it may suggest an inconsistency that is only virtual. The dilemma is: the best protection against such failures, namely using the best, most accurate and most detailed model available, tends to make the task intractable, at least unnecessarily complex for the "simpler cases". And what makes the system work efficiently, namely certain assumptions that simplify the model, may render it useless if these assumptions are violated in the situation to be dealt with. This is one reason why one would like to have **multiple models** of a system and let the diagnostic system use the one **appropriate** for the **particular**

case. It should also be appropriate **w.r.t. the stage of the diagnostic process.**

The system description, SD , is then considered as a repository for various, alternative models, and the diagnosis system uses only a subset SD' of it for prediction. The guiding principle for selecting models ought to be “Use models as simple as possible and as sophisticated as necessary”. It implies that the model might be exchanged during the diagnostic process. First, because it may not be obvious from the very beginning what model is required for the particular case. Second, because different stages in the process require different models (e.g. abstract and cheap models for generating initial diagnostic candidates and more detailed ones for discriminating among them). Switching to a different model should not mean merely starting the procedure again from zero, but should allow for carrying over of as much information as possible from previous diagnostic steps. This requirement sets the essential questions to be answered:

- How can a set of multiple models be structured appropriately? What are important, basic transformations and relations of models, and what are their properties and effects?
- How and under what conditions do results obtained from using one model carry over to diagnostic reasoning with a different model?
- How can the system deal with wrong information derived from an inappropriately simplified model? How can it handle contradictory parts of the model?
- How can the system obtain criteria for deciding which part of the model to use and when to switch to a different one?

Although, from the physics point of view, the various relationships among the models are of quite different nature and complexity (such as structural or temporal abstractions, linear approximations, neglecting certain influences), most of them can be reduced to two types of fundamental logical relations that suffice to characterize their impact on consistency-based diagnosis. The central question is whether one model is a weaker version of another one (i.e. its logical implication), or whether its use is additionally based on assumptions about the context.

Definition 15 (View):

A model M' is a view of another model, M , iff $M \Rightarrow M'$.

Definition 16 (Simplification):

M' is a simplification of M , iff

$$\exists \{whyp_i\} \subseteq WHYP (M \wedge \bigwedge_i whyp_i \Rightarrow M').$$

In both cases, the (intentionally simpler and “cheaper”) model M' can be used as a substitute for the original one, M . If the former is refuted, then so is the latter. However, in case of a simplification, one can only be sure if the modeling assumptions hold. The impact on consistency-based diagnosis is captured by the following theorem.

Theorem 4 :

Let $SD' \subset SD$, $WHYP' \subseteq WHYP$, $whyp_0 \in WHYP$, D a mode assignment and M, M' be two models. If D is a diagnosis for the system

$$(SD' \cup \{M\}, COMPS, OBS)$$

under $WHYP'$ and M' is a simplification of M : $M \wedge whyp_0 \Rightarrow M'$ then D is a diagnosis for the system

$$(SD' \cup \{M'\}, COMPS, OBS)$$

under $WHYP'$, or the modeling assumption $whyp_0$ has to be retracted:

$$SD' \cup \{M\} \cup OBS \cup WHYP' \cup D \vdash \neg whyp_0.$$

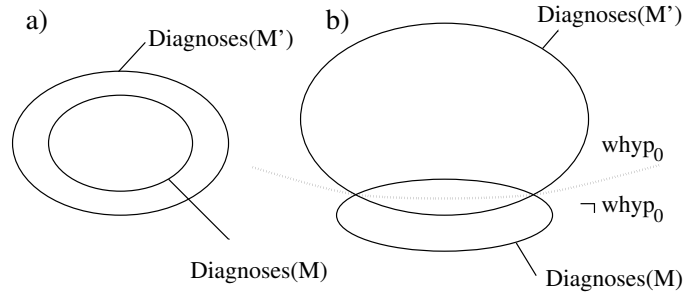


FIGURE 10 Relationship among sets of diagnoses for
a) a view, b) a simplification
(the dotted line bounds the set of diagnoses under $whyp_0$)

The case of a view is obtained if $whyp_0 \equiv True$, which means $\neg whyp_0$ cannot be entailed and all diagnoses obtained for M are also diagnoses for its view, M' . When using the original model, M , instead of its view, the space of diagnoses shrinks monotonically, while it may change non-monotonically if a simplifying assumption is dropped, but still, in using the simplified model, all diagnoses that are valid under the simplifying assumption are obtained (Figure 10).

The simplification can also be used to encode a careful version of “physical negation” (as discussed in section 3.1) by making the modeling as-

sumption explicit that really all possible faults have been enumerated and modelled:

$$\begin{aligned} & \text{ANY-POSSIBLE-FAULT}(C) \wedge \\ & \text{FAULT-MODELS-COMPLETE}(C) \\ & \Rightarrow \text{faulty}(C) \\ & \text{faulty}(C) \Rightarrow \text{faulty}_1(C) \vee \text{faulty}_2(C) \vee \dots \vee \text{faulty}_n(C) \end{aligned}$$

More examples for model relations and a simple model switching strategy are presented in [20]. As a result, a graph of models for a constituent is obtained with the model nodes linked by (hyper) arcs labeled by the introduced model relations.

Equipped with the logical formalism, multiple models including abstract, simplified, and approximate models can be exploited in consistency-based diagnosis. However, there remains the problem to recognize or establish the various model relations when analyzing or constructing variations of models. After all, the models of the physical systems that are of interest here are usually not given as sets of first order formulas.

Frequently, the ordinary representation of available behavior descriptions is given in terms of (differential) equations involving a number of characteristic physical variables and parameters that often take values in the domain of real numbers, integers, or intervals of such. And this is the kind of representation many standard techniques for transforming models apply to (think of linear approximation, dropping negligible terms, changing scale, etc.). This is why we present a theory of **relational** models of physical systems that allows us in a rather straightforward way to check for or achieve the model relations introduced above. It is also quite close to the implementation basis of many model-based diagnosis systems, namely constraint systems. In the following, we summarize some of the key concepts of this theory.

The best way to understand this viewpoint (at least from the logician's perspective) is probably the following. Rather than regarding behavior models of components as logical theories and manipulating them, consider the (logical) models of these theories, i.e. relations. The various relationships of behavior models are then obtained from modifications and transformations of these relations.

The behavior of some (primitive or aggregate) constituent of the system to be diagnosed is described by (vectors of) possible values of some variables and parameters $v_i \in VARS$ (local to the constituent) taken from domains $DOM(v_i)$.

Definition 17 (Representation):

A representation is a pair

$$(\underline{v}, DOM(\underline{v})) = ((v_1, v_2, \dots, v_k), DOM(v_1) \times DOM(v_2) \times \dots \times DOM(v_k)).$$

Under the multiple modeling aspect, different representations for the same constituent are allowed, varying \underline{v} and/or $DOM(\underline{v})$. If a constituent is in a particular condition (the intended one, or damaged or disturbed in a particular way), its respective behavior can be characterized by constraining the set of possible values for \underline{v} which can be expected in a physically realizable situation. This means, it is specified by some relation $R \subseteq DOM(\underline{v})$.

To turn this into a behavior model stated as a formula, let SIT denote the set of physically possible situations (“worlds” reflecting different contextual and internal conditions, e.g. input vectors and parameter settings).

Definition 18 (Value-Predicate):

For, $s \in SIT$, $v_i \in VARS$, $v_{i0} \in DOM(v_i)$, $Value(s, v_i, v_{i0})$ v_i has the value v_{i0} in situation s .

$Value$ is extended to the vector $\underline{v} = (v_1, v_2, \dots, v_k)$ by

$$\forall s \in SIT \quad Value(s, \underline{v}, (v_{10}, v_{20}, \dots, v_{k0})) \Leftrightarrow \bigwedge_{i \in \{1, \dots, k\}} Value(s, v_i, v_{i0})$$

Observation of a value, \underline{v}_0 , in a situation, s , implies that the Value-predicate holds: $Obs(s, \underline{v}, \underline{v}_0) \Rightarrow Value(s, \underline{v}, \underline{v}_0)$. Note that this does not postulate uniqueness of values, i.e. we allow

$$\exists s \in SIT \quad Value(s, \underline{v}, \underline{v}_0) \wedge Value(s, \underline{v}, \underline{v}'_0) \wedge \underline{v}_0 \neq \underline{v}'_0$$

even in one domain. A behavior model is then the claim that all values that can be observed in a real situation, are contained in the characterizing relation R .

Definition 19 (Model):

A relation $R \subseteq DOM(\underline{v})$ specifies a model by

$$M(R) \Leftrightarrow \forall \underline{v}_0 \in DOM(\underline{v}) ((\exists s \in SIT \quad Value(s, \underline{v}, \underline{v}_0)) \Rightarrow \underline{v}_0 \in R).$$

Note that in this definition the implication was used only in one direction (thus allowing R to cover more than what is physically realizable). This opens one dimension for variations in the model, but still enables the system to refute the model if any value outside the respective relation is observed and, hence, suffices to serve consistency-based diagnosis.

More formally, if a context,

$$CTX \subseteq SD \cup WHY P' \cup D,$$

given by the system description, a set of working hypotheses, $WHYP' \subseteq WHYP$, and a mode assignment $D = \{m_{ji}(C_i) \mid C_i \in COMPS' \subseteq COMPS\}$, together with a set of observations, OBS , predicts (entails, that is) a value \underline{v}_0 in a situation s ,

$$CTX \cup OBS \vdash Value(s, \underline{v}, \underline{v}_0),$$

and $\underline{v}_0 \notin R$, then $M(R)$ is refuted in CTX :

$$CTX \cup OBS \vdash \neg M(R).$$

Otherwise, a model is called valid w.r.t. $CTX \cup OBS$.

We now discuss how logical relations between different behavior models can be obtained based on the analysis or construction of the respective specifying relations. A model is specified by some relation $R \subseteq DOM(\underline{v})$. Hence, a change in a model may be caused by the use of

- different variables, \underline{v} , or
- different domains, $DOM(\underline{v})$, for the same variables, or
- different relations for defining a model while \underline{v} and $DOM(\underline{v})$ remain fixed.

The first two cases imply a shift in representation, whereas the third preserves it and directly modifies the model.

First, we consider such shifts in the overall representation. A “reasonable” transformation that turns one representation into another one should respect the *Value* predicate in the following sense:

- If *Value* holds for a value in the original representation, then it also holds for the transformed value. For instance: if we map real numbers to $\{negative, zero, positive\}$, then $Value(s, i, -5)$ implies $Value(s, i, negative)$.
- If *Value* holds for a transformed value, this must be the case for one of its pre-images in the original representation: $Value(s, v, neg)$ is true only if there exists some real number $r < 0$ such that $Value(s, v, r)$ holds.

Definition 20 (Representational Transformation):

Let $(\underline{v}, DOM(\underline{v}))$ and $(\underline{v}', DOM'(\underline{v}'))$ be two representations.

A mapping $\tau : DOM(\underline{v}) \rightarrow DOM'(\underline{v}')$ is Value-preserving iff

$$\forall \underline{v}_0 \in DOM(\underline{v}) \forall s \in SIT \ Value(s, \underline{v}, \underline{v}_0) \Rightarrow Value(s, \underline{v}, \tau(\underline{v}_0)).$$

It is Value-grounding iff

$$\forall \underline{v}'_0 \in DOM'(\underline{v}') \forall s \in SIT$$

$$Value(s, \underline{v}', \underline{v}'_0) \Rightarrow \exists \underline{v}_0 \in DOM(\underline{v}) (Value(s, \underline{v}, \underline{v}_0) \wedge \tau(\underline{v}_0) = \underline{v}'_0).$$

It is a representational transformation, iff it is Value-preserving and Value-grounding.

The fundamental theorem for representational transformations is the following:

Theorem 5 :

Let $R \subseteq \text{DOM}(\underline{v})$, $R' \subseteq \text{DOM}'(\underline{v}')$, and $\tau : \text{DOM}(\underline{v}) \rightarrow \text{DOM}'(\underline{v}')$.

If τ is **Value-grounding** then the **image of a model is a view of this model**:

$$M(R) \Rightarrow M(\tau(R)).$$

If τ is Value-preserving, then the inverse transformation also creates a view:

$$M(R') \Rightarrow M(\tau^{-1}(R')).$$

Obviously, these kinds of transformations are important, because they transform models into **views** (in the sense of Definition 15), and, hence, Theorem 4 applies.

Of course, the results of the previous subsection remain true if a representation is mapped onto itself. But now all kinds of “surgery” are allowed to be applied to relations. Consider linear approximation as a widespread technique that modifies relations without changing the representation.

The question to be answered is: Given a relation $R \subseteq \text{DOM}(\underline{v})$ that specifies a valid model, $M(R)$. When modifying R to $R' \subseteq \text{DOM}(\underline{v})$ what can be said about the validity of $M(R')$? And, more specifically: How can one tell whether a view or a simplification is obtained? Answering this question is quite straightforward in this formalism.

If R specifies a valid model, i.e. covers the actual behavior, then R' can fail to do so only where it does not cover R (shaded area in Figure 11).

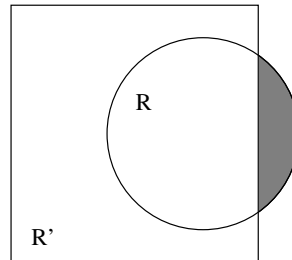


FIGURE 11 Where the simplified model $M(R')$ may fail

More formally, let $R_0 \subseteq \text{DOM}(\underline{v})$ be the “exact” modeling relation. This implies

$$\begin{aligned} & \forall \underline{v}_0 \in \text{DOM}(\underline{v}) (\exists s \in \text{SIT Value}(s, \underline{v}, \underline{v}_0)) \Rightarrow \underline{v}_0 \in R_0, \\ & \forall \underline{v}_0 \in \text{DOM}(\underline{v}) ((\exists s \in \text{SIT Value}(s, \underline{v}, \underline{v}_0)) \Rightarrow \underline{v}_0 \in R_0 \cap R' \vee \underline{v}_0 \in R_0 \setminus R') \\ & \text{and, therefore} \end{aligned}$$

$$\forall \underline{v}_0 \in \text{DOM}(\underline{v}) ((\exists s \in \text{SIT Value}(s, \underline{v}, \underline{v}_0) \wedge \underline{v}_0 \notin R_0 \setminus R') \Rightarrow \underline{v}_0 \in R')$$

Thus, we have obtained a condition dependent on R_0 , which may be unknown or too complex to represent, and independent of R . But if $M(R)$ is a model, then $R_0 \subseteq R$ and, hence, we get a sufficient condition for $\underline{v}_0 \in R'$:

$$\underline{v}_0 \notin R \setminus R' \Rightarrow \underline{v}_0 \notin R_0 \setminus R' \Rightarrow \underline{v}_0 \in R'.$$

Lemma 2:

Let $R, R' \subseteq \text{DOM}(\underline{v})$.

$$\begin{aligned} & M(R) \wedge (\forall \underline{v}_0 \in \text{DOM}(\underline{v}) (\exists s \in \text{SIT Value}(s, \underline{v}, \underline{v}_0)) \Rightarrow \underline{v}_0 \notin R \setminus R') \\ & \Rightarrow M(R') \end{aligned}$$

The condition is always satisfied, if $R \setminus R' = \emptyset$, i.e. R is contained in R' .

Corollary 1:

Every superset of a relation specifying a model specifies a view of this model:

$$R \subseteq R' \subseteq \text{DOM}(\underline{v}) \Rightarrow (M(R) \Rightarrow M(R')).$$

Otherwise, there is a modeling assumption involved.

Corollary 2:

Let $R, R' \subseteq \text{DOM}(\underline{v})$ and $\text{whyp}_{R' \setminus R} \in \text{WHYP}$ be a diagnostic hypothesis such that

$$\begin{aligned} & \text{whyp}_{R' \setminus R} \Rightarrow \\ & (\forall \underline{v}_0 \in \text{DOM}(\underline{v}) (\exists s \in \text{SIT Value}(s, \underline{v}, \underline{v}_0)) \Rightarrow \underline{v}_0 \notin R \setminus R'). \end{aligned}$$

Then $M(R')$ is a simplification (according to Definition 16) of $M(R)$:

$$M(R) \wedge \text{whyp}_{R' \setminus R} \Rightarrow M(R').$$

We emphasize, that Corollary 2 gives modeling assumptions a precise semantics.

This is a way to modify representations and models and to analyze their relationships. Theorem 4 then determines the effect model switching will have on the space of diagnoses. However, there remains the question of how the system can determine both the necessity of model switching and the appropriate model to switch to. This is an important aspect of controlling the diagnostic process, and since modeling assumptions are working

hypotheses, it becomes part of the problem to control the set of hypotheses under which the system works.

4.3 Control

In both the theoretical characterization of preferred diagnoses and their computation, the focusing principle was the motivation behind the preferences.

4.3.1 The Basic Idea

Let us revisit the idea behind the mode preferences. Figure 12 shows an abstract example of a portion of the preference lattice for some component C_i .

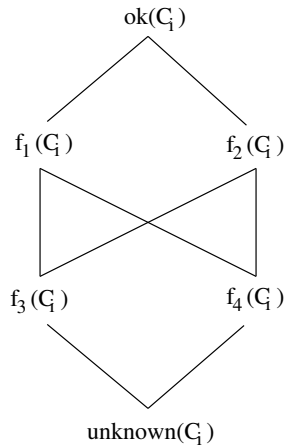


FIGURE 12 A portion of a preference lattice

In the beginning, C_i , like all other components, is assumed to work correctly. What do we need to instantiate at this time?

- First, the **ok default** with the respective assumption,
- second, the **ok model** in order to check the consistency of this mode, and
- finally, the preference defaults for the successors of $ok(C_i)$,

$$\neg ok(C_i) : f_j(C_i) / f_j(C_i) \quad \text{for } j = 1, 2,$$

in order to know where to proceed in case the check of $ok(C_i)$ reveals an inconsistency.

What we do **not** yet need, is

- the behavior models of any of the fault modes $f_j(C_i)$ since they are not yet considered, and
- the preference defaults and respective mode assumptions for all modes $f_j(C_i)$, $j \leq 3$, since their immediate predecessors have not yet been checked, leave alone refuted.

If $ok(C_i)$ is really refuted in some context (given by models for the other components and working hypotheses) then defaults def_{i1} , def_{i2} apply in this context, which means now behavior models of $f_1(C_i)$ and $f_2(C_i)$ have to be checked, and we need to install the defaults def_{i3} and def_{i4} for potential continuation, but not their respective models. These defaults will not apply and activate these models unless there exists a context in which both $f_1(C_i)$ and $f_2(C_i)$ are negated.

So, the focusing principle is to successively introduce

- the behavior models of modes and
- their successor defaults

at the time when their respective preference defaults apply in at least one context. This is the case when all predecessor modes have been refuted through checking their models. This process will stop when the remaining mode combinations survive the check, and our intuition tells us that they represent the preferred diagnoses.

However, preferred diagnoses are defined w.r.t. the complete default theory

$$DT_{complete} = (DEF, SD \cup OBS \cup WHY P'),$$

while our iterative focus extension along the preference links in a hypothesize-and-test cycle amounts to determining extensions of a sequence of default theories

$$DT_k = (DEF_k, SD_k \cup OBS \cup WHY P')$$

given by monotonically growing subsets $DEF_0 \subseteq DEF_1 \subseteq \dots \subseteq DEF$ and $SD_0 \subseteq SD_1 \subseteq \dots \subseteq SD$ of defaults and behavior models. (SD_0 contains the structural description of the device and the models of correct component behaviors) The respective sequence of preferred diagnoses w.r.t. the partial default theories actually converges against those of the complete one, even though it usually terminates with proper subsets of D and SD . This effectively focuses

- hypothesizing diagnosis candidates, because the necessary justification graph as outlined in section 4.1 as well as the label of ϕ is kept small by limiting the set of mode assumptions and the corresponding χ nodes; the incremental step from DEF_k to DEF_{k+1} can be handled easily as shown by the remark in section 3.3.3,

- testing diagnosis candidates by limiting the set of activated behavior models for checking hypothesized preferred diagnoses.

4.3.2 Sequentially Approaching Preferred Diagnoses

We inductively define a sequence of default theories

$$DT_k = (DEF_k, SD_k \cup OBS \cup WHY P').$$

$PD(DT_k)$ denotes the preferred diagnoses w.r.t. DT_k . Newly generated preferred diagnoses are called candidates:

$$CAND_k = PD(DT_{k+1}) \setminus PD(DT_k).$$

$$DEF_0 = \{ : ok(C_i) / ok(C_i) \mid i = 1, \dots, n \}$$

$$DEF_{k+1} = DEF_k \cup \dots \cup \bigcup_{D \in CAND_k} \bigcup_{m_{ij} \in D} succ(m_{ij})$$

where $succ(m_{ij})$ is the set of defaults introducing the immediate successor modes for m_{ij} in the preference order.

$$SD_{k+1} = SD_k \cup \dots \cup \bigcup_{D \in CAND_k} models(D)$$

where $models(D) = \{ model_{ij} \mid m_j(C_i) \in D \}$ is the set of models associated with the modes given by D .

The sequence reaches a fixpoint when no new candidates have been generated. Let us denote the respective default theory by

$$DT_{stop} = (DEF_{stop}, SD_{stop} \cup OBS \cup WHY P').$$

The theorems below justify stopping at this point. The preferred diagnoses computed from DT_{stop} are exactly the same as those for $DT_{complete}$.

Theorem 6 (Soundness):

Let the sequence of default theories have its fixpoint at DT_{stop} . Then

$$PD(DT_{stop}) \subseteq PD(DT_{complete}).$$

On the one hand, adding more preference defaults (from $DEF \setminus DEF_{stop}$) cannot change the set of extensions, and hence the preferred diagnoses, because their preconditions (negated modes) are not satisfied. On the other hand, adding more behavioral models (from $SD \setminus SD_{stop}$) cannot change the set of preferred diagnoses either. Behavioral models are conditioned on their respective behavioral mode as in $f_1(C_i) \Rightarrow output(C_i) = 0$. Since the modes for the added behavioral models are not contained in any extension, the extensions do not change their assignment of modes.

The next theorem shows that still a complete account of the diagnostic situation is given when stopping at DT_{stop} .

Theorem 7 (Completeness):

Let the sequence of default theories have its fixpoint at DT_{stop} . Then

$$PD(DT_{complete}) \subseteq PD(DT_{stop}).$$

4.3.3 Focusing Candidate Testing

The process of approaching preferred diagnoses sequentially deserves more attention. From the diagnosis viewpoint, computing the extensions comprises two different tasks. At step $k + 1$

- the candidates generated at the previous step, $CAND_k$, are tested for consistency with their models added to SD_k .
- if this leads to the refutation of a candidate, possibly new candidates are generated.

With each step, the set of models in SD_k grows, and so the set of possible model combinations explodes. There is, however, only a need to check the model combinations corresponding to candidates. If a candidate is not refuted, we have found a final preferred diagnosis. This is stated by the following theorem.

Theorem 8 (Focusing):

Let $D \in PD(DT_k)$. If

$$SD_0 \cup models(D) \cup OBS \cup WHYP' \cup D$$

is consistent, then $D \in PD(DT_{complete})$.

Figure 13 shows an algorithm that makes use of the theorem by focusing model-based prediction on the model combinations indicated by the candidates. (Its implementation is based on a focused ATMS ([12]).

```

    Compute preferred diagnoses
    PD := ∅
    CANDIDATES := {{ok(Ci) | Ci ∈ COMPS}}
    while CANDIDATES ≠ ∅ do
      choose D ∈ CANDIDATES
      check consistency of SD0 ∪ models(D) ∪ OBS ∪ WHYP' ∪ D
      if consistent then PD := PD ∪ {D}
      else create preference defaults for ∪mij ∈ D succ(mij)
           justify χ'ijs
           justify new ϕ' to obtain PD'
           CANDIDATES := PD' \ PD
    end while
    return PD
  
```

FIGURE 13: Focused generation of preferred diagnoses

A candidate never needs to be in the focus twice. Even stronger, some candidates need never be in the focus, since, while checking one candidate, inconsistencies may be detected that refute other existing candidates. Hence, it pays off to update candidates after each check.

4.3.4 Focusing Candidate Generation

The focusing theorem makes obvious that during the process of approaching the preferred diagnoses sequentially, some candidates may turn out to be final preferred diagnoses early on while other preferred diagnoses may show up after many iterations. It is not possible to know in advance **when** final preferred diagnoses occur. The order, however, in which they occur is not arbitrary and can be subject to further control. If, for the next consistency check, we always select a candidate with a minimal number of assigned fault modes, the algorithm generates preferred diagnoses ordered by their number of faults. This can be exploited by restricting the generation of preferred diagnoses (which may still form a prohibitively large set) by limiting the number of iterations and/or computed preferred diagnoses. In our experience, a set of 5 to 10 preferred diagnoses suffices for effectively deciding on next steps in the overall diagnostic process, such as probe selection or change in working hypotheses.

5 Discussion, Research Issues

There has been considerable progress in both developing a sound theoretical basis for automated diagnosis and building powerful systems applied to interesting problems. Despite these accomplishments, there are a number of open issues that require more efforts. There are extensions necessary for many applications. Some generalizations seem possible to cover similar tasks, but there also exist some limitations that appear hard to overcome.

5.1 Dynamics

Systems with a behavior changing over time and, in particular, with memory (i.e. storage of energy or information from previous states) pose a number of hard problems. Besides the basic problem of modeling (finding appropriate representations of time, the right granularity, etc.), we face a new dimension of complexity. As we mentioned before, the strong and useful restriction provided by only local interaction of components may be rendered ineffective. Not only model-based prediction is affected, the diagnostic reasoning has to reflect this dimension. For instance, focusing (w.r.t. measurements or prediction) has to include the temporal aspect. Even stronger, the fault may change dynamically, and we may face the problem of modeling intermittent faults or exploiting the assumption that a non-intermittent fault is present. An intermittent fault may well cause a permanent deviation in the system's behavior, whereas a stable fault might

create symptoms only occasionally. The time-varying behavior of systems (in conjunction with the requirement to prevent or minimize damage and costs) can impose restrictions on the time available for diagnostic reasoning. Currently, we are lacking general principles for solving this problem. Neither special hardware and implementations, nor prioritizing rules and similar techniques that may have been successfully exploited in one “real-time application”, can guarantee that for a different application (or even a different example) the system comes up with a useful diagnosis in time. A **real real-time architecture** for diagnostic systems is needed, which is guided by the goal to have a useful (potentially refineable) diagnosis available at all times. It seems that we already have produced some theories and tools that support this goal, such as (structural and behavioral) abstraction of models, reasoning under simplifying assumptions, and handling context shifts and non-monotonicity.

5.2 Diagnosis and Repair

Real diagnostic processes include reasoning **and acting** as well as the aim of **re-establishing the desired functionality**. But what we have discussed so far (and what basically has been achieved in the field) is the task of finding the fault, not fixing it. The fact that diagnosis should serve the repair task, ought to have an impact on the diagnosis system. For instance, it does not make sense to spend efforts on discriminating between two faults that require the same repair action. Repair actions can support finding the fault (e.g. by replacing a suspect component or bridging it by a structural change), and we need theories and systems that integrate both.

One has to be aware that this raises new fundamental problems. First, we inherit all problems related to **reasoning about actions**, in particular, the frame problem, introducing a new source and a new kind of non-monotonicity into diagnosis. The non-monotonicity we have considered in the previous section has its roots in the incomplete picture of the real world and the necessity to revise beliefs about it. Every possible change was specified by the closed world of component models (plus the device topology). With actions, **the world itself may change** and, hence, enforce revisions. We face similar problems, if we try to extend diagnosis in another direction. Rather than merely identifying the broken component, we might want to reason about **what happened when it broke**. The occurrence of a fault is nothing to be predicted within the behavior model and appears as a “causeless” action (as for monitoring tasks).

The second problem raised, or rather emphasized, by the introduction of the repair task is representing and exploiting teleology. So far, the model describes the (physical) **behavior** of a device, but not the **purpose** this behavior implements. Although the importance of teleological reasoning has been recognized early, there is little to grasp at the present time.

5.3 Structural Faults

The concept of diagnosis we have been using throughout this paper is strongly based on the preservation of the original structure. By the definition of a diagnosis as a mode assignment, faults are confined to broken components, and the structure description in SD is assumed to be unchanged. But a fault can exactly be a violation of the designed structure, such as a bridge fault and an unintended heating of a component by an adjacent one.

Structural changes due to **broken connections** can easily be handled within the approach by treating and modeling connections like components (at the cost of increasing their number).

It is the **additional** connections, the unintended interactions between components in the case of failures that cause problems for a very deep reason. There is no straightforward general way to extend our theory by simply opening the model to revisions. What would be the nature of the beliefs to be revised? In contrast to propositions concerning the behavior of a component or the existence of a connection, it is nothing positively present in the theory. Rather, what has to be revised is the non-existence of unknown influences. Each behavior model (even the fault models) are based on some implicit assumptions that there are no influences on the component other than the ones specified within the device model. This can be expressed by some proposition stating that all parts being correct implies the whole functioning well, provided the structure has not changed:

$$ok(Part_1) \wedge \dots \wedge ok(Part_n) \wedge NoStructuralFault(C) \Rightarrow ok(C)$$

But, in general, we are not able to enumerate the potential disturbances. In the example, there might be evidence for retracting the diagnostic hypothesis $NoStructuralFault$, but this does not provide a clue to **what kind** of structural fault might be present.

Note that, besides the theoretical foundation, many principles and techniques that make the solution to the problem feasible depend on presumptions about the structure. For instance, we emphasized that the locality of the constituent models and of the interactions appears to be crucial for the tractability of the approach, and that efficiency is gained by focusing and control techniques that heavily exploit the (unchanged) structure description. Structural faults are threatening these accomplishments to some degree. But practical considerations require an extended theory of diagnosis, and we have some prerequisites for it. The reason why we have a chance to solve this problem is that, even under structural faults, changes still tend to be local. After all, we are not diagnosing a car after it went down the slope of the coast of Big Sur, but, say, the effect of a marten chewing on some cables and brake fluid hoses in our garage, or a workpiece that got stuck in a production line and blocked it.

We may exploit the fact that our diagnosis framework does not necessarily fail completely. Though unable to identify the structural fault, it suspects a component, or, more likely, a set of components. This information in combination with knowledge about the structure and adjacency of components can be the basis for focused hypotheses, e.g. of bridge faults. We can also try to transform the problem into the existing framework by modeling the **non-existence** of a particular structural fault as a (virtual) component whose fault model describes the effect of the structure violation. For instance, a non-existing bridge fault is represented by a perfect insulator between two wires. Beyond such specialized solutions, we need a theory of diagnosis that includes structural faults.

5.4 Knowledge-based Diagnosis - A Challenge and Touchstone for AI

We hope to have shown that automating diagnosis of technical systems involves several problems that are central to AI research and require substantial efforts in theory and system development. If our goal is to build powerful and general systems whose competence is significantly superior to that of specialized systems based on symptom-fault associations, the task includes non-monotonic reasoning, (qualitative) reasoning about physical systems, and controlling and focusing complex reasoning. It is a challenge for methods developed in these fields to prove useful in solving a real problem.

The diagnostic task has some inherent properties that restrict it in a strong way, turning an interesting real problem into a feasible one. These restrictions are basically reflecting the fact that the system to be diagnosed is an artifact. Having been designed and produced by people, it can be expected to be well-structured. This structure provides essential features, in particular well-defined constituent elements, limited interaction of these elements, and locality and minimality of changes considered to be faults.

Especially the last feature distinguishes diagnosis from other tasks, such as theory formation about physical phenomena or (innovative) design. In contrast to design, diagnosis is in a transition zone between basic theoretical work and applicable systems. What has not been discussed in this paper, is that this has indeed lead to the implementation of powerful diagnostic frameworks and non-trivial applications. This is very important, because it makes diagnosis a concrete context for evaluating the utility of various AI methods and techniques. If we propose knowledge-based diagnosis as a touch-stone for AI, we do so because we are convinced that only real applications can verify or falsify the value of our theories, as opposed to flying penguins, shot turkeys, or a simple block on a spring. The progress that has been stimulated in theoretical work related to diagnosis confirms this. And still, as we emphasized, there remains a lot to be done.

References

- [1] C. Böttcher. No faults in structure? – how to diagnose hidden interactions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1728–1734, San Mateo, CA, 1995. Morgan Kaufmann Publishers.
- [2] C. Böttcher and O. Dressler. Diagnosis process dynamics: Holding the diagnostic trackhound in leash. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93)*, San Mateo, CA, 1993. Morgan Kaufmann Publishers.
- [3] R. Davis. Expert systems: Where are we? and where do we go from here? *Artificial Intelligence*, 3(2), 1982.
- [4] R. Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410, 1984.
- [5] J. de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28(2):127–161, 1986.
- [6] J. de Kleer. Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45(3), 1990.
- [7] J. de Kleer. Focusing on probable diagnoses. In *Proceedings of the National Joint Conference on Artificial Intelligence (AAAI-91)*, pages 842–848, San Mateo, CA, 1991. Morgan Kaufmann Publishers.
- [8] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.
- [9] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.
- [10] J. de Kleer and B. C. Williams. Diagnosis with behavioral modes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1324–1330, San Mateo, CA, 1989. Morgan Kaufmann Publishers.
- [11] O. Dressler. Problem solving with the NM-ATMS. In *Proceedings of the European Conference on Artificial Intelligence (ECAI-90)*, Stockholm, 1990.
- [12] O. Dressler and A. Farquhar. Putting the problem solver back in the driver’s seat: Contextual control over the ATMS. In J. P. Martins M. Reinfrank, editor, *Truth Maintenance Systems*, pages 1–16. Springer, 1990.
- [13] O. Dressler and P. Struss. Back to defaults: Characterizing and computing diagnoses as coherent assumption sets. In John Wiley & Sons, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI-92)*, pages 719–723, 1992.
- [14] O. Dressler and P. Struss. Model-based diagnosis with the default-based diagnosis engine: Effective control strategies that work in practice. In John Wiley & Sons, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI-94)*, pages 677–681, 1992.
- [15] B. Faltings and P. Struss, editors. *Recent Advances in Qualitative Physics*. MIT Press, 1992.
- [16] W. Hamscher. Modeling digital circuits for troubleshooting. *Artificial Intelligence*, 1991.

- [17] L. J. Holtzblatt. Diagnosing multiple failures using knowledge of component states. In *Proceedings of the Conference on Artificial Intelligence Applications (CAIA)*, pages 139–143, 1988.
- [18] C. Preist and B. Welham. Modeling bridge faults for diagnosis in electronic circuits. In *Working Notes of the International Workshop on Principles of Diagnosis (DX-90)*, 1990.
- [19] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
- [20] P. Struss. Diagnosis as a process. In L. Console W. Hamscher, J. de Kleer, editor, *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
- [21] P. Struss. What’s in SD? Towards a theory of modeling for diagnosis. In L. Console W. Hamscher, J. de Kleer, editor, *Readings in Model-based Diagnosis*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
- [22] P. Struss and O. Dressler. Physical negation - integrating fault models into the general diagnostic engine. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1318–1323, San Mateo, CA, 1989. Morgan Kaufmann Publishers.
- [23] D. Weld and J. de Kleer, editors. *Qualitative Reasoning About Physical Systems*. Morgan Kaufmann Publishers, San Mateo, CA, 1990.

