Model-Based Reconfiguration: Toward an Integration with Diagnosis*

This paper was presented at AAAI-91, July 14-19, Anaheim, CA and appears in *Proceedings, AAAI-91, Volume 2*, pages 836-841.

Judith Crow and John Rushby
Computer Science Laboratory
SRI International
Menlo Park, California 94025 USA
crow@csl.sri.com, rushby@csl.sri.com

Abstract

We extend Reiter's general theory of model-based diagnosis [13] to a theory of reconfiguration. The generality of Reiter's theory readily supports an extension in which the problem of reconfiguration is viewed as a close analogue of the problem of diagnosis. Using a reconfiguration predicate **rcfg** analogous to the abnormality predicate **ab**, we formulate a strategy for reconfiguration by transforming that for diagnosis. A benefit of this approach is that algorithms for diagnosis can be exploited as algorithms for reconfiguration, thereby promoting an integrated approach to fault detection, identification, and reconfiguration.

*The research reported here was supported by the National Aeronautics and Space Administration under Contract No. NAS1-18969.

1 Introduction

Automated diagnosis has been one of the most fruitful applications of AI. However, while it is important to identify the faults in a malfunctioning system, the real problem is usually to repair the system so that it can continue its mission. Thus in many applications, diagnosis is only part of a larger problem known as Fault Detection, Identification, and Reconfiguration (FDIR). Surprisingly, despite continued interest in diagnosis, there has been relatively little work on the foundations of recovery and reconfiguration, and virtually none on the problem of integrated FDIR—although the practical benefits of an integrated approach could be considerable, especially if knowledge of available reconfigurations can be used to help discriminate among competing diagnoses.

1.1 The Approach

The practical motivation for our work derives from systems such as airplanes and spacecraft, which typically possess considerable redundancy in the form of back-up systems, as well as degraded operating modes. In this paper we present a theory of reconfiguration for such systems as an analogue of Reiter's model-based theory of diagnosis [13]. We chose Reiter's theory as a point of departure because it provides a formal characterization of diagnosis shared to some extent by most of the model-based systems described in the literature, including DART [8], GDE [4] and its descendents [5, 9], and the work of Davis [2]. Our approach follows from two basic insights: first, the generality of Reiter's theory of diagnosis makes it applicable to other domains; second, a productive analogy exists between the problem of diagnosis and that of reconfiguration. Diagnosis is the problem of identifying components whose abnormality is sufficient to explain an observed malfunction. Similarly, reconfiguration can be viewed as the problem of identifying components whose reconfiguration is sufficient to restore acceptable behavior. There are two potential benefits to characterizing reconfiguration as an extension of Reiter's theory of diagnosis in this way: first, we can exploit algorithms for diagnosis as algorithms for reconfiguration, and second, we have a unified foundation which should facilitate the development of an integrated theory of FDIR.

The paper is organized as follows: in the next section we motivate an integrated approach to FDIR in general and delineate our approach in particular. We then develop both an intuitive and a formal characterization of reconfiguration, followed by an example. After the example, we discuss the limitations of the consistency-based analogy exploited here, focusing on issues of minimality, consistency and entailment. The last two sections provide a survey of related work and concluding remarks, respectively.

1.2 Why FDIR?

We view the limited focus of extant work on automated fault diagnosis (whether rule-based or model-based) as a serious drawback to its practical applicability. Classical approaches to diagnosis, which simply identify faulty components, solve only half the problem of automated FDIR. Reconfiguration and recovery, the other half of the problem, is typically either ignored, reduced to a set of preplanned procedures (which is inherently at odds with the expressed intent of model-based approaches), or handled as a planning problem distinct from the original diagnosis problem. In contrast, we believe that the most effective approaches to FDIR will be those that consider FDIR as an integrated problem, in which diagnosis and recovery are solved in concert. Some of the potential benefits of an integrated approach to FDIR include the following.

- A single computational engine can be used for both diagnosis and reconfiguration.
- A significant reduction of the search space can be achieved: only those diagnoses that require different reconfigurations need be distinguished, and the number of possible reconfigurations is typically much smaller than the number of diagnoses.
- Temporary reconfigurations can be used to discriminate among competing diagnoses: e.g., does the symptom disappear when we switch to a back-up system?
- Application to domains such as real-time, operative systems [1] becomes more relevant, accommodating, for example, the requirement to place the system in a safe state even without a firm diagnosis.
- A broader context is provided for both diagnosis and recovery, in which potential consequences of misdiagnosed faults and incorrect recovery actions can be properly evaluated, and resources effectively apportioned.

The theory we develop in this paper does not realize these benefits; our objective here is to propose a characterization of reconfiguration that will promote this goal of effective integration.

2 A Characterization of Reconfiguration

Reiter's formulation of the diagnosis problem can be informally described as follows. Given a description of the design or structure of a physical system and an observation of its behavior which differs from that expected, find a set of components whose abnormality explains the discrepancy between the observed and expected system behavior. The system description is couched in terms of the assumed "non-abnormality" of its components (e.g., "if a light bulb is not abnormal, and has a voltage applied, the bulb will be lit"). In the simplest realizations of this approach, the system description specifies the behavior of non-abnormal components only; later formulations have augmented the system description with axioms for physically impossible behavior [7], and explicit "fault models" [5, 14].

By analogy with this formulation of diagnosis, the problem of reconfiguration can be posed as follows: given a system description and a diagnosis, i.e., a set of components assumed abnormal, find a set of components whose reconfiguration yields an acceptable behavior. In terms of this analogy, the problem of finding a reconfiguration that will produce an acceptable behavior is formally identical to the problem of finding a diagnosis that explains an observed behavior. We can make the analogy more concrete by specifying an abstract engine and showing how the specification can be interpreted to provide either a diagnosis or a reconfiguration engine.

Let M be a domain model, including "normality" assumptions expressed in terms of a distinguished predicate P on components C, and let B be a specified (observed or desired) behavior. M and B are sets of first order formulas and an explanation E is a set of subsets of C. Intuitively, E generalizes the notions of diagnosis and reconfiguration; the members of E "explain" the discrepancy, if any, between the model M and behavior B. We define E to comprise satisfactory explanations relative to M and B just in case for all $\Phi \in E$,

$$M \cup B \cup \{P(c) | c \in \Phi\} \cup \{\neg P(c) | c \in C - \Phi\}$$

is consistent. The characterization of satisfactory explanations by logical consistency is the crucial notion in the approach to diagnosis exploited here; for this reason, it is often referred to as "consistency-based" diagnosis.

The transformation necessary to map a diagnosis engine to a reconfiguration engine can be viewed as the interpretations which instantiate the general formulation, as given below.

Interpretation	M	B	P	E
Diagnosis	**sd**	**obs**	**ab**	$\{\Delta_1, \ldots, \Delta_n\}$
Reconfiguration	**sd′**	**reqs**	**rcfg**	$\{\Re_{\Delta_1}, \ldots, \Re_{\Delta_n}\}$

Here **sd** and **sd′** are the system descriptions for diagnosis and reconfiguration, respectively, **obs** is the observed behavior, and **reqs** is the required or acceptable behavior. Δ_i is a diagnosis, and $\Re_{\Delta_i}$ is a reconfiguration.

Under the interpretation for diagnosis, the predicate P is the familiar abnormality predicate **ab**, which is used with negative polarity to express the normality assumptions, e.g., $\neg\mathbf{ab}(m_1)$ denotes that the component m_1 is behaving normally.

Similarly, under the interpretation for reconfiguration, the normal assumption is that a component is not **reconfig**ured, denoted by $\neg$**rcfg**.

A simple example should help clarify these notions. Consider the problem of diagnosing and repairing a flat on a bike equipped with a single spare tire. To simplify the statement of the problem, we use a typed logic. *Wheel* and *tire* are uninterpreted types, *front* and *back* are constants of type *wheel*, x is a variable of the same type, and a, b and *spare* are constants of type *tire*. The function *on* has signature $\mathit{wheel} \rightarrow \mathit{tire}$ and indicates which tire is on which wheel, *good* and *rcfg* are predicates on wheels, and *ab* is a predicate on tires. Intuitively, *ab* indicates whether or not a tire is serviceable, $\mathit{rcfg}(x)$ indicates whether the spare is to be mounted on wheel x, and *good* indicates whether or not a wheel has a serviceable tire. In this and subsequent discussion, we make the simplifying assumption that components used in reconfiguration are not abnormal; in this case, the spare tire is assumed serviceable. The system description is as follows.

$$\neg ab(on(x)) \supset good(x)$$
$$rcfg(x) \supset on(x) = spare$$
$$\neg rcfg(front) \supset on(front) = a$$
$$\neg rcfg(back) \supset on(back) = b$$
$$\neg rcfg(front) \vee \neg rcfg(back)$$
$$\neg rcfg(front) \wedge \neg rcfg(back)$$

The last of these axioms indicates the initial configuration—i.e., neither wheel is reconfigured. Suppose we notice that our back tire is rapidly loosing air, i.e.,

$$\neg good(back).$$

From the model, we discover there is a single diagnosis $\{b\}$, i.e., $ab(b)$ is consistent with the model and the observation. We now add

$$ab(b) \wedge \neg ab(a) \wedge \neg ab(spare)$$

to the system description, withdraw the initial configuration

$$\neg rcfg(front) \wedge \neg rcfg(back),$$

establish the requirement

$$good(front) \wedge good(back),$$

and seek a reconfiguration that is consistent with this requirement and the system description. Clearly $\{back\}$ does the job; we should put the spare on the back wheel.

Note that there are two phases to this approach: first we fix the configuration and seek diagnoses, then we fix a diagnosis and seek a reconfiguration. In general, there will be several diagnoses and we will probably seek a reconfiguration for each before committing to a final choice.

Although very simple, this example illustrates an important point: the concept of reconfiguration can be decoupled from the mechanisms for achieving it. In a later example, the reconfiguration predicate is applied to switches in an electrical circuit, thereby equating the concept and the mechanism of reconfiguration.

We are now ready to discuss the formal development of the analogy between diagnosis and reconfiguration, using Reiter's definition of a system as our point of departure.

Definition 1 A **system** is a pair **(sd, comps)** where **sd**, the *system description*, is a set of first-order sentences and **comps**, the *system components*, is a finite set of constants. An *observation*, **obs**, of a system is a finite set of first-order sentences. Thus **(sd, comps, obs)** denotes a system **(sd, comps)** with observation **obs** [13, pp. 59, 62].

A diagnosis, Δ, for **(sd, comps, obs)** is a possibly empty set of components $\{c_1, \ldots, c_n\}$, such that the union of $\{\mathbf{ab}(c) | c \in \Delta\}$, $\{\neg\mathbf{ab}(c) | c \in \mathbf{comps} - \Delta\}$, **sd**, and **obs** is consistent.

Given a system and a diagnosis Δ, we define a reconfiguration *relative to* Δ, i.e., a reconfiguration based on the assumption that the components specified by Δ are behaving abnormally. Let **sd′** be the original **sd** modified by removing axioms describing the initial status of reconfigurable components, and let the requirements, **reqs**, be a finite set of first-order sentences specifying a desired or acceptable behavior for the reconfigured system. In the definitions which follow, it is useful to remember that we are taking unions over sets of clauses, yielding conjunctions of (first-order) sentences. The predicate **rcfg** denotes "reconfigured."

Definition 2 A **reconfiguration** for **(sd′, comps, reqs)** relative to Δ is a minimal set $\Re_\Delta \subseteq \mathbf{comps}$ such that the following is consistent.

$$\begin{aligned}&\mathbf{sd}' \cup \mathbf{reqs}\\ &\cup \{\mathbf{ab}(c) | c \in \Delta\} \cup \{\neg\mathbf{ab}(c) | c \in \mathbf{comps} - \Delta\}\\ &\cup \{\mathbf{rcfg}(c) | c \in \Re_\Delta\} \cup \{\neg\mathbf{rcfg}(c) | c \in \mathbf{comps} - \Re_\Delta\}\end{aligned}$$

Definition 2 characterizes a reconfiguration relative to a diagnosis as the smallest set of components such that the assumption that these components are reconfigured and that all other components are not (reconfigured) is consistent with the diagnosis, the augmented system description, and the requirements. The notion of minimality implicit in this definition is explored in a later section.

This approach to reconfiguration provides insight, but ultimately not much practicality. We have captured the intuition that a reconfiguration is a conjecture that recovery can be achieved by reconfiguring (only) certain components, but we have not provided the basis for an effective mechanism for computing all reconfigurations. In an extended version of this paper[1] we give a formal development of a computational procedure which parallels the development given by Reiter for diagnosis. In the discussion here, we abbreviate this development and present only the key definitions and the basic theorem.

Following Reiter, we exploit the notion of *conflict set* and *hitting set* to arrive at an effective computational basis.

Definition 3 A **conflict set** for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs}, \Delta)$ is a set $\{c_1, \ldots, c_k\} \subseteq \mathbf{comps}$ such that $\mathbf{sd'} \cup \mathbf{reqs} \cup \{\mathbf{ab}(c) | c \in \Delta\} \cup \{\neg\mathbf{rcfg}(c_1) \wedge \ldots \wedge \neg\mathbf{rcfg}(c_k)\}$ is inconsistent. A conflict set for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs}, \Delta)$ is **minimal** iff no proper subset of it is a conflict set for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs}, \Delta)$.

We specify a reconfiguration in terms of this notion of conflict set.

Proposition 1 $\Re_\Delta \subseteq \mathbf{comps}$ is a reconfiguration for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs})$ relative to Δ IFF $\Re_\Delta$ is a minimal set such that $\mathbf{comps} - \Re_\Delta$ is not a conflict set for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs}, \Delta)$.

To characterize the computation of a reconfiguration, we also need the notion of hitting set.

Definition 4 A **hitting set** for a collection of sets C is a set $H \subseteq \bigcup_{S \in C} S$ such that $H \cap S \neq \{\ \}$ for each $S \in C$. A hitting set for C is **minimal** iff no proper subset of it is a hitting set for C.

We can now characterize the computation of a reconfiguration as follows.

Theorem 1 $\Re_\Delta \subseteq \mathbf{comps}$ is a reconfiguration for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs})$ relative to Δ IFF $\Re_\Delta$ is a minimal hitting set for a collection of conflict sets containing at least the minimal conflict sets for $(\mathbf{sd'}, \mathbf{comps}, \mathbf{reqs}, \Delta)$.

As in the case of diagnosis, this characterization of reconfiguration is the basis for an algorithm for computing all reconfigurations, namely Reiter's "algorithm"[2] for computing all minimal hitting sets from a given collection of (at least the minimal) conflict sets.

This completes our account of the formal analogy between reconfiguration and diagnosis. The model of reconfiguration suggested above is very simple. A serious

[1] Available on request from the authors.

[2] The quotes are Reiter's and are intended to remind the reader that the general problem of computing all diagnoses is undecidable, although there are effective computations for many applications.

account of FDIR must factor in several dimensions including the level of redundancy, the level of acceptable functionality, and the granularity of the diagnosis versus that of the reconfiguration. Diagnosis associates abnormality with components, whereas reconfiguration potentially associates malfunction with a range of system units, the smallest of which is the diagnosable component.

3 An Example

Our example is a minor variation on a standard one (see, e.g., [7, p. 332]) consisting of a battery and a series of bulbs connected in parallel as shown in Figure 1. We have added three reconfiguration switches, $r1$, $r2$, $r3$, normally set so that standby spares $b4$, $b5$ and auxiliary bulb $b6$ are not wired into the circuit. Unlike the bike example, here we represent the mechanisms of reconfiguration explicitly as elements of the model; together, the two examples illustrate that it is possible, but not necessary, to model the notion of reconfiguration explicitly in terms of physical components, such as valves, switches, or other mechanisms that actually perform reconfiguration.

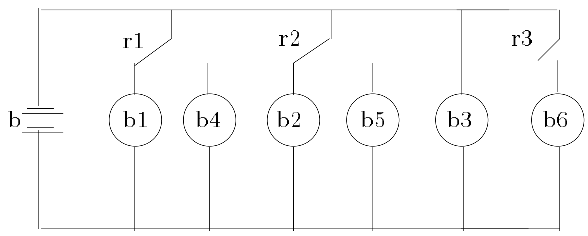

Figure 1: A simple circuit with an auxiliary bulb and two standby spares.

We specify the **comps** of the system in Figure 1 as $\{b, b1, b2, b3, b4, b5, b6, r1, r2, r3\}$ and the sentences that constitute the **sd** as shown below (where the first six sentences axiomatize the correct behavior of the components and the last three describe the physical configuration). Following the tradition of model-based diagnosis, we model our system in an untyped, first-order logic. Variables are denoted by capital letters and are (implicitly) universally quantified. We assume that wires always behave correctly.

$$bulb(X) \wedge \neg ab(X) \wedge powered(X) \supset lit(X)$$
$$bulb(X) \wedge \neg ab(X) \wedge \neg powered(X) \supset \neg lit(X)$$
$$bulb(X) \wedge \neg ab(X) \wedge lit(X) \supset powered(X)$$
$$bulb(X) \wedge \neg ab(X) \wedge \neg lit(X) \supset \neg powered(X)$$
$$battery(X) \wedge \neg ab(X) \supset powered(X)$$
$$wired(X,Y) \supset powered(X) \equiv powered(Y)$$
$$battery(b) \wedge bulb(b1) \wedge \ldots \wedge bulb(b6)$$
$$\neg rcfg(r1) \wedge \neg rcfg(r2) \wedge \neg rcfg(r3)$$
$$wired(b,b1) \equiv \neg rcfg(r1) \wedge wired(b,b2) \equiv \neg rcfg(r2)$$
$$\wedge wired(b,b3) \wedge wired(b,b4) \equiv rcfg(r1)$$
$$\wedge wired(b,b5) \equiv rcfg(r2) \wedge wired(b,b6) \equiv rcfg(r3)$$

The expected behavior is that bulbs $b1, b2, b3$ are lit. The observation

$$\neg lit(b1) \wedge \neg lit(b2) \wedge lit(b3)$$

yields the following set of conflict sets:

$$\{\{b, b1\}, \{b, b2\}, \{b1, b3\}, \{b2, b3\}\}.$$

There are two hitting sets for this collection of conflicts, i.e., two candidate diagnoses: $\{b, b3\}$ and $\{b1, b2\}$.[3]

Suppose that the system requirements under reconfiguration, **reqs**, are somewhat weaker than the original functionality: at least two bulbs should be lit, i.e.,

$$\exists X, Y \; (lit(X) \wedge lit(Y) \wedge X \neq Y),$$

and the candidate diagnosis is $\{b1,\ b2\}$. We add a sentence to the model reflecting the outcome of the diagnosis, i.e., $\mathbf{ab}(b1) \wedge \mathbf{ab}(b2)$ and withdraw the initial status of reconfigurable components, i.e.,

$$\neg rcfg(r1) \wedge \neg rcfg(r2) \wedge \neg rcfg(r3).$$

This gives the set of conflict sets $\{\{b4, b5, b6\}\}$ and the candidate reconfigurations $\Re_{\Delta} = \{b4\} \vee \{b5\} \vee \{b6\}$. In other words, assuming $b3$ is lit and three spare bulbs,

[3] This type of example is typically used to illustrate the necessity of augmenting the correct behavior model (traditionally assumed in model-based diagnosis) with some specification of incorrect behavior, e.g., fault models or physical impossibility axioms. This aspect of the example is irrelevant to our discussion, and we ignore the absurd diagnosis $\{b, b3\}$.

there are three ways to reconfigure the system satisfying the given requirements. Clearly if the reconfiguration requirements specified the original functionality, i.e.,

$$\exists X, Y, Z\,(lit(X) \wedge lit(Y) \wedge lit(Z) \wedge X \neq Y \neq Z),$$

then the set of conflict sets would be $\{\{b4, b5, b6\}, \{b4, b5\}, \{b5, b6\}, \{b4, b6\}\}$ and the candidate reconfigurations $\Re_\Delta = \{b4, b5\} \vee \{b5, b6\} \vee \{b4, b6\}$.

We can use this example to illustrate a further point. Suppose that we have no information about which bulbs are lit and know only that two bulbs have failed. The candidate diagnoses are: $\{b1, b2\}, \{b1, b3\}, \{b2, b3\}$. However, a single reconfiguration, namely $\{r3\}$, satisfies **reqs** and there is no need to further discriminate the diagnoses.

4 Limits of the Analogy: Minimality, Consistency, Entailment

Not surprisingly, the analogy we have pursued thus far has its limits. In this section we look at two particular points in the theory where the parallels between the problem of diagnosis and that of reconfiguration appear to weaken: the role of minimality in diagnosis and in reconfiguration, and the issue of consistency versus entailment.

As noted by [3], most earlier work in model-based diagnosis assumed a "superset property": any superset Δ' of a diagnosis Δ is also a diagnosis. The set of diagnoses can then be parsimoniously represented by the set of minimal diagnoses—those with no proper subsets that are also diagnoses. The algorithms of [13] and most early systems for consistency-based diagnosis construct only the minimal diagnoses and therefore rely on the superset property to ensure that they capture all diagnoses. The superset property can fail, however, with approaches that incorporate models of faulty, as well as correct, behavior.

Two approaches have been suggested for overcoming the inadequacy of minimal diagnoses in these cases [3]: one replaces the notion of minimal diagnosis with that of "kernel" diagnosis, the other places restrictions on the axioms that may appear in the system description so that the notion of minimal diagnosis remains adequate.

Our formulation of reconfiguration is similar to diagnosis with fault models in that the system description contains axioms describing behavior when a component is reconfigured, as well as when it is not. Thus it is not surprising that reconfigurations do not have the superset property: for example, it is not acceptable to reconfigure (i.e., put the spare tire on) *both* the front and back wheels in our bike example.

The question then is: does loss of the superset property matter? Pragmatically, we do not think it does, for we surely prefer to reconfigure as few components as possible and will be satisfied if we can generate the minimal reconfigurations, without worrying about their supersets. Theoretically, though, the problem is more serious because the correctness arguments for Reiter's algorithm [13, pp. 67-68,77] and for the similar algorithm for reconfiguration depend on the superset property. While we do not yet have a definitive resolution for this difficulty, the following seems plausible.

The reason that the bike example fails to have the superset property is because we have the axiom

$$\neg rcfg(front) \vee \neg rcfg(back) \tag{1}$$

that explicitly rules it out. If we remove this axiom, we have a system description that satisfies a condition called LKAB [3] that is sufficient to ensure the superset property. We can therefore safely use Reiter's algorithm to generate all minimal reconfigurations relative to this revised system description. When we come to evaluate the candidate reconfigurations, we first filter them by condition (1).

We suspect that this technique may be quite widely applicable. For the (admittedly very few) examples we have considered so far, the system description can be encoded in axioms satisfying the LKAB condition, plus a few additional axioms that describe inadmissible combinations of reconfigurations that can be used as filters.

We believe that the issue of consistency versus entailment can be resolved by a postpass filter in a similar way. The point here is that by our definition, a satisfactory reconfiguration (relative to a diagnosis) is one that is consistent with the given model, the diagnosis, and the requirements. But is this an adequate characterization? Surely we want to know that the proposed reconfiguration is not merely consistent with the requirements, but will actually achieve (i.e., entail) them. We are sympathetic to this point of view but do not have a good way to satisfy it directly. However, assuming our logical system is sound, we can verify entailment by proving the theorem

$$\begin{aligned} \mathbf{sd}' \cup \{\mathbf{ab}(c) | c \in \Delta\} \cup \{\neg \mathbf{ab}(c) | c \in \mathbf{comps} - \Delta\} \\ \cup \{\mathbf{rcfg}(c) | c \in \Re_\Delta\} \cup \{\neg \mathbf{rcfg}(c) | c \in \mathbf{comps}' - \Re_\Delta\} \\ \vdash \ \mathbf{reqs} \end{aligned} \tag{2}$$

Thus (2) can be added to superset constraints such as (1) as a further filter on acceptable reconfigurations. Note that if (2) is not a theorem, then the **sd** is surely rather weak, since it fails to adequately constrain the behavior of the system. A topic for further investigation is to determine whether constraints on the forms of axioms comprising the **sd** can be found that are sufficient to ensure entailment of requirements.

5 Related Work

Poole [12, p. 1310] has noted the generality of the model-based paradigm and suggested its applicability to a large class of recognition problems including planning. There have also been extensions to Reiter's algorithm, such as the work of Ng [11], which extends the algorithm to handle time-varying, physical devices. However, we know of no attempts to extend model-based diagnosis to accommodate FDIR, with the possible exception of the work of Friedrich and colleagues [6], who define a notion of "therapy" and sketch an algorithm for "the standard therapeutic approach." The latter can be characterized as a process of interleaving diagnosis and repair to suppress "undesired symptoms." This approach differs from ours in that it eliminates or repairs only those components whose treatment causes the disappearance of the observed symptoms; it assumes that granularity of reconfiguration is precisely that of diagnosis, i.e., the reconfigurable units are the same as the diagnosable units; and it assumes that the level of acceptable system functionality remains constant from diagnosis to reconfiguration.

6 Conclusions and Future Directions

We have proposed a characterization of reconfiguration as an extension of Reiter's theory of model-based diagnosis. Our contribution has been to recognize and exploit an analogy between the problem of model-based diagnosis and that of reconfiguration. The simplicity of this analogy suggests that it should be possible to use an existing diagnosis engine to compute reconfigurations. Hamscher's report [10] on the satisfactory mechanization of our light bulb example using an experimental diagnostic system is encouraging in this regard.

In order to realize the benefits claimed in the introduction, we need to develop methods for interleaving diagnosis and reconfiguration. The simplest approach requires massive iteration: for each candidate diagnosis, and for each acceptable behavior, compute the reconfigurations which achieve that behavior. If all diagnoses yield to a single reconfiguration, we are done; if not, we need methods for eliminating candidate diagnoses and reconfigurations. We plan to explore focusing strategies and other techniques for avoiding this unacceptably large number of iterations and exploring the combined space of diagnoses and reconfigurations more efficiently.

We also plan to evaluate our approach experimentally, and to examine the practical feasibility of using postpass filters to deal with the superset and entailment problems. Finally, we hope to explore connections between our work and Friedrich's therapeutic approach and to consider alternative characterizations of reconfiguration within an abductive framework.

6.1 Acknowledgments

The contents of this paper benefitted from several discussions with Walter Hamscher, who generously volunteered to run our light bulb example through one of his diagnosis systems, and from the constructive comments of two anonymous referees.

References

[1] K. Abbott. Robust operative diagnosis as problem solving in a hypothesis space. In *Proceedings, AAAI88*, pages 369–374, St. Paul, MN, 1988.

[2] R. Davis. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24(1-3):347–410, December 1984.

[3] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnoses and systems. Technical Report SSL-90-40, Xerox Palo Alto Research Center, 1990.

[4] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, April 1987.

[5] J. de Kleer and B. C. Williams. Diagnosis with behavioral modes. In *Proceedings, 11th IJCAI*, pages 1324–1330, Detroit, MI, August 1989.

[6] G. Friedrich, G. Gottlob, and W. Nejdl. Hypothesis classification, abductive diagnosis and therapy. In G. Gottlob and W. Nejdl, editors, *Expert Systems in Engineering*, pages 69–78. Springer-Verlag Lecture Notes in Artificial Intelligence Vol. 462 (International Workshop Proceedings), Vienna, Austria, September 1990.

[7] G. Friedrich, G. Gottlob, and W. Nejdl. Physical impossibility instead of fault models. In *Proceedings, AAAI 90 (Volume 1)*, pages 331–336, Boston, MA, July 1990.

[8] M.R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, 24(1-3):411–436, December 1984.

[9] W. Hamscher. Temporally coarse representation of behavior for model-based troubleshooting of digital circuits. In *Proceedings, 11th IJCAI*, pages 887–893, Detroit, MI, August 1989.

[10] W. Hamscher. Private communication, 1991.

[11] H. T. Ng. Model-based, multiple fault diagnosis of time-varying, continuous physical devices. In *Proceedings, 6th IEEE Conference on AI Applications*, pages 9–15, Santa Barbara, CA, 1990.

[12] D. Poole. Normality and faults in logic-based diagnosis. In *Proceedings, 11th IJCAI*, pages 1304–1310, Detroit, MI, August 1989.

[13] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, April 1987.

[14] P. Struss and O. Dressler. "Physical negation"—integrating fault models into the general diagnostic engine. In *Proceedings, 11th IJCAI*, pages 1318–1323, Detroit, MI, August 1989.