Explanatory Diagnosis: Conjecturing actions to explain observations

Sheila A. McIlraith[*]

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94301
mcilrait@parc.xerox.com

Knowledge Systems Laboratory
Stanford University
Stanford, CA 94305-9020
sam@ksl.stanford.edu

Abstract

Our concern in this paper is with conjecturing diagnoses to explain *what happened* to a system, given a theory of system behaviour and some observed (aberrant) behaviour. We characterize what happened by introducing the notion of explanatory diagnoses in the language of the situation calculus. Explanatory diagnoses conjecture sequences of actions to account for a change in system behaviour. We show that determining an explanatory diagnosis is analogous to the classical AI planning task. As such, we exploit previous results on goal regression in the situation calculus to show that determining an explanatory diagnosis can be achieved by regression followed by theorem proving in the database describing what is known of the initial state of our system. Further, we show that in the case of incomplete information, determining explanatory diagnoses is an abductive plan synthesis task.

Introduction

Given a theory of system behaviour and some observed aberrant behaviour, the traditional objective of diagnosis is to conjecture *what is wrong* with the system, (e.g., which components of the device are behaving abnormally, what diseases the patient is suffering from, etc.). Each candidate diagnosis consists of a subset of distinguished literals that are conjectured to be true or false in order to account for the observation in some way. Different criteria have been proposed for determining the space of such candidate diagnoses. Within formal accounts of diagnosis, two widely accepted definitions of diagnosis are consistency-based diagnosis (e.g., (Reiter 1987), (de Kleer, Mackworth, & Reiter 1992)), and abductive explanation (e.g., (de Kleer, Mackworth, & Reiter 1992), (Poole 1988), (Console & Torasso 1991), (McIlraith 1994a)).

Our concern in this paper is with conjecturing diagnoses to explain *what happened* to a system, given a theory of system behaviour and some observed (aberrant) behaviour (i.e., what actions or events occurred to result in the observed behaviour) (e.g., (McIlraith 1994b), (Cordier & Thiebaux 1994)). Knowing or conjecturing what happened is interesting in its own right, but can also assist in the prediction of abnormal components or other relevant behaviour, and in the prescription of suitable procedures for testing, repairing or reacting. Compared to our traditional notion of *what is wrong* diagnoses, knowing *what happened* can more accurately capture the root cause of system malfunction, rather than its manifestations.

In the spirit of previous foundational work in model-based diagnosis (MBD) (e.g., (Reiter 1987), (Console & Torasso 1991), (de Kleer, Mackworth, & Reiter 1992)), this paper presents a mathematical characterization for the notion of explanatory diagnosis. We take as our starting point the existing MBD research on characterizing diagnoses for static systems *without* a representation of actions (e.g., (de Kleer, Mackworth, & Reiter 1992), (Console & Torasso 1991), (Reiter 1987)). Next, we exploit a situation calculus representation scheme previously proposed by the author (McIlraith 1997a) that enables the integration of a representation of action with the representation of the behaviour of a static system. With this representation in hand, we provide a logical characterization for the task of determining *what happened* to a system. The characterization is presented in the guise of *explanatory diagnosis*.

The distinguishing features of our characterization are afforded in great part by the richness of our representation scheme which provides a comprehensive and semantically justified representation of action and change. In particular, our representation provides an axiomatic closed-form solution to the frame and ramification problems, thus capturing the direct and indirect effects of actions in a compiled representation. This is critical to the ability to generate explanatory diagnoses efficiently. Further, our representation provides a closed-form solution to the qualification problem, thus identifying the conditions underwhich an action is possible. It is interesting to note that when we are dealing with incomplete knowledge of our initial state, conjecturing an action or sequence of actions also requires conjecturing that its preconditions are satisfied, which in many instances serves to further constrain our search.

[*] This work was carried out while the author was a doctoral student at the University of Toronto, Canada.

As we show in the sections to follow, our characterization establishes a direct link between explanatory diagnosis and planning, deductive plan synthesis and abductive planning. As a consequence of a completeness assumption embedded in our representation, we show how to exploit goal-directed reasoning in the form of regression (Waldinger 1977) in order to generate diagnoses. This completeness assumption also provides for an easy mapping of our situation calculus representation to Prolog.

Representation Scheme

Situation Calculus Language

The situation calculus language we employ to axiomatize our domains is a sorted first-order language with equality. The sorts are of type $\mathcal{A}$ for primitive *actions*, $\mathcal{S}$ for *situations*, and $\mathcal{D}$ for everything else, including domain objects (Lin & Reiter 1994). We represent each action as a (possibly parameterized) first-class object within the language. Situations are simply sequences of actions. The evolution of the world can be viewed as a tree rooted at the distinguished initial situation S_0. The branches of the tree are determined by the possible future situations that could arise from the realization of particular sequences of actions. As such, each situation along the tree is simply a history of the sequence of actions performed to reach it. The function symbol do maps an action term and a situation term into a new situation term. For example, $do(turn\_on\_pmp, S_0)$ is the situation resulting from performing the action of turning on the pump in situation S_0. The distinguished predicate $Poss(a, s)$ denotes that an action a is possible to perform in situation s (e.g., $Poss(turn\_on\_pmp, S_0)$). Thus, $Poss$ determines the subset of the situation tree consisting of situations that are possible in the world. Finally, those properties or relations whose truth value can change from situation to situation are referred to as *fluents*. For example, the fluent $on(Pmp, s)$ expresses that the pump is on in situation s.

The situation calculus language we employ in this paper is restricted to primitive, determinate actions. Our language does not include a representation of time, concurrency, or complex actions, but we believe the results presented herein can be extended to more expressive dialects of the situation calculus (e.g., (Reiter 1996)) without great difficulty.

Domain Representation: An Example

In this section we briefly describe the representation scheme we use to characterize our system. The scheme, proposed in (McIlraith 1997a), integrates a situation calculus theory of action with a MBD system description, SD (de Kleer, Mackworth, & Reiter 1992). The resultant representation of a system comprises both domain-independent and domain-specific axioms. The domain-independent axioms are the foundational axioms of the discrete situation calculus, Σ_{found} (Lin & Reiter 1994). They are analogous to the axioms of Peano arithmetic, modified to define the branching structure of our situation tree, rather than the number line. The domain-specific axioms, T specify both the *behaviour of the static system*, and the *actions*[1] that can affect the state of the system, as well as those actions required to achieve testing and repair. Together they define our domain representation $\Sigma = \Sigma_{found} \wedge T$.

We determine T using a procedure proposed in (McIlraith 1997a) that compiles a typical MBD system description, SD and a set of axioms relating to the preconditions and effects of actions into a representation that provides a closed-form solution to the frame, ramification and qualification problems. The resultant domain axiomatization $T = T_{SC}^{S_0} \wedge T_{domain} \wedge T_{SS} \wedge T_{AP} \wedge T_{UNA} \wedge T_{DCA} \wedge T_{S_0}$ is described below. The representation is limited to a syntactically restricted but commonly occurring class of theories called solitary stratified theories (McIlraith 1997a). Intuitively, the dependency graphs of the actions and state constraints of these theories contain no loops or cycles. It is also important to note that a completeness assumption is embedded in this representation. The assumption states that all the conditions underwhich an action a can lead, directly or indirectly, to fluent F becoming true or false in the successor state are captured in the axiomatization of our system.

We illustrate the representation in terms of a small portion of a power plant feedwater system (McIlraith 1997b) derived from the APACS project (Kramer & et al. 1996). Our example models the filling of a vessel either by the operation of an electrically powered (Pwr) pump (Pmp), or by manual filling. For notational convenience, all formulae are understood to be universally quantified with respect to their free variables, unless explicitly indicated otherwise. For a more thorough description of this representation scheme, please see ((McIlraith 1997a), (McIlraith 1997b)).

The set of state constraints relativized to situation S_0, $T_{SC}^{S_0}$ is as follows. These constraints could be acquired from a typical MBD system description, SD.

$$\neg AB(Pwr, S_0) \wedge \neg AB(Pmp, S_0) \wedge on(Pmp, S_0) \supset filling(S_0) \qquad (1)$$

$$manual\_fill(S_0) \supset filling(S_0) \qquad (2)$$

The set of domain constraints, T_{domain} is as follows.

$$Pwr \neq Pmp \qquad (3)$$

The set of successor state axioms, T_{SS} is composed of axioms of the following general form, one for each fluent F.

$$Poss(a, s) \supset [F(do(a, s)) \equiv \Phi_F] \qquad (4)$$

[1] Actions can be performed by agents: a human, another system, or nature.

where Φ_F is a simple formula[2] of a particular syntactic form. E.g.,

$$Poss(a,s) \supset [on(Pmp, do(a,s)) \equiv a = turn\_on\_pmp \\ \vee (on(Pmp,s) \wedge a \neq turn\_off\_pmp)] \quad (5)$$

$$Poss(a,s) \supset [AB(Pwr, do(a,s)) \equiv a = pwr\_failure \\ \vee (AB(Pwr,s) \wedge a \neq aux\_pwr \wedge a \neq pwr\_fix)] \quad (6)$$

$$Poss(a,s) \supset [AB(Pmp, do(a,s)) \equiv \\ a = pmp\_burn\_out \\ \vee (AB(Pmp,s) \wedge a \neq pmp\_fix)] \quad (7)$$

$$Poss(a,s) \supset [manual\_fill(do(a,s)) \equiv \\ a = turn\_on\_manual\_fill \\ \vee (manual\_fill(s) \\ \wedge a \neq turn\_off\_manual\_fill)] \quad (8)$$

$$Poss(a,s) \supset [filling(do(a,s)) \equiv \\ a = turn\_on\_manual\_fill \\ \vee \ (manual\_fill(s) \wedge a \neq turn\_off\_manual\_fill) \\ \vee \ [(a \neq pwr\_failure \\ \wedge \ (\neg AB(Pwr,s) \vee a = aux\_pwr \\ \vee a = pwr\_fix)) \\ \wedge \ (a \neq pmp\_burn\_out \\ \wedge \ (\neg AB(Pmp,s) \vee a = pmp\_fix)) \\ \wedge \ (a = turn\_on\_pmp \\ \vee \ (on(Pmp,s) \wedge a \neq turn\_off\_pmp))] \\ \vee \ (filling(s) \wedge a \neq stop\_siphon)] \quad (9)$$

Axiom (5) states that if action a is possible in situation s, then the pump is on in the situation resulting from performing action a in situation s (i.e., $on(Pmp, do(a,s))$) if and only if the action a is $turn\_on\_pmp$, or the pump was already on in s and a was not the action $turn\_off\_pmp$.

The set of action precondition axioms, T_{AP} is composed of axioms of the following general form, one for each action prototype A in the domain.

$$Poss(A(\vec{x}), s) \equiv \Pi_A \quad (10)$$

where Π_A is a simple formula with respect to s.

$$Poss(stop\_siphon, s) \equiv (\neg manual\_fill(s) \\ \wedge \neg on(Pmp,s)) \quad (11)$$

$$Poss(pmp\_fix, s) \equiv \neg on(Pmp,s) \quad (12)$$

$$Poss(pmp\_burn\_out, s) \equiv on(Pmp,s) \quad (13)$$

$$Poss(turn\_on\_manual\_fill, s) \equiv \neg on(Pmp,s) \quad (14)$$

$$Poss(turn\_on\_pmp, s) \equiv \neg manual\_fill(s) \quad (15)$$

$$Poss(turn\_off\_pmp, s) \equiv Poss(pwr\_failure, s) \equiv \quad (16)$$

$$Poss(pwr\_fix, s) \equiv Poss(aux\_pwr, s) \equiv \quad (17)$$

$$Poss(turn\_off\_manual\_fill, s) \equiv true \quad (18)$$

[2] A *simple formula* only mentions domain-specific predicate symbols, fluents do not include the function symbol do, there is no quantification over sort $situation$, and there is at most one free $situation$ variable.

Finally, we provide a possible set of initial conditions for our system. These constitute the initial database, T_{S_0}. Note that in general we do not have complete knowledge of the initial state of our system. This makes the task of diagnosis all the more challenging. In this example, we do not know initially whether the pump and power are operating normally. We also do not know whether the vessel was filling in the initial state.

$$on(Pmp, S_0) \wedge \neg manual\_fill(S_0) \quad (19)$$

In the interest of space, we do not show the unique names axioms for actions, T_{UNA} and the domain closure axiom for actions, T_{DCA}.

Relationship to Logic Programming

It is interesting to note that our proposed situation calculus representation can be viewed as an executable specification because it is easily realized in Prolog by exploiting Prolog's completion semantics and simply replacing the equivalence connectives characteristic of axioms in T_{SS} and T_{AP} by implication connectives. The Lloyd-Topor transformation (Lloyd 1987) must then be applied to convert this theory into Prolog clausal form. Later in this paper, we will advocate using Waldinger's notion of regression to rewrite axioms of our representation and simplify computation. This type of regression rewriting is precisely achieved by Prolog's backwards chaining mechanism.

Preliminaries

With our representation in hand, we turn our attention to the task of diagnosis. In this section we introduce the framework for performing diagnosis relative to our representation. For our purposes we adopt the ontological and notational convention of the MBD literature and view the systems we are diagnosing as comprising a number of interacting *components*, $COMPS$. These components have the property of being either abnormal or normal in a situation. We express this property in our situation calculus language using the fluent AB. For example, $AB(Pmp, s)$ denotes that the pump component is abnormal in situation s. Note that the use of AB is not mandatory to the contributions of this paper. Once again, following the convention in the MBD literature, we define our diagnoses relative to the domain-independent concept of a *system* (de Kleer, Mackworth, & Reiter 1992), adapted to our situation calculus framework.

Definition 1 (System)

A system is a quadruple $(\Sigma, HIST, COMPS, OBS)$ *where:*

- Σ*, the background theory, is a set of situation calculus sentences describing the behaviour of our system and the actions that can affect it.*
- $HIST$*, the history, is a sequence of ground actions* $[a_1, \ldots, a_k]$ *that were performed starting in* S_0.

- *$COMPS$, the components, is a finite set of constants.*
- *OBS_F, the observation, is a simple formula composed of fluents whose only free variable is the situation variable s, and which are otherwise ground.*

Example 1
In our power plant example above, Σ is our axiomatization $\Sigma_{found} \wedge T$ and $COMPS = \{Pmp, Pwr\}$. The observation, OBS_F could be $filling(s)$, for example. $HIST$ could be empty, i.e., [ ], or perhaps $[turn\_on\_pmp]$.

Explanatory Diagnosis

In this section we introduce and formally characterize the notion of an explanatory diagnosis which conjectures *what happened* to result in some observed (aberrant) behaviour. In particular, given a system, (Σ, $HIST$, $COMPS$, OBS_F), the objective of explanatory diagnosis is to conjecture a sequence of actions, $[\alpha_1, \ldots, \alpha_n]$ such that our observation is true in the situation resulting from performing that sequence of actions in $do(HIST, S_0)$. Since we may have incomplete information about the initial state of our system, we also provide characterizations of weaker forms of explanatory diagnosis, which we propose to aid in the search for diagnoses. Finally, we exploit the preference criterion of chronological simplicity to define a preferred subset of our explanatory diagnoses.

Characterizing Explanatory Diagnosis

The problem of determining explanatory diagnoses is an instance of temporal explanation or postdiction (e.g., (Shanahan 1993)), and is related to the classical AI planning problem, as we see below and in the section to follow.

Definition 2 (Explanatory Diagnosis)
An explanatory diagnosis for system (Σ, $HIST$, $COMPS$, OBS_F) is a sequence of actions $E = [\alpha_1, \ldots, \alpha_n]$ such that,

- $\Sigma \models Poss(HIST \cdot E, S_0)^2 \wedge OBS_F(do(HIST \cdot E, S_0))$.

Thus E is an explanatory diagnosis if the observation is true in the situation resulting from performing the sequence of actions E in situation $do(HIST, S_0)$, and further that the preconditions for each action of the action sequence $HIST \cdot E$ are true in the appropriate situations, commencing at S_0.

Identifying the sequence of actions composing an explanatory diagnosis, E is analogous to the plan synthesis problem, and thus is realizable using deduction on the situation calculus axioms. According to Green (Green 1969), a plan to achieve a goal $G(s)$ is obtained as a side effect of proving $Axioms \models \exists s.G(s)$. The bindings for the situation variable s represent the sequence of actions. In our case, $Axioms \models \exists s.G(s)$ is analogous to $\Sigma \models \exists s.OBS_F(s)$. As such, our representation enables us to generate explanatory diagnoses deductively, just as we could deductively generate a plan in the situation calculus.

Example 2
Continuing with our power plant example, given the system (Σ, [], $\{Pwr, Pmp\}$, $\neg filling(s)$), the sequence of actions $[pwr\_failure]$ constitutes one example of an explanatory diagnoses for the system. Another explanatory diagnosis for our system is $[turn\_off\_pmp]$.

Observe that for certain problems there can be an infinite number of sequences of actions that constitute explanatory diagnoses. For example, the following sequences of actions also constitute valid explanatory diagnoses for our example system:
$[pwr\_failure, pwr\_fix, pwr\_failure]$,
$[pwr\_failure, pwr\_aux, pwr\_failure]$,
$[turn\_off\_pmp, pwr\_failure, turn\_on\_pmp]$,
and so on.

Definition 2 is not sufficiently discriminating to eliminate these, clearly suboptimal explanatory diagnoses. We must define a preference criterion. Probability measures, even simple order of magnitude probabilities have provided an effective preference criterion for many applications of MBD (de Kleer 1991). Likewise, we believe that in the case of determining explanatory diagnoses in the context of the situation calculus, probabilities will serve us well in identifying preferred explanatory diagnoses. Unfortunately, probability measures are not always available. In this paper, we limit our discussion to what we refer to as a chronologically simple preference criterion.

In our chronologically simple preference criterion, we prefer diagnoses that are relativized to situations reached without performing any extraneous actions. Note that this preference criterion is syntactic in nature, relying on the notion of a primitive action as a unit measure.

Definition 3 (Simpler)
Given a sequence of actions $HIST = [\alpha_1, \ldots, \alpha_n]$, define $ACTS(HIST)$ to be the set $\{\alpha_1, \ldots, \alpha_n\}$, and $LEN(HIST)$ to be the the length of the sequence of actions composing $HIST$.

Thus, given $HISTA = [a_1, \ldots, a_n]$ and $HISTB = [b_1, \ldots, b_n]$, situation $S_A = do(HISTA, S_0)$ is simpler than situation $S_B = do(HISTB, S_0)$ iff $ACTS(HISTA) \subseteq ACTS(HISTB)$ and $LEN(HISTA) < LEN(HISTB)$.

[2] **Notation:**
$HIST \cdot E$ is an abbreviation for $[a_1, \ldots, a_k, \alpha_1, \ldots, \alpha_n]$.
$do([a_1, \ldots, a_m], s)$ is an abbreviation for $do(a_m,(do(a_{m-1},(do(a_{m-2},(\ldots,(do(a_1,s))))))))$.
Finally, $Poss([a_1, \ldots, a_n], s)$ is an abbreviation for $Poss(a_1, s) \wedge Poss(a_2, do(a_1, s)) \wedge \ldots \wedge Poss(a_n, do([a_1, \ldots, a_{n-1}], s))$.

Definition 4 (Chronologically Simple Expl. Diagnosis) *E is a chronologically simple explanatory diagnosis for system* $(\Sigma, HIST, COMPS, OBS_F)$ *iff E is an explanatory diagnosis for the system, and there is no explanatory diagnosis E' such that situation $S' = do(HIST \cdot E', S_0)$ is simpler than situation $S = do(HIST \cdot E, S_0)$.*

We might further distinguish this criterion to prefer chronologically simple explanatory diagnoses where the actions are limited to those, for example, performed by nature. It is no doubt possible to provide a more general and formal account of explanatory diagnosis in terms of circumscription. Nevertheless the account provided serves our purposes for diagnosis, so we leave this issue, and the more pragmatic issue of exploiting probabilities, for future work.

Observe that the characterization of explanatory diagnosis just presented assumes that E and OBS_F occur *after* $HIST$. While this assumption is not critical to characterizing explanatory diagnoses, it acts as a form of preference, facilitating computation of E.

Dealing with Incomplete Information

Note that in Example 2, we do not have complete information about the initial state of our system. It could be the case that observation $\neg filling$ was true in S_0, i.e., $\neg filling(S_0)$, but it is simply not entailed by Σ. Consequently, the empty action sequence is not a valid explanatory diagnosis, and we must conjecture a sequence of actions that make our observation true. To accommodate a lack of information about the initial state, we may instead wish to generate explanations by assuming additional information about the world, and making our explanations conditioned on these assumptions. We capture this intuition in an assumption-based explanatory diagnosis.

Definition 5 (Assumption-based Expl. Diagnosis) *Given an assumption $H(S)$ relativized to ground situation S such that*

- $S_0 \leq$ [3] $S \leq do(HIST \cdot E, S_0)$,
- $\Sigma \wedge H(S)$ *is satisfiable, and*
- $\Sigma \wedge H(S) \models Poss(HIST, S_0)$.

An assumption-based explanatory diagnosis for system $(\Sigma, HIST, COMPS, OBS_F)$ *under assumption $H(S)$ is a sequence of actions $E = [\alpha_1, \ldots, \alpha_k]$ such that,*

- $\Sigma \wedge H(S) \models Poss(HIST \cdot E, S_0) \wedge OBS_F(do(HIST \cdot E, S_0))$.

[3] **Notation:** The transitive binary relation $<$ defined in Σ_{found} further limits our situation tree by restricting the actions that are applied to a situation to those whose preconditions are satisfied in the situation. Intuitively, if $s < s'$, then s and s' are on the same branch of the tree with s closer to S_0 than s'. Further, s' can be obtained from s by applying a sequence of actions whose preconditions are satisfied by the truth of the $Poss$ predicate.

In some instances, we may want to make a priori assumptions about the world, conjoin these assumptions to our theory and then try to compute our explanatory diagnoses. For example, we may wish to assume that all components are operating normally in S_0, if this is consistent with our theory and action history. This would be achieved by making $H(S)$ in our definition above equal to $\bigwedge_{c \in COMPS} \neg AB(c, S_0)$ (i.e., $\neg AB(Pmp, S_0) \wedge \neg AB(Pwr, S_0)$). Similarly, we may wish to assume that the observation, OBS_F is true in $do(HIST, S_0)$, if this can be consistently assumed. In our example above, this would mean assuming $\neg filling(S_0)$.

In still other instances, we may not want to fix our assumptions a priori but rather make a minimum number of assumptions, as necessary to generate an explanatory diagnosis with a minimal number of actions. Such assumptions might be limited to a distinguished set of literals, which the domain axiomatizer considers to be legitimately assumable (e.g., AB fluents).

Finally, we observe that the requirement in Definition 2 and Definition 5 that $\Sigma \models Poss(HIST \cdot E, S_0)$ may be too stringent in the case of an incomplete initial database (i.e., it may not be reasonable to require that we know that an action is possible in a situation that is incompletely specified). We may prefer to consider explanatory diagnoses, where the theory allows us to consistently assume that the preconditions for $HIST$ or for $HIST \cdot E$ hold, but not necessarily that they are entailed by our theory. To this end, we propose the following alteration on our definition of explanatory diagnoses. A comparable refinement can be made to our definition of assumption-based explanatory diagnosis.

Definition 6 (Potential Explanatory Diagnosis) *A potential explanatory diagnosis for system* $(\Sigma, HIST, COMPS, OBS_F)$ *is a sequence of actions $E = [\alpha_1, \ldots, \alpha_k]$ such that,*

- $\Sigma \wedge Poss(HIST \cdot E, S_0)$ *is satisfiable, and*
- $\Sigma \wedge Poss(HIST \cdot E, S_0) \models OBS_F(do(HIST \cdot E, S_0))$.

Note in particular, that $HIST$ is a set of actions that we know to have been performed starting in situation S_0. This also tells us that the preconditions for each of the actions in $HIST$ were true in the corresponding situations, providing us with further information concerning the truth values of fluents at various situation along the situation tree.

Exploiting Regression

In the previous section, we provided characterizations of explanatory diagnosis, given a potentially incomplete initial state. Upon first glance, the general problem of computing explanatory diagnoses does not look very promising because of the second-order induction axiom in Σ_{found}, and the potentially large size of the situation search space. In this section, we show how diagnoses can be computed by exploiting regression. Given a system (Σ, $HIST$, $COMPS, OBS_F$), we are interested in finding a sequence of actions E such that $\Sigma \models Poss(HIST \cdot E, S_0) \wedge OBS_F(do(HIST \cdot E, S_0))$. Generating a sequence of actions that constitutes an explanatory diagnosis for an observation OBS_F is analogous to generating a sequence of actions that constitutes a plan to achieve a goal $OBS_F(s)$. The sequence of actions following $HIST$ that determine s constitutes an explanatory diagnosis E.

As is commonly done in planning tasks, we advocate exploiting regression (e.g., (Waldinger 1977)) to generate explanatory diagnoses. In this context, regression is a recursive rewriting procedure used to reduce the nesting of the do function in situation terms. We will show that generating explanatory diagnoses reduces to regression followed by entailment with respect to the initial database. Computationally, the merit of regression is that it searches backwards through the situation space from the observation rather than searching forward from the initial database. Under the assumption that the observation consists of fewer literals than the initial database, regression will make for more efficient search. Observe that Prolog's backwards chaining mechanism achieves the substitution performed by regression.

Regression

In earlier work, Reiter proved soundness and completeness results for regression (Reiter 1992b). We exploit these results in our treatment of explanatory diagnosis. To that end, we define two regression operators, $\mathcal{R}^*$ and $\mathcal{R}_{Poss}$. Following in the spirit of (Reiter 1991) and (Reiter 1992b),

Definition 7 (Regression Operator $\mathcal{R}^*$)
Given a set of successor state axioms, T_{SS} composed of axioms of the following form

$$Poss(a, s) \supset [F(do(a, s)) \equiv \Phi_F], \tag{20}$$

$\mathcal{R}^*[\Psi]$*, the repeated regression of formula Ψ with respect to successor state axioms T_{SS} is the formula that is obtained from Ψ by repeatedly replacing each fluent atom $F(do(a, s))$ in Ψ by Φ_F, until the resulting formula makes no mention of the function symbol do.*

We can similarly define a $Poss$ regression operator over the set of action precondition axioms, T_{AP}. This regression operation rewrites each occurrence of the literal $Poss(a, s)$ by Π_A as defined in the action precondition axioms.

Definition 8 ($Poss$ Regression Operator)
Given a set of action precondition axioms, T_{AP} composed of axioms of the form

$$Poss(A(\vec{x}), s) \equiv \Pi_A, \tag{21}$$

$\mathcal{R}_{Poss}[W]$ *is the formula obtained by replacing each occurrence of predicate $Poss(A(\vec{x}), s)$ by Π_A. All other literals of W remain the same.*

Hence,

$$\mathcal{R}_{Poss}[Poss(A(\vec{x}), s)] = \Pi_A, \tag{22}$$

and for formulae W, W_1 and W_2,

$$\mathcal{R}_{Poss}[\neg W] = \neg \mathcal{R}_{Poss}[W], \tag{23}$$
$$\mathcal{R}_{Poss}[(\forall v)W] = (\forall v)\mathcal{R}_{Poss}[W], \tag{24}$$
$$\mathcal{R}_{Poss}[(\exists v)W] = (\exists v)\mathcal{R}_{Poss}[W]. \tag{25}$$
$$\mathcal{R}_{Poss}[W_1 \wedge W_2] = \mathcal{R}_{Poss}[W_1] \wedge \mathcal{R}_{Poss}[W_2], \tag{26}$$
$$\mathcal{R}_{Poss}[W_1 \vee W_2] = \mathcal{R}_{Poss}[W_1] \vee \mathcal{R}_{Poss}[W_2], \tag{27}$$
$$\mathcal{R}_{Poss}[W_1 \supset W_2] = \mathcal{R}_{Poss}[W_1] \supset \mathcal{R}_{Poss}[W_2], \tag{28}$$
$$\mathcal{R}_{Poss}[W_1 \equiv W_2] = \mathcal{R}_{Poss}[W_1] \equiv \mathcal{R}_{Poss}[W_2]. \tag{29}$$

Finally for any formula W containing no occurrence of predicate $Poss$,

$$\mathcal{R}_{Poss}[W] = W. \tag{30}$$

Next we adapt Reiter's results on the soundness and completeness of regression (Theorem 1, Theorem 2, (Reiter 1992b)) to our representation. They are presented in the following proposition. The theory Σ_{init} mentioned in the proposition below is a subset of Σ containing only the initial database, and no information about successor situations. It also excludes the induction axiom in Σ_{found}.

Proposition 1 (Soundness & Completeness)
Given

- Σ_{init}*, a subset of the situation calculus theory Σ, such that $\Sigma_{init} = \Sigma_{UNS} \wedge T_{S_0} \wedge T_{ram}^{S_0} \wedge T_{domain} \wedge T_{UNA}$, where Σ_{UNS} is a subset of Σ_{found} containing the set of unique names axioms for situations.*
- *a sequence of ground actions, s_HIST such that $\Sigma_{init} \wedge \mathcal{R}^*\ [\mathcal{R}_{Poss}\ [Poss(s\_HIST,\ S_0)]]$ is satisfiable.*
- *$Q(s)$, a simple formula whose only free variable is the situation variable s.*

Suppose $S = do(s\_HIST, S_0)$, then

- *$\Sigma \models Q(do(s\_HIST, S_0))$ iff $\Sigma_{init} \models \mathcal{R}^*[Q(do(s\_HIST, S_0))]$,*
- *$\Sigma \models Poss(s\_HIST, S_0)$ iff $\Sigma_{init} \models \mathcal{R}^*[\mathcal{R}_{Poss}[Poss(s\_HIST, S_0)]]$,*
- *$\Sigma \wedge Poss(s\_HIST, S_0) \wedge Q(do(s\_HIST, S_0))$ is satisfiable iff $\Sigma_{init} \wedge \mathcal{R}^*[\mathcal{R}_{Poss}[Poss(s\_HIST, S_0)]] \wedge \mathcal{R}^*[Q(do(s\_HIST, S_0))]$ is satisfiable,*

Thus, assuming situation s is a possible situation and exploiting regression, $Q(s)$ holds at situation s iff its regression is entailed in the initial database. The beauty of proposition 1 is that it enables us to generate explanatory diagnoses via regression followed by theorem proving in the initial database, without the need for the second-order induction axiom in Σ_{found}.

From these results, we can characterize explanatory diagnosis with respect to regression.

Proposition 2 (Expl. Diagnosis with Regression) *The sequence of actions $E = [\alpha_1, \ldots, \alpha_k]$ is an explanatory diagnosis for system $(\Sigma,\ HIST,\ COMPS,\ OBS_F)$ iff*

$$\Sigma_{init} \models \mathcal{R}^*[\mathcal{R}_{Poss}[Poss(HIST \cdot E, S_0)]] \\ \wedge \mathcal{R}^*[OBS_F(do(HIST \cdot E, S_0))]. \qquad (31)$$

There are obvious analogues for assumption-based explanatory diagnoses and potential explanatory diagnoses. We do not restate them here.

Determining Diagnoses

Different applications will use these characterizations of diagnoses in different ways, to meet the needs of the specific domain. In this section we examine two such uses: verifying a given diagnosis and generating diagnoses.

Verifying a Diagnosis

For many systems, particularly those that have an incomplete initial database, it may be pragmatic to maintain a library of most likely diagnoses and attempt to verify or refute these diagnoses in order of descending likelihood. These libraries could be indexed by observations and/or situation histories. The diagnoses themselves could be sequences of actions and possibly assumptions.

Given a system $(\Sigma,\ HIST,\ COMPS,\ OBS_F)$, and a candidate diagnosis E, such that $S = do(HIST \cdot E, S_0)$, we are interested in verifying that E is indeed a diagnosis of the system. Verifying this candidate diagnosis is simply a query evaluation problem. It can be accomplished by regression and theorem proving in the initial database, as per Proposition 2 above.

Example 3 *Given the system $(\Sigma,\ [\ ],\ \{Pwr, Pmp\}, \neg filling(s))$, and the candidate diagnosis E=$[pwr\_failure]$, E can be verified to be an explanatory diagnosis with respect to the system by evaluating the query*

$$\mathcal{R}^*[\mathcal{R}_{Poss}[Poss(do(pwr\_failure, S_0))]] \\ \wedge \mathcal{R}^*[\neg filling(do(pwr\_failure, S_0))]$$

with respect to the initial database, Σ_{init}.

Again, verifying such a diagnosis in Prolog is straightforward through exploitation of Prolog's backwards chaining mechanism which is analogous to regression.

Generating Diagnoses

In contrast, to generate diagnoses for a a system $(\Sigma,\ HIST,\ COMPS, OBS_F)$, we are interested in finding a sequence of actions E such that $\Sigma \models Poss(HIST \cdot E,\ S_0) \wedge OBS_F(do(HIST \cdot E,\ S_0))$. As observed previously, generating a sequence of actions that constitutes an explanatory diagnosis for an observation OBS_F is identical to generating a sequence of actions that constitutes a plan to achieve a goal $\exists s.OBS_F(s)$. Thus, following the work on deductive plan synthesis (e.g., (Green 1969)), we can view the generation of explanatory diagnoses as a deductive plan synthesis problem that is realizable using theorem proving. As a side effect of proving $\exists s.OBS_F(s)$, a theorem prover will generate bindings for s. The sequence of actions following $HIST$ that determine s constitutes an explanatory diagnosis E. Further, following Proposition 2 of the previous section, we can achieve such an explanatory diagnosis by regression and theorem proving. In trying to prove $\exists s.OBS_F(s)$, Prolog's backwards chaining mechanism takes precisely this approach.

Unfortunately, as we observed previously, Σ_{init} may not be sufficiently complete to determine a situation S such that $OBS_F(S)$ is entailed by our theory. In our earlier discussion, we proposed assumption-based explanatory diagnoses and potential explanatory diagnoses as means of addressing the problem. The idea was to further complete our theory by making assumptions regarding the truth value of selected literals. We observed that these assumptions could be made prior to attempting to determine an explanatory diagnosis, or that they could be made on an as-needs basis during computation. In the former case, we simply conjoin the regression of the assumption to the initial database and use regression and theorem proving as we would for generating normal explanatory diagnoses. In the latter case, generating an assumption-based explanatory diagnosis is analogous to an abductive planning problem, and we can use computational machinery for abduction to realize the computation of assumption-based explanatory diagnoses. We single out the work by Eshghi (Eshghi 1988) on abductive planning with event calculus as an example of abductive planning. While his representation and objectives are different from ours, his discussion of abductive planning illustrates the possibilities for the generation of assumption-based explanatory diagnoses. In the context of explanatory diagnosis, abductive plan generation would attempt to prove $OBS_F(s)$, if an attempted proof failed because it dead-ended on a literal or literals that were assumable, then these would be abduced and the proof continued. We have not yet implemented our own abductive explanation generator for the situation calculus.

Related Work

As previously noted, this work has been influenced by formal characterizations of diagnosis for systems without an explicit representation of actions (e.g., (de Kleer, Mackworth, & Reiter 1992), (Reiter 1987), (Console & Torasso 1991)). It has also been influenced by Reiter's work on the frame problem and the problem of temporal projection

(Reiter 1992a). With the exception of previous work by the author (McIlraith 1997a), the research to date on diagnostic problem solving has not really addressed the problem of incorporating a representation of action into the representation of the behaviour of a system. As such, there is little related work that exploits a comprehensive representation of action. In (McIlraith 1997b), the author has also provided a mapping of the traditional notions of *what is wrong* diagnosis, i.e., consistency-based and abductive diagnosis, into this rich representation scheme. There is other diagnosis research that is loosely related to the subject of this paper, particularly research on temporal diagnosis and diagnosis of dynamic systems (e.g., (Brusoni *et al.* 1995), (Console *et al.* 1994), (Hamscher 1991), (Friedrich & Lackinger 1991), (DeCoste 1990), (Lackinger & Nejdl 1991), (Dressler 1994)). Brusoni et al. ((Brusoni *et al.* 1995), (Brusoni *et al.* 1995)) recently provided a characterization of temporal abductive diagnosis together with algorithms for computing these diagnoses under certain restrictions. This builds on earlier work by Console et al. on diagnosing time-varying misbehaviour (Console *et al.* 1994). Their work decouples atemporal and temporal diagnoses, using SD to represent the behaviour of the atemporal components and transition graphs to represent the temporal components. The later work uses temporal constraints to represent the temporal components. Also related is the work by Cordier and Thiébaux on event-based diagnosis (Cordier & Thiebaux 1994). Their work is similar in motivation to our work on explanatory diagnosis, viewing the diagnosis task as the determination of the event-history of a system between successive observations. None of the work cited above provides a comprehensive representation of the preconditions for and the effects of actions. Nor does it address the frame, ramification and qualification problems.

In the area of reasoning about action, research on temporal explanation and postdiction is also very much related to results on explanatory diagnosis (e.g., (Crawford & Etherington 1992), (Baker 1991)). Of particular note is research by Shanahan on temporal explanation in the situation calculus (Shanahan 1993). While Shanahan also proposes the situation calculus as a representation language for axiomatizing his domain, he does so without an axiomatic solution to the frame and ramification problems. As such these problems must be addressed coincidentally with generating explanatory diagnoses. In contrast, our characterization of explanatory diagnosis, with its axiomatic solution to the frame and ramification problems, provides for much simpler characterization and computation of temporal explanation.

Contributions

The results in this paper provide contributions in two areas of research: model-based diagnosis and knowledge representation. We focus here on the contributions to the model-based diagnosis community. Our concern in this paper was, given a system that affects and can be affected by the actions of agents, and given some observed (aberrant) behaviour, how do we capture the notion of *what happened*, i.e., how do we go about conjecturing a sequence of actions that account for the behaviour we have observed.

We addressed this problem by providing a mathematical characterization of the notion of explanatory diagnosis in the context of a rich situation calculus representation, proposed in (McIlraith 1997a). Observing that we often have incomplete information about our initial state, we also proposed the notions of assumption-based and potential explanatory diagnosis, to allow for the conjectured sequences of actions that constitute a diagnosis to be predicated on some other assumptions we choose to make about the world. Our characterizations of explanatory diagnosis immediately make apparent the direct relationship of explanatory diagnosis to plan synthesis. As such, following the results of Green on deductive plan synthesis, we observed that generating explanatory diagnosis could be achieved by deduction. Of course, searching a situation space can be very inefficient, and further, our representation scheme for our situation space includes a second-order induction axiom. We showed that we can generate explanatory diagnoses more efficiently by regression followed by theorem proving in the initial database, which excludes the second-order induction axiom. Following regression, verifying a diagnosis involves simple query evaluation, whereas generating a diagnosis relies on deduction. We further observed that generating assumption-based explanatory diagnoses where the assumptions are conjectured on an as-needs basis requires abduction. Finally, while this paper's focus is on contributing to the mathematical foundations of diagnostic problem solving, we noted that there is a straightforward translation from our situation calculus representation scheme to Prolog, and further that Prolog's backwards chaining mechanism performs regression for us. The issue of computation has only been addressed in a cursory manner. In future research we will investigate the feasibility of various algorithms for computing explanatory diagnoses.

Acknowledgements

I would like to thank Ray Reiter, Yves Lespérance, Hector Levesque and Allan Jepson for helpful comments on the work presented in this paper. I would also like to thank the reviewers for their thoughtful reviews.

References

Baker, A. 1991. Nonmonotonic reasoning in the framework of the situation calculus. *Artificial Intelligence* 49:5–23.

Brusoni, V.; Console, L.; Terenziani, P.; and Dupré, D. T. 1995. Characterizing temporal abductive diagnosis. In *Proceedings of the Sixth International Workshop on Principles of Diagnosis*, 34–40.

Console, L., and Torasso, P. 1991. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence* 7(3):133–141.

Console, L.; Portinale, L.; Dupré, D. T.; and Torasso, P. 1994. Diagnosing time-varying misbehavior: an approach based on model decomposition. *Annals of Mathematics and Artificial Intelligence* 11(1–4):381–398.

Cordier, M., and Thiebaux, S. 1994. Event-based diagnosis for evolutive systems. In *Proceedings of the Fifth International Workshop on Principles of Diagnosis*, 64–69.

Crawford, J., and Etherington, D. 1992. Formalizing reasoning about change: A qualitative reasoning approach. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, 577–582.

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

de Kleer, J. 1991. Focusing on probable diagnoses. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 842–848.

DeCoste, D. 1990. Dynamic across-time measurement in interpretation. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, 373–379.

Dressler, O. 1994. Model-based diagnosis on board: Magellan-MT inside. In *Proceedings of the Fifth International Workshop on Principles of Diagnosis*, 87–92.

Eshghi, K. 1988. Abductive planning with event calculus. In *Proceedings of the Fifth International Logic Programming Conference*, 562–579.

Friedrich, G., and Lackinger, F. 1991. Diagnosing temporal misbehaviour. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, 1116–1122.

Green, C. C. 1969. Theorem proving by resolution as a basis for question-answering systems. In Meltzer, B., and Michie, D., eds., *Machine Intelligence 4*. New York: American Elsevier. 183–205.

Hamscher, W. 1991. Modeling digital circuits for troubleshooting. *Artificial Intelligence* 51(1–3):223–271.

Kramer, B., and et al., J. M. 1996. Developing an expert system technology for industrial process control: An experience report. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI'96)*, 172–186.

Lackinger, F., and Nejdl, W. 1991. Integrating model-based monitoring and diagnosis of complex dynamic systems. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, 1123–1128.

Lin, F., and Reiter, R. 1994. State constraints revisited. *Journal of Logic and Computation* 4(5):655–678. Special Issue on Action and Processes.

Lloyd, J. 1987. *Foundations of Logic Programming*. Springer Verlag, second edition.

McIlraith, S. 1994a. Further contributions to characterizing diagnosis. *Annals of Mathematics and Artificial Intelligence* 11(1–4):137–167.

McIlraith, S. 1994b. Towards a theory of diagnosis, testing and repair. In *Proceedings of The Fifth International Workshop on Principles of Diagnosis*, 185–192.

McIlraith, S. 1997a. Representing actions and state constraints in model-based diagnosis. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97).* To appear.

McIlraith, S. 1997b. *Towards a Formal Account of Diagnostic Problem Solving*. Ph.D. Dissertation, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.

Poole, D. 1988. Representing knowledge for logic-based diagnosis. In *Proceedings of the Fifth Generation Computer Systems Conference (FGCS-88)*, 1282–1290.

Reiter, R. 1987. A theory of diagnosis from first principles. *Artificial Intelligence* 32:57–95.

Reiter, R. 1991. *The Frame Problem in the Situation Calculus: A Simple Solution (sometimes) and a completeness result for goal regression*. Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of J. McCarthy. San Diego, CA: Academic Press. 359–380.

Reiter, R. 1992a. The frame problem in the situation calculus: A soundness and completeness result, with an application to database updates. In *Proceedings First International Conference on AI Planning Systems*.

Reiter, R. 1992b. The projection problem in the situation calculus: a soundness and completeness result, with an application to database updates. In *Proceedings First International Conference on AI Planning Systems*, 198–203.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In Aiello, L.; Doyle, J.; and Shapiro, S., eds., *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, 2–13. Cambridge, Massachusetts, USA.: Morgan Kaufmann.

Shanahan, M. 1993. Explanation in the situation calculus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, 160–165.

Waldinger, R. 1977. Achieving several goals simultaneously. In Elcock, E., and Michie, D., eds., *Machine Intelligence 8*. Edinburgh, Scotland: Ellis Horwood. 94–136.