Incorporating Action into Diagnostic Problem Solving (An Abridged Report)

Sheila A. McIlraith
Department of Computer Science, University of Toronto
Toronto, ON M5S 1A4 Canada
e-mail: mcilrait@cs.toronto.edu

Abstract

Reasoning about action and change is integral to the diagnosis, testing and repair of many artifacts, and yet there is no formal account of diagnostic problem solving which incorporates a theory of action and change. In this abridged report, we provide a situation calculus framework for diagnostic problem solving. Using this framework, we present results towards a characterization of diagnosis for behaviorally static systems whose state can be affected by events which occur in the world, and which require world-altering actions to achieve tests and repairs. Diagnosis is defined more broadly in terms of *what happened* in addition to the traditional conjecture of *what is wrong*. Observations of events and actions in the world are used to diagnose some system malfunctions. By developing our characterization in terms of the situation calculus, we are able to contribute towards a formal characterization and semantics for this broader notion of diagnosis, testing and repair, in addition to dealing formally with issues such as the frame problem.

Introduction

Diagnostic problem solving[1] *presents many practical challenges to our formal theories of action, both in terms of knowledge representation and in terms of reasoning. Not only must the frame problem be addressed, but the ramification and qualification problems occur when incorporating actions into an axiomatization of system behavior. Diagnostic problem-solving tasks, some of which are outline here, can be characterized formally as explanation, temporal projection and planning. Compilation procedures and simplifying assumptions facilitate characterization and computation of some of these tasks.*

Traditionally, the AI research on diagnosis has focused on the problem of determining a set of candidate diagnoses, given a description of system behavior and an observation of aberrant behavior (e.g., [1],[11]). Testing could subsequently be performed to acquire sufficient discriminatory observations in order to identify a unique diagnosis. In the broader context, a unique diagnosis isn't always an end in itself. The generation and discrimination of candidate diagnoses may only be performed to the extent that they enable an action to be selected. Recently, some researchers have cast diagnostic problem solving in a more purposive role (e.g., [2],[3],[10],[16]), making diagnosis a secondary side effect of reasoning to select a unique *repair*.

Much of the work to date on diagnosis, testing and repair has been devoted to reasoning about circuits or related electro-mechanical systems, where testing is nonintrusive and repair involves the simple replacement of component parts. These domains did not require explicit representation of actions for the tasks they were trying to address. In contrast, there are many applications for which testing and repair require a sequence of actions which actually change the state of the world in important ways [8]. For example, the achievement of a test, such as biopsying a tumor or testing the spark plugs in a car involve actions which change the state of the world. Similarly, repair procedures such as those required in medical treatment or machinery repair change the state of the world and affect subsequent observations and actions. Selecting appropriate actions for testing and repair is nontrivial and may be precluded by the state of the world or perceived state of the system. External actions and events which occur in the world are also relevant to diagnostic problem solving because they can affect the state and behavior of systems. (e.g., a power failure, an electrical shock, dropping a portable system on the floor, etc.). Reasoning about these events aids in the diagnosis of system malfunction, identifying the extent of malfunction, and in prescribing suitable repair. In these real-world domains, diagnostic problem solving *must* involve reasoning about action and change.

There have been some attempts at incorporating rudimentary actions into programs which reason to repair (e.g., [2],[3],[13],[16]). Friedrich et al. ([2],[3]) have provided a procedure to choose between performing simple observations and repair actions, assuming a most likely diagnosis. Sun and Weld [16] present a diagnostic reasoner which calls a decision-theoretic planner subroutine to plan repair actions. Their planning language distinguishes between information-gathering and state-altering actions. Both pieces of work are sound computational contributions to addressing the problem of reasoning to *repair*, in *restricted* domains. Neither

[1] The term *diagnostic problem solving* is used here to refer collectively to one or all of the tasks of diagnosis, testing and repair. More generally, the term also encompasses such tasks as monitoring, control and preventative maintenance.

system provides for the specification of diagnostic goals. Furthermore, neither addresses the fundamental knowledge representation challenges in reasoning about actions, such as how to integrate actions into the description of the behavior of a complex system, and how to address the frame, ramification and qualification problems, etc. Most importantly, neither provides formal characterizations of diagnosis, testing and repair in the context of a theory of action.

This paper proposes a situation calculus framework which casts diagnosis, testing and repair in the context of a theory of action and change. For the time being, we assume that the systems being addressed are static in the sense that their behavior only changes as the result of some event or action. We do not include explicitly time-varying or continuous systems in this characterization. However, with recent extensions to the expressiveness of the situation calculus, we believe that in the long-term this framework will also serve as the foundation for the diagnostic problem solving of these systems. Furthermore, this framework is sufficiently expressive to perform a variety of other computational tasks commonly addressed by researchers in the qualitative reasoning community.

Incorporating a theory of action into a diagnostic problem-solving framework will enable us to:

- characterize the effects of actions and events on the state and behavior of a system.
- **plan** a sequence of actions to achieve system testing, repair etc.; to understand the effects of those actions; and to ensure that they are both desirable and achievable in the current situation, given candidate diagnoses.
- expand our diagnostic reasoning beyond inferring *what is currently wrong*, to perform **explanation** (e.g., *what happened, what must have been wrong previously*, etc.); and **temporal projection** (e.g., *if event X occurs, what will be wrong with the system*, etc.).
- represent dynamic system behavior.
- distinguish explicitly between knowledge-producing actions and world-altering actions, and thereby to create a cognitive diagnostic agent.

In the following section, we describe our methodology, justifying our choice of the situation calculus. Further, we outline the challenges in incorporating action into a classical model-based diagnosis system description (SD), adopting a pragmatic means of addressing the frame, ramification and qualification problems. Following that, we recast consistency-based and abductive diagnosis in terms of the situation calculus to define a form of intra-situation diagnosis. We also provide two new definitions of inter-situation diagnosis. *Explanatory diagnosis* conjectures what events or actions must have occurred to result in the observed behavior. *Predictive diagnosis* takes observed events or actions and infers system malfunctions. In a longer version of this paper [7] we characterize the selection of actions to realize tests and repairs in the face of competing diagnoses. Further, we introduce the notion of a cognitive diagnostic agent.

The Situation Calculus

We employ the situation calculus as a logical specification language for characterizing diagnosis, testing and repair in the context of a theory of action and change. There are many advantages to such a formalization. We provide a non-procedural characterization of various diagnostic reasoning tasks for which meta-theoretic properties can be proven. The characterization enables us to formally address issues such as the frame, ramification and qualification problems; the complexity of various tasks; and to contribute towards a semantics. It also forces us to explicitly identify assumptions and enables us to assess the impact of assumptions and syntactic restrictions on the tasks. Clearly, characterization of these tasks in the situation calculus does not dictate that they must be realized in the situation calculus using a theorem prover. Ultimately, our characterization will provide the insight necessary to identify syntactic restrictions and notions of limited reasoning which will enable compact representations and efficient computation.

Several logics of action and change could have been employed in the development of this characterization (e.g., dynamic logic, event calculus). We chose the situation calculus because of the following desirable qualities: it is a logical language with Tarskian semantics; a solution to the frame problem exists; actions are treated as first-class objects in the situation calculus, enabling us to reason *about* actions and events within the language; finally, the expressive power of the situation calculus has been extended to incorporate time [9], knowledge-producing actions [14], and the agent programming language GOLOG [5], all of which can be employed in extensions to the work presented herein.

We assume a certain familiarity with the situation calculus. As a brief review, the situation calculus is essentially a sorted first-order language with sorts representing *actions*, *situations*, *fluents* and other *domain objects*. The state of the world is represented with respect to logical terms called *situations*, which can be thought of as snapshots of the world. *Actions* or events change the state of the world. A distinguished function do (i.e., $do(\alpha, s)$) denotes the result of performing action α in situation s. *Propositional fluents* are distinguished predicate symbols employing a situation term as their last argument. They represent relations whose truth values vary from situation to situation. For example $AB(c_1, s)$ represents the fact that component c_1 is abnormal in situation s. Similarly, *functional fluents* such as $temperature(patient, s)$ denote functions whose values vary from situation to situation. Fluents enable us to reason about changes in the truth value of diagnoses and observation values as the result of performing actions. For a more extensive discussion of the situation calculus, the reader is referred to ([5],[12]).

Axiomatizing a System

Incorporating actions into the axiomatization of the behavior of a system requires (1) specification of the behavior of the static system, (2) specification of the external events that can affect the system, and (3) specification of actions required to achieve system testing and repair. This task is not as straightforward as one would imagine. Even though we can incorporate Reiter's solution to the frame problem [12], the interaction between the static system description and the actions produces indirect effects and constraints on action

preconditions which present ramification and qualification problems. These must be addressed in order to preserve a solution to the frame problem. Proper axiomatization is critical to this venture. As a consequence we adopt a compilation and minimization procedure similar to that proposed in [6], to address the ramification, qualification and frame problems simultaneously. The result is a parsimonious description of system behavior in the presence of actions.

Since we are dealing with static systems, the representation of system behavior can be mapped from the first-order representations of SD traditionally used for diagnosis [1]. In the situation calculus framework, we characterize the behavior of our system relative to a situation. Any aspect of the system which can change as the result of an action or event is indexed by a situation, and represented as a fluent. These axioms, traditionally found in SD, are thus transformed into situation calculus *state constraints*. We will refer to them collectively as SD_s. Following the convention in the diagnosis literature [11], we distinguish the predicate AB to represent abnormal (and normal) behavior. AB literals and properties such as $emits\_light$ are represented as fluents, since their truth status can change.

Example We will illustrate the concepts in this paper in terms of a very simple example involving a flashlight. Our flashlight has three components with the potential to malfunction: the battery, the bulb, and the connection. When the flashlight is on, and all components are operating normally, a light is emitted. When the flashlight is off, no light is emitted. The flashlight can be opened, closed, turned on, and turned off. An abnormal battery component can be restored to normal functioning by replacing the battery. An abnormal connection between the battery and the bulb can be restored to normal functioning by closing the flashlight. Replacing the batteries closes the flashlight as a side effect. Dropping the flashlight or crushing the bulb will cause the bulb to operate abnormally. Soaking the flashlight in water will cause the battery to operate abnormally.

The static behavior of a flashlight with battery B, bulb D and connection C can be represented by SD_s:

$battery(t) \wedge connection(u) \wedge bulb(v) \wedge on(s) \wedge$
$\neg AB(t,s) \wedge \neg AB(u,s) \wedge \neg AB(v,s) \supset emits\_light(s)$
$\neg on(s) \supset \neg emits\_light(s)$
$battery(B) \wedge connection(C) \wedge bulb(D)$

Free variables in formulae are considered to be universally quantified from the outside, unless specified otherwise.

In addition to our axiomatization of the behavior of the static system SD_s, we must identify and characterize the behavior of all actions relevant to the testing and repair of our system, as well as actions or events which may change the state of our system. The axiomatizer may commence by providing the *necessary conditions for an action to be performed* and *effect axioms* for each fluent F. Effect axioms capture the "causal laws" of our domain. They describe the changes in the values of fluents as a result of performing actions. For example, a typical positive effect axiom is:

$Poss(a,s) \wedge bulb(v) \wedge [a = drop \vee a = crush(v)]$
$\supset AB(v, do(a,s))$.

This axiom states that the bulb will be abnormal in the situation resulting from either dropping the flashlight or crushing the bulb. The distinguished fluent $Poss(a,s)$ is used to represent that it is possible to perform action a in situation s.

All such effect axioms, necessary conditions for actions, and state constraints SD_s, must then collectively be transformed into:

- successor state axioms for each fluent F, and
- action precondition axioms for each action a.

Successor State Axioms To address the frame problem, an axiomatizer would normally have to provide numerous *frame axioms* specifying which fluents remain unchanged after actions are performed. Successor state axioms provide a parsimonious representation of frame and effect axioms, under the *completeness assumption* [12]. By assuming that the positive and negative effect axioms encode *all* conditions underwhich realizing an action results in a fluent F becoming true (false, respectively) in the successor situation, a parsimonious representation of frame axiom information as well as effect axiom information can be encoded into successor state axioms. There will be one successor state axiom per fluent.

Unfortunately, the description of system behavior SD_s yields state constraints which contribute indirect effects of actions. Consequently, generation of successor state axioms from effect axioms and state constraints suffers from the *ramification problem*. The indirect effects must be compiled into the successor state axioms using a minimization policy such as the one described in detail in [6]. The following example illustrates typical successor state axioms.

Example (continued)
Successor state axioms, SD_{SSA} are as follows:

$Poss(a,s) \supset [on(do(a,s)) \equiv$
$a = turn\_on \vee (on(s) \wedge a \neq turn\_off)]$
$Poss(a,s) \supset [closed(do(a,s)) \equiv$
$a = close\_up \vee \exists(b,n).a = replace\_bat(b,n) \vee$
$(closed(s) \wedge a \neq open\_up)]$
$Poss(a,s) \supset [AB(x, do(a,s)) \equiv$
$[battery(x) \wedge (a = soak \vee$
$[AB(x,s) \wedge \neg\exists(n).a = replace\_bat(x,n)])] \vee$
$[connection(x) \wedge (a = open\_up \vee$
$[AB(x,s) \wedge a \neq close\_up \wedge$
$\neg\exists(b,n).a = replace\_bat(b,n)])] \vee$
$[bulb(x) \wedge (a = crush(x) \vee a = drop]]$
$Poss(a,s) \supset [emits\_light(do(a,s)) \equiv$
$[(\exists t,u,v).(battery(t) \wedge connection(u) \wedge bulb(v)) \wedge$
$(\neg AB(t,s) \wedge \neg AB(u,s) \wedge \neg AB(v,s) \wedge$
$(a = turn\_on \vee$
$[on(s) \wedge a \neq turn\_off \wedge a \neq drop \wedge$
$a \neq soak \wedge a \neq open\_up \wedge a \neq crush(v)])) \vee$
$(AB(t,s) \wedge \neg AB(u,s) \wedge \neg AB(v,s) \wedge$
$on(s) \wedge (\exists n).(a = replace\_bat(t,n))) \vee$
$(\neg AB(t,s) \wedge AB(u,s) \wedge \neg AB(v,s) \wedge on(s) \wedge$
$[a = close\_up \vee (\exists n).(a = replace\_bat(t,n))]) \vee$
$(AB(t,s) \wedge AB(u,s) \wedge \neg AB(v,s) \wedge$
$on(s) \wedge (\exists n).(a = replace\_bat(t,n)))]]$

The first axioms states that given that the action a is possible in situation s, the flashlight will be on in the situation resulting from performing a in s iff a turned on the flashlight or it was already on and a did not turn it off.

Action Precondition Axioms Action precondition axioms specify the preconditions for each action to occur. They are collectively referred to as SD_{APA}. As with the successor state axioms, generating action precondition axioms in the presence of state constraints can be problematic. In particular, the state constraints may contribute implicit axioms about action preconditions, resulting in the *qualification problem.* We appeal to [6] for a minimization policy to create action preconditions from the successor state axioms, relevant state constraints and the stated necessary conditions for actions to be performed. Our example does not illustrate the qualification problem.

Example (continued)
The resultant SD_{APA} for our flashlight is:
$Poss(turn\_on, s) \equiv closed(s)$
$Poss(replace\_bat(b, n), s) \equiv battery(b) \wedge new\_bat(n)$
$Poss(crush(x), s) \equiv bulb(x)$
$Poss(turn\_off, s) \wedge Poss(open\_up, s) \wedge$
$Poss(close\_up, s) \wedge Poss(drop, s) \wedge Poss(soak, s)$

Although this paper does not detail the transformation and compilation policy for generating SD_{SSA} and SD_{APA}, utilization of such a procedure is critical to the incorporation of actions into traditional SD's [1].

Diagnosis

Given some observed aberrant behavior, diagnosis traditionally conjectures *what is wrong* with the system. (e.g., which components are behaving abnormally, what diseases a patient is suffering from etc.) By incorporating actions into our diagnostic framework, we can define a richer notion of diagnosis, distinguishing between *intra-situation* diagnosis and *inter-situation* diagnosis. Intra-situation diagnosis corresponds to traditional characterizations of diagnosis with respect to a static situation or "snapshot" of the world. In contrast, inter-situation diagnoses reason over more than one situation, to determine diagnoses. To this end, we provide a definition of *explanatory diagnosis*, which conjectures what *events* or actions occurred to result in our observation. (e.g., did the patient suffer an electrical shock, etc.) Further, by observing events in the world, we define a notion of *predictive diagnosis* which entails what must be wrong with a system, given the observed event. Extending the framework found in [1],

Definition 0.1 (System) *A system is a triple* $(SD, COMPS, OBS)$ *where:*
• SD, *the system description is a set of first-order situation calculus sentences consisting of:*
- SD_{S_0}, *the description of the static system behavior, relativized to the initial situation* S_0. *Any other problem-specific information about the initial state.*
- SD_{SSA}, *successor state axioms for actions.*
- SD_{APA}, *action precondition axioms for actions.*
• $COMPS$, *the system components is a finite set of constants.*
• OBS, *the observations, can be a first-order sentence or a sequence of actions.*

Intra-Situation Diagnosis

Traditional definitions of consistency-based diagnoses, abductive explanations, etc. (e.g., [1],[11]) map naturally into our framework. They enable us to conjecture *what is wrong* in a situation or static snapshot of the world, given an observation. The distinguishing feature of our situation calculus definitions is the indexing of diagnoses with respect to situations. Thus, the truth status of diagnoses and observations is defined relative to a situation. The persistence of diagnoses and observations across situations is captured by the solution to the frame problem integrated into our successor state axioms.

Definition 0.2 (AB-hypothesis) *Given two mutually exclusive sets of components* $\triangle_1, \triangle_2 \subseteq COMPS$, *define an AB-hypothesis* $\mathcal{D}_s(\triangle_1, \triangle_2)$ *of the system* $(SD, COMPS, OBS)$ *to be the conjunction:*
$[\bigwedge_{c \in \triangle_1} AB(c, s)] \wedge [\bigwedge_{c \in \triangle_2} \neg AB(c, s)]$,
with free variable s.

Definition 0.3 (Consistency-based Diagnosis) *Assume a system* $(SD, COMPS, OBS)$, *where* OBS *is a conjunction of ground fluents relativized to* S_0. *A consistency-based diagnosis for* $(SD, COMPS, OBS)$ *is* $\mathcal{D}_{S_0}(\triangle_1, \triangle_2)$, *an AB-hypothesis relativized to* S_0, *such that:* $\triangle_1 \cup \triangle_2 = COMPS$; *and* $SD \wedge OBS \wedge \mathcal{D}_{S_0}(\triangle_1, \triangle_2)$ *is satisfiable.*

Example (continued) Let $SD = SD_{S_0} \wedge SD_{APA} \wedge SD_{SSA}$ conjoined with the initial condition $on(L, S_0)$. $COMPS = \{B, C, D\}$ and let $OBS = \neg emit\_light(L, S_0)$. The space of consistency-based diagnoses in S_0 is as follows: $\mathcal{D}_{S_0}(\{B\}, \{C, D\})$, $\mathcal{D}_{S_0}(\{C\}, \{B, D\})$, $\mathcal{D}_{S_0}(\{D\}, \{B, C\})$, $\mathcal{D}_{S_0}(\{B, C\}, \{D\})$, $\mathcal{D}_{S_0}(\{C, D\}, \{B\})$, $\mathcal{D}_{S_0}(\{B, D\}, \{C\})$, $\mathcal{D}_{S_0}(\{B, C, D\}, \{\})$.

Definition 0.4 (Abductive Explanation) *Assume a system* $(SD, COMPS, OBS)$, *where* OBS *is a conjunction of ground fluents relativized to* S_0. *An abductive explanation for* $(SD, COMPS, OBS)$ *is* $\mathcal{D}_{S_0}(\triangle_1, \triangle_2)$, *an AB-hypothesis relativized to* S_0 *such that:* $SD \wedge \mathcal{D}_{S_0}(\triangle_1, \triangle_2) \models OBS$; $SD \wedge \mathcal{D}_{S_0}(\triangle_1, \triangle_2)$ *is satisfiable; and* $SD \not\models OBS$.

Following [1], we can adopt comparable notions of minimal and kernel diagnoses and explanations for the above definitions.

The computation of consistency-based diagnoses and abductive explanations in our framework can be accomplished by the traditional computational machinery, applied to SD_{S_0}. (e.g., ATMS, GDE, prime implicate generator, hitting set algorithm, etc. (e.g.,[1],[11]).) The proof is straightforward. Incremental diagnosis across situations exploits the successor state axioms, which capture the persistence of diagnoses.

Inter-Situation Diagnosis

Explanatory Diagnosis In addition to the traditional characterizations of diagnosis outlined above, we propose the

notion of an *explanatory diagnosis*, which conjectures what actions or events could have occurred in order to result in OBS. Knowing/conjecturing what happened is interesting in its own right, but also may assist in the prediction of other aberrant behavior or abnormal components and the prescription of suitable repair procedures. For example, if the television is not functioning, but it is conjectured that the television was dropped on the floor, then it is likely that many components of the television may be broken and the repair procedure will be affected. In the broader view of diagnostic problem solving, explanatory diagnoses may assist in preventative maintenance, artifact redesign, or control system alteration.

The problem of generating explanatory diagnoses is related to the problem of temporal explanation or postdiction (e.g., [15]). The task is as follows: from a description of system behavior and a history of actual system observations, conjecture a sequence of actions which account for the new observations. In a diagnostic setting, it is unlikely that we will have a history, unless our system is being continuously monitored. Consequently, we assume that our history is composed of the assumption that all components were behaving normally in the initial state. Alternatively, our history could be composed of the assumption that no aberrant behavior was being exhibited by the system in the initial state. Note, in this paper we denote $do(\alpha_n, do(\alpha_{n-1}, do(\ldots do(\alpha_0, s)))))$ by the abbreviation $do([\alpha_0, \ldots, \alpha_n], s)$.

Definition 0.5 (Explanatory Diagnosis) *Assume a system $(SD, COMPS, OBS)$, where OBS is a conjunction of fluents, relativized to free variable s. Let $H = \bigwedge_{c \in COMPS} \neg AB(c, S_0)$ be the history of the system. An explanatory diagnosis for OBS is a sequence of action $\alpha_0, \alpha_1, \ldots \alpha_n$ such that:*

1. $SD \wedge H \models O(do([\alpha_0, \ldots, \alpha_n], S_0))$

2. $SD \wedge H \models Poss(\alpha_0, S_0) \wedge Poss(\alpha_1, do(\alpha_0, S_0)) \wedge \ldots \wedge Poss(\alpha_n, do([\alpha_0, \ldots, \alpha_n], S_0))$.

The first condition states that OBS is true in the situation resulting from performing the sequence of actions $\alpha_0, \alpha_1, \ldots \alpha_n$, commencing in the initial state, S_0. The second condition ensures that the necessary preconditions are satisfied for each of the proposed actions. There may be many sequences of actions which meet these criteria. One or more best explanatory diagnoses may be selected by adopting different preference criteria. Following a syntactic definition of minimality, we could favor those diagnoses which contain the fewest primitive actions. Alternately, we may define a preference for "natural" events, or other distinguished actions.

Definition 0.6 (Minimal Explanatory Diagnosis) *An explanatory diagnosis D_E for $(SD, COMPS, OBS)$, given history H, is a minimal explanatory diagnosis iff there is no explanatory diagnosis D_E' such that the actions composing the sequence D_E' are a subset of the actions composing the sequence D_E.*

Explanatory diagnosis can be defined more generally in terms of a suitable circumscriptive policy, which minimizes the occurrence of unnecessary actions. Alternately, it can be defined in terms of abduction and minimization. The definition provided above allows for easy illustration of the correspondence between generating explanatory diagnoses and plan synthesis.

Identifying the sequence of actions composing an explanatory diagnosis is clearly a plan synthesis problem, realizable using theorem proving. According to [4], a plan to achieve a goal $G(s)$ is obtained as a side effect of proving $(\exists s)G(s)$. The bindings for the situation variable s represent the sequence of actions. In order to adhere to condition 2, we can appeal to work on goal regression such as [12].

An interesting special case of explanatory diagnosis occurs when we assume that only one action or event has occurred. The potential actions can then be "read off" from the successor state axioms. The computational advantage is a side effect of our compilation of the axioms. It is illustrated in the following example.

Example (continued) SD and $COMPS$ remain as defined in the previous example. Let the history $H = \neg AB(B, S_0) \wedge \neg AB(C, S_0) \wedge \neg AB(D, S_0)$, and let $OBS = \neg emits\_light(s)$ There are five different minimal explanatory diagnoses, each involving a single action. These diagnoses were simply read off the successor state axiom in SD_{SSA}: $turn\_off(S_0)$, $drop(S_0)$, $soak(S_0)$, $open\_up(S_0)$, $crush(B, S_0)$. Clearly there is an infinite number of nonminimal explanatory diagnoses. Note that from these different conjectured events we can conditionally entail the truth value of other fluents, and thus use testing to discriminate these competing explanatory diagnoses. Alternately, we could choose an action to perform.

Predictive Diagnosis In this subsection, we define the notion of *predictive diagnosis*, another type of inter-situation diagnosis which employs simple temporal projection. Given the observation of one or more events or actions in sequence, and a history of normal/abnormal components, a predictive diagnosis entails which components must be behaving normally/abnormally. Unlike other definitions of diagnosis found in this paper, predictive diagnoses are *not* defeasible.

Definition 0.7 (Predictive Diagnosis) *Assume a system $(SD, COMPS, OBS)$, where OBS is a sequence of actions $\alpha_0, \ldots, \alpha_n$, the observation that events $\alpha_0, \alpha_1, \ldots \alpha_n$ occurred in sequence from our initial situation S_0. Let $H = \bigwedge_{c \in COMPS} \neg AB(c, S_0)$ be the history of the system. A predictive diagnosis $(SD, COMPS, OBS)$ is $\mathcal{D}_s(\Delta_1, \Delta_2)$, an AB-hypothesis relativized to s, such that $\Delta_1 \cup \Delta_2 = COMPS$, and $SD \wedge H \models \mathcal{D}_s(\Delta_1, \Delta_2)$, where $s = do([\alpha_0, \ldots, \alpha_n], S_0)$.*

As with the definition of explanatory diagnosis, this definition of predictive diagnosis has been presented with some simplifying assumptions to make it more computationally feasible. We assume in the formulation of our definition that no events or actions occur, other than those that we observe. This is achieved by representing the observed events/actions as the sequence $do(\alpha_n, do(\alpha_{n-1}, do(\ldots do(\alpha_0, S_0))))$. Further, we assume an explicit history. Making these simplifying assumptions avoids the necessity of defining a circumscriptive-style minimization policy with respect to the

occurrence of unobserved events or actions. Similarly, the presence of a history enables us to avoid having to minimize the occurrence of AB fluents by assuming the persistence of $\neg AB$ from S_0.

Example (continued) SD and $COMPS$ remain as defined in the previous example. Let the history $H = \neg AB(B, S_0) \wedge \neg AB(C, S_0) \wedge \neg AB(D, S_0)$, and let $OBS = crush(B, S_0), soak((do(crush(B, S_0))))$. (i.e., we observed that the bulb was crushed and then the flashlight was soaked in water.) The predictive diagnosis is $\mathcal{D}_s(\{B, D\}, \{C\})$. We predict that the bulb and battery are operating abnormally, and that the connection is OK.

Summary

This paper presents a situation calculus knowledge representation framework for incorporating actions into diagnostic problem solving. Further, it presents a broad characterization of diagnosis, including definitions for explanatory diagnosis and predictive diagnosis as well as mapping traditional notions of consistency-based diagnosis and abductive explanation into the situation calculus framework.

Diagnostic problem solving clearly provides a rich domain for application of formal theories of action.

- The axiomatization of diagnostic problem solving domains presents some challenging knowledge representation problems. The behavior of a static system is represented as state constraints, resulting in ramification and qualification problems, in addition to the frame problem.
- Characterization of different diagnostic tasks appeals to explanation, temporal projection, and planning, in addition to consistency and entailment.
- Compilation procedures and simplifying assumptions facilitate characterization and computation of some of these tasks.

Not discussed in this abridged paper:

- The achievement of tests and repairs requires reasoning to achieve a state of the world in the face of competing diagnoses. This can be characterized as a planning problem.
- To integrate diagnosis, testing, and repair within the language we need to distinguish between observed behavior and expected behavior. We can employ Scherl and Levesque's extension to the situation situation [14] to incorporate knowledge and knowledge-producing actions. In this context, diagnosis can be viewed as planning to achieve a state of knowledge, and repair as planning to achieve some state of the world. A test changes the state of knowledge, while the realization of tests can also change the state of the world.

Acknowledgements

We would like to acknowledge the Cognitive Robotics Group, University of Toronto, whose research on the situation calculus provided a foundation for this work. Thanks to Mike Grüninger, Yves Lespérance, Ray Reiter, Dale Schuurmans and Steven Shapiro for discussion and comments.

References

[1] J. de Kleer, A. Mackworth and R. Reiter (1992). Characterizing diagnoses and systems. In *Artificial Intelligence Journal* **56**:197–222.

[2] G. Friedrich, G. Gottlob and W. Nejdl (1992). Formalizing the repair process. In *European Conference on Artificial Intelligence.*

[3] G. Friedrich and W. Nejdl (1992). Choosing observations and actions in model-based diagnosis/repair systems. In B. Nebel et al. (ed.), *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 489–498. Morgan Kaufmann.

[4] C. C. Green (1969). Theorem proving by resolution as a basis for question-answering systems. In B. Meltzer and D. Michie (ed.), *Machine Intelligence 4*,183–205. Elsevier.

[5] Y. Lespérance, H. Levesque, F. Lin, D. Marcu, R. Reiter, and R. Scherl (1994). A logical approach to high-level robot programming – A progress report. In B. Kuipers (ed.), *Control of the Physical World by Intelligent Systems, Papers from the 1994 AAAI Fall Symposium*, 79–85.

[6] F. Lin and R. Reiter (1994b). State constraints revisited. In *Journal of Logic and Computation, Special Issue on Action and Processes.* To appear.

[7] S. McIlraith (1995). Incorporating action into diagnostic problem solving. submitted to *The Fourteenth International Joint Conference on Artificial Intelligence.*

[8] S. McIlraith and R. Reiter (1992). On tests for hypothetical reasoning. In W. Hamscher et al. (ed.), *Readings in model-based diagnosis*, 89–96. Morgan Kaufmann.

[9] J. Pinto (1993). Temporal Reasoning in the Situation Calculus. Ph.D. Thesis, Dept. of Computer Science, Univ. of Toronto.

[10] G. Provan and D. Poole (1991). The utility of consistency-based diagnostic techniques. In J. Allen et al. (ed.), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 461–472. Morgan Kaufmann.

[11] R. Reiter (1987). A theory of diagnosis from first principles. *Artificial Intelligence Journal* **32**:57–95.

[12] R. Reiter (1991). The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression, In V. Lifschitz (ed.), *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, 359–380. Academic Press.

[13] R. Rymon (1993). Diagnostic reasoning and planning in exploratory-corrective domains. PhD thesis. Department of Computer Science, University of Pennsylvania.

[14] R. Scherl and H. Levesque (1993). The frame problem and knowledge-producing actions, In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 689–695.

[15] M. Shanahan (1993). Explanation in the situation calculus, In *Proc. of the Thirteenth International Joint Conference on Artificial Intelligence*, 160–165.

[16] Y. Sun and D. Weld (1993). A framework for model-based repair. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 182–187.