Towards a Theory of Diagnosis, Testing and Repair

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

Abstract

In this paper, we provide a situation calculus framework for diagnostic problem-solving in the context of a theory of action and change. Using this framework, we present results towards a characterization of diagnosis, testing and repair for behaviorally static systems 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*. We contrast between two related roles played by our situation calculus framework: modeling the behavior of a system to enable reasoning about diagnosis, testing and repair; and using the situation calculus as a representation language for a cognitive diagnostic agent to represent its model of the behavior of a system, its knowledge of the world, and the perceptual actions it can perform. By formulating these notions 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. This paper provides an introduction to the situation calculus framework we employ and an overview of our characterization.

1 INTRODUCTION

My flashlight is not working. It does not emit light when I turn it on. I conjecture that the batteries may be dead, the bulb may be malfunctioning, or there may simply be a loose connection somewhere between the batteries and the bulb. My first action is to open the body of the flashlight, remove the old batteries, and replace them with new batteries. Now when I turn on my flashlight, a light is emitted. I still haven't diagnosed what was wrong with my flashlight. It might have been the batteries, but it could also have been a loose connection which I fixed when replacing the batteries. Either way, my flashlight is now emitting a light, which was my ultimate goal.

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., (de Kleer and Williams, 1987), (Reiter, 1987)). Testing could subsequently be performed to acquire sufficient discriminatory observations in order to identify a unique diagnosis. Recently, some researchers have cast diagnostic problem-solving in a more purposive role (e.g., (Provan and Poole, 1991), (Friedrich and Nejdl, 1992), (Friedrich et al., 1992), (Sun and Weld, 1993)), making diagnosis a secondary side-effect of reasoning to *repair* a system. In this setting, diagnosis and testing are only performed to the extent that they enable the problem-solver to select an appropriate 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 actions involve the simple replacement of component parts. There are many applications for which testing and repair require a series of actions which actually change the state of the world in important ways (McIlraith and Reiter, 1992). For example, the achievement of a test, such as biopsying a tumor or testing the spark plugs in a car require 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.

In order to perform diagnostic problem-solving[1] in these more complex domains, we must represent the achievement of tests and repairs as actions and reason about how these actions affect the state of the world. Selecting appropriate actions for testing and repair is nontrivial. A suspected diagnosis or state of the world

[1]The term *diagnostic problem-solving* is used here to refer collectively to one or all of the tasks of diagnosis, testing and repair

may preclude the execution of a particular test or repair action. For example, if we knew that a patient was pregnant, we might not use radiological testing. Furthermore, the achievement of tests changes the state of the world, and in so doing, may change our space of diagnoses. This was the case in the flashlight example.

While there has been some procedural work (as cited above) on selecting repairs, there has been little work which deals with the difficult knowledge representation issues associated with reasoning about action and change, and there is no formal specification of a framework or theory of diagnosis, testing and repair in such a context. (The most notable contribution towards this goal is (Rymon, 1993) which details an architecture and system for diagnosis and repair which was applied to medical multiple trauma management.)

This paper proposes a situation calculus framework which casts diagnosis, testing and repair in the context of a theory of action and change. By formulating these notions in terms of the situation calculus we are able to contribute towards a formal characterization and semantics for this broader notion of diagnostic problem-solving, in addition to dealing formally with issues such as the frame problem[2]. For the time being, we assume that the systems to be diagnosed, tested and repaired 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 dynamic systems.

The purpose of this paper is to introduce our situation calculus framework to the diagnosis community and to present contributions towards a characterization of diagnosis, testing and repair in the context of a theory of action and change. As such, we gloss over some details in favor of the broad perspective. The paper serves as a critical foundation to and overview of the definitions and issues related to this area of diagnosis research. In Section 2, we review the situation calculus and discuss how to represent system behavior. In Section 3 we recast consistency-based and abductive diagnosis in terms of the situation calculus as well as providing a new definition of *explanatory diagnosis* which conjectures *what happened* to cause the observed aberrant behavior. Section 4 outlines the definition and achievement of tests in our framework, while Section 5 deals with the issue of repair. In Section 6, we discuss the integration of diagnosis, testing and repair. To this end, we introduce the notion of a cognitive diagnostic agent, and discuss how the requirements of such an agent result in a modification to our representation and reasoning strategies. We summarize the contributions of this paper in Section 7.

[2]The frame problem is the problem of characterizing the aspects of a situation that are unchanged as the result of performing an action.

2 THE SITUATION CALCULUS

We employ the situation calculus as a logical specification language for characterizing diagnosis, testing and repair. There are many advantages to such a formalization. We provide a non-procedural characterization of the tasks from which meta-theoretic properties may be proven. The characterization enables us to formally deal with issues such as the frame problem, 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. Finally, characterization of these tasks in the situation calculus does not determine that they must be realized in the situation calculus using a theorem prover. In particular, versions of STRIPS have been formally related to the situation calculus, providing a semantics for STRIPS and a ready algorithm for some of our work (Lin and Reiter, 1994). Any algorithm that can be proven correct with respect to the characterization may be reliably employed. In this section, we review the salient features of the situation calculus and demonstrate its use in axiomatizing the behavior of a static system which includes actions to achieve system testing and repair. For a more extensive discussion of the situation calculus, the reader is referred to (McCarthy and Hayes, 1969) and (Reiter, 1991).

The situation calculus was first proposed by McCarthy as a first-order language for reasoning about actions and their effects on the state of the world. Following the description in (Reiter, 1991) and (Lesperance et al., 1994), the situation calculus is 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 in essence are snapshots of the world. *Actions* or events change the state of the world. A distinguished function do (i.e., $do(\alpha, s)$) defines 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 which 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.

We propose two uses for the situation calculus:
1. As a language to represent the behavior of a static system and the actions required for testing and repair.

2. As a language to represent a cognitive diagnostic agent's *"brain"* — its model of the behavior of a system, (which should correspond functionally to 1., above), perceptual actions it can perform and its knowledge of the world.

For the time being, we concentrate on the use of the situation calculus for 1, returning to the notion of a cognitive diagnostic agent in Section 6.

2.1 AXIOMATIZING A SYSTEM

Axiomatization of a system requires specification of the behavior of the static system, and specification of the actions required to achieve system testing and repair. 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 (de Kleer et al., 1992). 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. Note, in this paper, actions are restricted to primitive actions[3]. Free variables in formulae are considered to be universally quantified from the outside.

Example 1.
For example, the behavior of an inverter A can be specified as follows:
$INV(c) \wedge \neg AB(c,s) \supset in(c,s) = out(c,s)$
$INV(A)$

Example 2.
The behavior of flashlight L with battery B, bulb D and connection C between the bulb and the battery can be represented by SD_s:
$flashlight(x) \wedge battery(t) \wedge connection(u) \wedge bulb(v) \wedge$
$on(x,s) \wedge \neg AB(t,s) \wedge \neg AB(u,s) \wedge \neg AB(v,s) \supset$
$emits\_light(x,s)$
$flashlight(x) \wedge \neg on(x,s) \supset \neg emits\_light(x,s)$
$flashlight(L) \wedge battery(B) \wedge connection(C) \wedge bulb(D)$

Following the convention in the diagnosis literature (Reiter, 1987), we distinguish the predicate AB to represent abnormal (and normal) behavior. Note that the AB literals and properties such as $emits\_light$ are represented as fluents, since their truth status can change.

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. The axiomatizer may commence by providing the *necessary conditions for an action to be performed* and *effect axioms* for each fluent F. These axioms and the 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 provide a succinct encoding of effects under the assumption of a complete axiomatization, providing a solution to the frame problem. Action precondition axioms define necessary and sufficient conditions for an action to be executable in a situation. SD_s imposes constraints on the effects of and preconditions for actions, resulting in ramification and qualification problems. Fortunately, these are addressed by our framework.

Example 2. (continued)
In the case of the flashlight example, we will need to specify the actions $turn\_on(x)$ and $turn\_off(x)$ to turn on (off, respectively) flashlight x; $open\_up(x)$ and $close\_up(x)$ to open (close, respectively) the body of flashlight x; and $replace\_bat(t,b,x)$ to put a new battery b in the battery component t of flashlight x.

We introduce the notions of effect axioms, successor state and action precondition axioms, which are illustrated in the context of our flashlight example.

2.1.1 Effect Axioms

Effect axioms capture the "causal laws" of our domain. For each fluent F, there are both positive and negative effect axioms. They represent the changes in the values of fluents as a result of performing actions. Effect axioms are conditioned on the fact that it is possible to execute the action in the situation in question. To this end, a distinguished fluent $Poss(a,s)$ is used to represent that it is possible (not possible, respectively) to do action a in situation s.

General Positive Effect Axioms for fluent F are of the form: $Poss(a,s) \wedge \gamma_F^+(\vec{x},a,s) \supset F(\vec{x},do(a,s))$, where $\gamma_F^+(\vec{x},a,s)$ is a formula describing the conditions underwhich doing an action a in situation s will cause the fluent F to become true. For example, the formula:
$Poss(a,s) \wedge [(a = drop(x) \wedge fragile(x)) \vee$
$a = crushed(x)] \supset broken(x, do(a,s))$
says that if the preconditions for the action $drop(x)$ are met and x is fragile, then doing the action $drop(x)$ in situation s will result in x being broken in the resulting situation. Similarly, for action $crushed(x)$.

General Negative Effect Axioms for fluent F are of the form: $Poss(a,s) \wedge \gamma_F^-(\vec{x},a,s) \supset \neg F(\vec{x},do(a,s))$, where $\gamma_F^-(\vec{x},a,s)$ is a formula describing the conditions underwhich doing an action a in situation s will cause the fluent F to become false. For example,
$Poss(a,s) \wedge a = repair(x) \wedge broken(x,s) \wedge y = x \supset$
$\neg broken(y, do(a,s))$
says that if the preconditions for $repair(x)$ are met and x is broken, then doing $repair(x)$ in situation s will result in x not being broken in the resulting situation.

[3] Primitive actions are roughly those actions which are not defined in terms of other actions.

2.1.2 Successor State Axioms

To address the frame problem, an axiomatizer would normally have to provide *frame axioms* specifying which fluents remain unchanged after actions. This would result in 2 x $\mathcal{A}$ x $\mathcal{F}$ frame axioms, where $\mathcal{A}$ and $\mathcal{F}$ represent the number of actions and fluents in our language. Successor state axioms provide a parsimonious representation of frame and effect axioms, under the *completeness assumption* (Reiter, 1991). 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, resulting in only $\mathcal{F}$ axioms. Successor state axioms are of the following form: $Poss(a,s) \supset [F(\vec{x}, do(a,s)) \equiv \gamma_F^+(\vec{x},a,s) \vee (F(\vec{x},s) \wedge \neg\gamma_F^-(\vec{x},a,s))]$. Similar successor state axioms may be written for functional fluents. For the fluent $broken(x)$ above, the successor state axiom is as follows:

$Poss(a,s) \supset [broken(x, do(a,s)) \equiv$
$(a = drop(x) \wedge fragile(x)) \vee a = crushed(x) \vee$
$(broken(x,s) \wedge a \neq repair(x))]$

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. This issue is discussed in (Lin and Reiter, 1994b). The results of such a transformation are illustrated below.

Example 2. (continued)
Successor state axioms, SD_{SSA} are as follows:

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

To keep the example simple, we are not encoding any actions which cause the battery or the connections to become abnormal. Furthermore, we have not encoded actions which change the state of normalcy of the bulb.

2.1.3 Action Precondition Axioms

Action precondition axioms specify the preconditions $\pi_\alpha(\vec{x},s)$ for action $\alpha(\vec{x})$ to occur. Action precondition axioms are of the form: $Poss(\alpha(\vec{x}),s) \equiv \pi_\alpha(\vec{x},s)$

Example 2. (continued)
For example, the action $turn\_on(x)$ requires that the flashlight x be closed. $turn\_off(x)$, $open\_up(x)$ and $close\_up(x)$ require that x be a flashlight, while the action $replace\_bat(t,b,x)$ requires that b be new and that t be the battery component of flashlight x which is open. The action precondition axioms SD_{APA} for our flashlight example are as follows:

$Poss(turn\_on(x),s) \equiv flashlight(x) \wedge closed(x,s)$
$Poss(turn\_off(x),s) \equiv flashlight(x)$
$Poss(open\_up(x),s) \equiv flashlight(x)$
$Poss(close\_up(x),s) \equiv flashlight(x)$
$Poss(replace\_bat(t,b,x),s) \equiv$
$battery(t) \wedge flashlight(x) \wedge open(x,s) \wedge new(b)$

Generating action precondition axioms in the presence of state constraints can be complicated. In particular, the state constraints may contribute implicit axioms about action preconditions, resulting in the *qualification problem*. (Lin and Reiter, 1994b) again describes a transformation policy to produce action preconditions using successor state axioms, relevant state constraints and the stated necessary conditions for actions to be performed. The transformation provides a solution to the qualification problem. Note: special care must be taken by the axiomatizer to ensure that if the execution of any actions is precluded by the existence of certain diagnoses, then they must be specified explicitly in the action preconditions.

3 DIAGNOSIS

Given some observed aberrant behavior, diagnosis traditionally tells us *what is wrong* with the system; e.g., which components are behaving abnormally, what diseases a patient is suffering from etc. Viewing diagnosis from the context of a theory of action and change, we can extend our definition of diagnosis to also encompass the question of *what happened* to cause certain aberrant behavior; e.g., did the patient suffer an electrical shock, was the television dropped etc.

In this section we provide definitions of consistency-based diagnosis and abductive explanation in the context of our situation calculus framework. Additionally we provide a definition of explanatory diagnosis, which accounts for "what happened" as opposed to "what is wrong". Following the definitions found in (de Kleer et al., 1992),

Definition 3.1 (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 employed in system testing and repair.*
- SD_{APA}*, action precondition axioms for actions employed in system testing and repair.*
• $COMPS$*, the system components is a finite set of constants.*
• OBS*, a set of observations is a set of first-order sentences.*

Example 2. (continued)
The axiomatization of the flashlight example in the previous section, combined with the initial condition $on(L, S_0)$, would constitute a typical SD. $COMPS = \{B, C, D\}$ and let $OBS = \neg emit\_light(L, S_0)$.

3.1 WHAT IS WRONG?

The definitions for consistency-based diagnoses, abductive explanations etc. (e.g., (de Kleer et al., 1992)) map very naturally into our framework. These definitions of diagnoses and explanations correspond to what is referred to here as *"what is wrong"* diagnoses.

The distinction between "what's wrong" diagnoses in the situation calculus and previous definitions, is the indexing of our diagnoses with respect to situations. The truth status of diagnoses and observations are 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. We provide new definitions of AB-literal and AB-hypothesis.

Definition 3.2 (AB-literal) *An AB-literal is* $AB(c, s)$ *or* $\neg AB(c, s)$ *for some* $c \in COMPS$.

Definition 3.3 (AB-hypothesis) *Given two mutually exclusive sets of components* $\triangle_1, \triangle_2 \subseteq COMPS$*, define an AB-hypothesis* $\mathcal{D}(\triangle_1, \triangle_2)$ *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.

Consistency-based diagnosis, abductive explanation and other related concepts, conform exactly to those found in the literature (e.g., (de Kleer et al., 1992)).

Definition 3.4 (Consistency-based Diagnosis) *A consistency-based diagnosis of* $(SD, COMPS, OBS)$ *is an AB-hypothesis* $\mathcal{D}(\triangle_1, \triangle_2)$ *such that* $\triangle_1 \cup \triangle_2 = COMPS$*, and* $SD \cup OBS \cup \{\mathcal{D}(\triangle_1, \triangle_2)\}$ *is satisfiable.*

Definition 3.5 (Abductive Explanation) *An abductive explanation for* $(SD, COMPS, OBS)$ *is any AB-hypothesis* $\mathcal{D}(\triangle_1, \triangle_2)$ *such that:* **1.** $SD \cup \{\mathcal{D}(\triangle_1, \triangle_2)\} \models OBS$*;* **2.** $SD \cup \{\mathcal{D}(\triangle_1, \triangle_2)\}$ *is satisfiable; and* **3.** $SD \not\models OBS$.

The computation of consistency-based diagnoses and abductive explanations in our framework may be accomplished by the computational machinery used for traditional definitions of the above, applied to SD_{S_0}. Incremental diagnosis exploits the successor state axioms, which capture the persistence of diagnoses.

3.2 WHAT HAPPENED?

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 the development of preventative maintenance procedures or artifact redesign. For example, if the fuses in a car have blown, then normally one would simply replace them, but if they blew because they got wet from a hole in the wheel well of the car, then the wetting of the fuses was the event that caused the fuses to blow. Preventative measures would dictate that the hole in the wheel well be repaired, as well as the fuses.

The problem of generating explanatory diagnoses is related to the problem of temporal explanation or postdiction (e.g., (Shanahan, 1993)). The task is as follows: from a description of system behavior and a history of actual system observations or actions, 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. Below we provide a formal definition of explanatory diagnosis.

Definition 3.6 (Explanatory Diagnosis) *Assume a system* $(SD, COMPS, OBS)$*, where* $OBS = (\exists s)O(s)$*, and* $O(s)$ *is any first-order formula with 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. $SD \cup H \models O(do(\alpha_n, do(\alpha_{n-1}, do(\ldots do(\alpha_0, S_0)))))$.
2. $SD \cup H \models Poss(\alpha_0, S_0) \wedge Poss(\alpha_1, do(\alpha_0, S_0)) \wedge$
$\ldots \wedge Poss(\alpha_n, do(\alpha_{n-1}, do(\ldots (do(\alpha_0, S_0)))))$.

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. We will favor those which are most direct and are irredundant. (More on this later.)

Identifying the sequence of actions is clearly a plan synthesis problem, which can be realized formally using theorem proving. According to (Green, 1969), 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 appeal to work on goal regression by (Reiter, 1991) who solves this problem by the introduction of a new predicate $ex(s)$ to represent that a plan is executable. Following Reiter: $ex(s) \equiv s = S_0 \wedge (\exists a, s^{'})s = do(a, s^{'}) \wedge Poss(a, s^{'}) \wedge ex(s^{'})$. This recursive formula states that s is an executable plan if it is composed of a sequence of actions whose preconditions are true in the previous state. Thus, generating an explanatory diagnosis can be formulated as the problem of establishing that $SD \cup H \models (\exists s)O(s) \wedge ex(s)$, where $OBS = (\exists s)O(s)$

An interesting special case of explanatory diagnosis is the case where we assume that only one action or event has occurred. The potential actions can then be generated easily from the successor state axioms.

4 TESTING

Given a theory of system behavior and a set of candidate diagnoses, we may wish to perform tests in order to discriminate these candidate diagnoses in some fashion. In previous papers ((McIlraith and Reiter, 1992), (McIlraith, 1994)), we have proposed a theory of testing for hypothetical reasoning which relates directly to the diagnosis literature. A test specifies some initial condition A which the tester establishes, and an observable O whose truth or instantiated value the tester is to determine from the physical world. The outcome of such a test ideally provides further discriminatory information which will allow for the refutation of certain candidate diagnoses.

As pointed out in these papers, the achievement of tests is assumed to be nonintrusive. Thus, the realization of a test has no effect on the state of the world. While a reasonable assumption for many application domains where tests might require the probing of circuits, or the reading of sensor values, there are domains where the achievement of tests requires the execution of actions which can change the state of the world.

In this section, we discuss the achievement of tests which *are* intrusive, and which can and do change the state of the world. We extend the research in (McIlraith and Reiter, 1992) and (McIlraith, 1994) to do so. We distinguish a subset of ground literals of our language, called the *achievables*. Achievables are generally fluents of the language, though they need not be. These will specify the initial conditions for a test – the conditions which must be true before we make an observation in the physical world. In addition, we define the *observables*, a distinguished set of fluents of our language whose truth value we wish to establish.

Definition 4.1 (Test) *A test is a pair (A, O) where A is a conjunction of achievable literals and O is an observable fluent.*

(McIlraith, 1994) distinguishes between two types of tests, **truth tests** and **value tests**. In the interest of brevity, we only provide a definition for the former.

Definition 4.2 (Truth Test) *Let the observable O be a ground fluent, indexed with respect to the current situation. A truth test is a test (A, O), whose outcome β is one of $O, \neg O$.*

Example 2. (continued)
$(on(F, s), emits\_light(F, s))$ is a truth test. It states that we must achieve the condition of the flashlight F being on and then we must observe whether or not it will emit a light.

In order to perform a test, we must ensure that the achievable conditions of the test are true, or plan a sequence of actions in order to achieve them. Using the abbreviation $D_i(S_0)$ for $D_i(\Delta_1, COMPS - \Delta_1)$ situated in the initial situation, we can define a plan to achieve a test (A, O).

Definition 4.3 *Let $DIAGS$ be the set of diagnoses, initially $\{D_1(S_0), D_2(S_0), \ldots D_k(S_0)\}$. A sequence of actions $\alpha_0, \alpha_1, \ldots \alpha_n$ constitutes a plan to achieve test (A, O) iff*
1. $SD \cup \bigvee_{D_i \in DIAGS} D_i(S_0) \models$
$A(\vec{x}, (do(\alpha_n, do(\alpha_{n-1}, do(\ldots do(\alpha_0, S_0))))))$.
2. $SD \cup \bigvee_{D_i \in DIAGS} D_i(S_0) \models$
$Poss(\alpha_0, S_0) \wedge Poss(\alpha_1, do(\alpha_0, S_0)) \wedge$
$\ldots \wedge Poss(\alpha_n, do(\alpha_{n-1}, do(\ldots (do(\alpha_0, S_0)))))$.

The first condition states that the achievable is true in the situation that results from the application of actions $\alpha_0, \alpha_1, \ldots \alpha_n$. The second condition states that each action α_i is possible in the situation to which it will be applied. None of the diagnoses precludes the execution of the actions. As with explanatory diagnoses, the sequence of actions may be generated as the side effect of theorem proving.

(McIlraith and Reiter, 1992), (McIlraith, 1994) also provide definitions for confirmation and refutation as well as distinguishing between discriminating, relevant

and constraining tests. These definitions translate readily into our situation calculus framework.

The selection of appropriate tests in this framework is complicated by the fact that the achievement of tests can change the state of the world and thus change the space of diagnoses, before actual testing occurs. Of related note, the completion assumption incorporated into the successor state axioms ensures that every observation must be accounted for. This causes both confirming and refuting tests (McIlraith and Reiter, 1992) to be discriminatory.

5 REPAIR

The long-term objective of diagnostic problem-solving is often repair of the system. We use the term repair loosely in this context to cover both the repair of abnormal components as well as simply the alleviation of aberrant behavior. Using our framework, we may plan a sequence of actions to carry out a nontrivial repair.

Definition 5.1 *Given a system* $(SD,COMPS,OBS)$ *and a known diagnosis* $D(S_0)$*, A sequence of actions* $\alpha_0, \alpha_1, \ldots \alpha_n$ *constitutes a plan to repair the abnormal components of* SD *iff:*

1. $SD \cup D(S_0) \models$
$\bigwedge_{c \in COMPS} \neg AB(c, (do(\alpha_n, do(\alpha_{n-1}, do(\ldots$
$\ldots do(\alpha_0, S_0))))))$.

2. $SD \cup D(S_0) \models$
$Poss(\alpha_0, S_0) \wedge Poss(\alpha_1, do(\alpha_0, S_0)) \wedge$
$\ldots \wedge Poss(\alpha_n, do(\alpha_{n-1}, do(\ldots(do(\alpha_0, S_0)))))$.

The first condition states that in the situation resulting from applying the repair plan, all components will be normal. The second condition assures that each of the actions in the plan is possible to execute in the relevant situation. Again, computation of this plan synthesis problem may be achieved through theorem proving with the use of the $ex(s)$ predicate.

As pointed out by previous researchers (e.g., (Provan and Poole, 1991)), we need not identify a unique diagnosis before repairing a system. Often there are equivalence classes of diagnoses which require the same repair. Additionally, we may not always have a unique diagnosis, before contemplating repair procedures. In such instances, we may choose to attempt repair based on what is believed to be the most probable diagnosis. We must then ensure that any of the actions we take would not be precluded, given that one of the other diagnoses were true instead. This requires a simple modification to the definition above – replacing $D(S_0)$ with $\bigvee_{D_i \in DIAGS} D_i(S_0)$, in criterion 2.

6 INTEGRATION

In the previous sections we have demonstrated the use of the situation calculus as a language to represent the behavior of static systems and the actions required for their testing and repair. While each individual task (diagnosis, testing, repair) is itself performed quite easily in this paradigm, automatic integration and interplay between diagnosis, testing and repair is not attainable *within the language as presented.* A meta-reasoner, a user or system controller of some sort is required to integrate the sequencing of diagnosis, testing and repair activities. Furthermore, in the presented version of the situation calculus, we cannot plan to achieve *information-related* objectives, such as reasoning in order to know that the bulb in not malfunctioning. We have no way of specifying these goals within our language. While this is the way most diagnostic problem-solving is performed currently, we *can* attain integration, within the situation calculus.

The situation calculus may be used as a language to represent a cognitive diagnostic agent's *"brain"*: its model of the behavior of a system, perceptual actions the agent can perform, and its knowledge of the world. By distinguishing between the model of system behavior and what is actually *known*, the agent can formulate planning objectives in terms of its state of knowledge, as well as the state of the world. It can then plan to achieve goals such as knowing whether a particular component is faulty or knowing whether a piece of evidence is true or false, in addition to conventional goals such as repairing a particular component. The state of knowledge of the cognitive diagnostic agent can be captured by the use of a distinguished knowledge fluent. Knowledge-producing actions can be applied to attain new knowledge from the world. Diagnosis thus becomes a planning problem, to achieve some state of knowledge; repair is planning to achieve some state of the world. Testing changes the state of knowledge, while the realization of tests can additionally change the state of the world. (Scherl and Levesque, 1993) have recently extended the situation calculus to include knowledge and have provided a solution to the frame problem for knowledge-producing actions. In this way, diagnosis, testing and repair can be integrated *within* the situation calculus.

This work will be described in detail in another paper. However, it is appropriate to contrast our cognitive diagnostic agent with previous endeavors to integrate diagnosis with repair. Friedrich et al. (as previously cited) have provided a pragmatic account of reasoning to repair an artifact. They include a procedural description for how to choose between performing simple observations and repair actions, assuming a most likely diagnosis. (Sun and Weld, 1993) 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 are interesting pieces of work from a computational perspective, but ignore some of the fundamental knowledge representation challenges in reasoning about actions and change.

Both works focus on the issue of repair. Neither system provides for the specification of diagnostic goals. Most importantly, neither provides a formal account of diagnosis, testing and repair in the context of a theory of action and change.

7 CONTRIBUTIONS

The major contributions of this paper are:
• provision of a situation calculus knowledge representation framework for diagnostic problem-solving in the context of a theory of action and change. This framework provides the expressive power to axiomatize a wide range of system behavior while solving the frame problem and addressing the ramification and qualification problems.
• significant contributions towards a formal account of diagnosis, testing, and repair, for behaviorally static systems which require world-altering actions to achieve tests and repairs.
• a new definition of *explanatory diagnosis*, which conjectures what actions or events occurred to explain the current observations.

This paper is also important because it presents the foundation for a number of extensions to the work presented herein, in particular:
• development of a cognitive diagnostic agent which incorporates knowledge-producing actions in order to integrate diagnosis, testing and repair within the logic. The agent can reason to achieve diagnostic as well as repair or contingency goals.

Future research includes:
• the utilization of GOLOG as an agent programming language. (Lesperance et al., 1994)
• extension of the characterization of diagnosis, testing and repair to deal with explicitly time-varying discrete and continuous systems.

Acknowledgements

I would like to acknowledge the Cognitive Robotics Group, University of Toronto, whose research on the situation calculus provided a foundation for this work.

References

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

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

G. Friedrich and G. Gottlob and W. Nejdl (1992). Formalizing the repair process. In *European Conference on Artificial Intelligence (ECAI-92).*

G. Friedrich and W. Nejdl (1992). Choosing observations and actions in model-based diagnosis/repair systems. In B. Nebel, C. Rich and W. Swartout (ed.), *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, 489–498. Morgan Kaufmann Publishers..

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. American Elsevier Publishers.

Y. Lesperance, H. Levesque, F. Lin, D. Marcu, R. Reiter and R. Scherl (1994). A logical approach to high-level robot programming – a progress report. In *Control of the Physical World by Intelligent Systems, Working Notes of the 1994 AAAI Fall Symposium.* To appear.

F. Lin and R. Reiter (1994). How to progress a database II: The STRIPS connection. *Manuscript.*

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

J. McCarthy and P. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, *Machine Intelligence 4*, 463–502. Edinburgh University Press.

S. McIlraith (1994). Generating tests using abduction. In J. Doyle, E. Sandewall and P. Torasso (ed.), *Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, 449–460. Morgan Kaufmann Publishers.

S. McIlraith and R. Reiter (1992). On tests for hypothetical reasoning. In W. Hamscher, L. Console and J. de Kleer (ed.), *Readings in model-based diagnosis*, 89–96. Morgan Kaufmann Publishers.

G. Provan and D. Poole (1991). The utility of consistency-based diagnostic techniques. In J. Allen, R. Fikes and E. Sandewall (ed.), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, 461–472. Morgan Kaufmann Pub.

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

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.

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

R. Scherl and H. Levesque (1993). The frame problem and knowledge-producing actions, In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 689–695.

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

Y. Sun and D. Weld (1993). A framework for model-based repair. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 182–187.