ACTIONS AND OTHER EVENTS IN SITUATION CALCULUS

John McCarthy
Computer Science Department
Stanford University
Stanford, CA 94305
jmc@cs.stanford.edu
http://www-formal.stanford.edu/jmc/

Abstract

This article presents a situation calculus formalism featuring events as primary and the usual actions as a special case. Events that are not actions are called *internal events* and actions are called *external events.* The effects of both kinds of events are given by effect axioms of the usual kind. The actions are assumed to be performed by an agent as is usual in situation calculus. An internal event e occurs in situations satisfying an *occurrence assertion* for that event.

A formalism involving actions and internal events describes what happens in the world more naturally than the usual formulations involving only actions supplemented by state constraints. Ours uses only ordinary logic without special causal implications. It also seems to be more *elaboration tolerant.*

The first example is the buzzer with only internal events and which cannot be treated at all with state constraints, because the system never settles down to a steady state.

Our second example is the stuffy room scenario. One occurrence axiom states that when both vents are blocked and the room isn't stuffy, the event $Getstuffy$ occurs. State constraints are unneeded. The stuffy room formalization tolerates an elaboration asserting that when the room becomes stuffy someone unblocks a vent. If we further add that someone else then finds the room cold and blocks the vent again, we get a system that oscillates.

The third example is the blocks world.

The nonmonotonic reasoning involves circumscribing occurrences, changes, and prevention one situation at a time.

Then we offer a general viewpoint on the situation calculus and its applications to real world problems. It relates the formalism of [MH69] which regards a situation as a snapshot of the world to situation calculus theories involving only a few fluents.

1 Introduction: Actions and other events

This article emphasizes the idea that an action by an agent is a particular kind of event. The idea of event is primary and an action is a special case. The treatment is simpler than those regarding events as *natural actions.*

The main features of our treatment are as follows.

1. There are the usual *effect axioms* involving the function $Result(e, s)$, the situation that results when event e occurs in situation s.

2. There are *occurrence axioms* giving conditions for an event to occur. They have the form conditions$(s) \rightarrow occurs(e, s)$.

3. The theory distinguishes between *external events* for which occurrence axioms are not given and *internal events* governed by occurrence axioms. Older treatments of situation calculus often do not provide for internal events. Usually human actions are properly treated as external events, but if the theory contains assertions that a person will perform a certain action, then such an assertion can be given by an occurrence axiom, and the action is an internal event. We include an example of an elaboration of the theory of Ginsberg's stuffy room scenario [GS88] that uses an occurrence axiom to assert that a person will unblock a vent when the room becomes stuffy. Thus an action

can be either an external or internal event. Elaborating the theory by giving an occurrence axiom for the action makes it an internal action in the elaborated theory.

4. Our theories are nonmonotonic and minimize certain predicates situation by situation. The approach is proposed only when information about the future is obtained only by projection from earlier situations. Thus it is not appropriate for the stolen car scenario.

We use internal events instead of state constraints in the stuffy room example. Thus we say that when the vents are blocked, the room becomes stuffy rather than regarding stuffiness as a state constraint. This is closer to human common sense reasoning and natural language usage, as well as being logically simpler.

We begin with formalizing a buzzer which has only internal events, continue with the stuffy room scenario which has both. Our third example is the blocks world.

After these examples, we discuss the nonmonotonic reasoning.

Then we offer a general viewpoint on the situation calculus and its applications to real world problems. It relates the formalism of [MH69] which regards a situation as a snapshot of the world to situation calculus theories involving only a few fluents.

2 The situation calculus formalism

Situations are denoted by the letter s decorated with subscripts. Constants like $S0$ are capitalized, and variables are lower case. However, we do not assume that all situations are generated from a big bang situation $S0$ as does Reiter [Rei01].

The sentence $Holds(pfluent, s)$ asserts that the propositional fluent $pfluent$ holds in the situation s. Sometimes we write just $pfluent(s)$, but the notation with $Holds$ allows quantifying over fluents. We also have term fluents, and $Value(tfluent, s)$ gives the value of $tfluent$ in the situation s. We also sometimes write just $tfluent(s)$.

$Result(e, s)$ denotes the situation that arises when the event e occurs in the situation s. In this simple formalism, neither situations nor events have durations.

$Occurs(e, s)$ is the assertion that the event e occurs in the situation s.

$Next(s)$ is the next situation after s. It is defined by

$$Occurs(e, s) \rightarrow Next(s) = Result(e, s) \tag{1}$$

for those situations in which an occurrence assertion determines what event will occur.

As an example, three of the axioms of the stuffy room phenomenon are

$$\begin{aligned} &Holds(Blocked1, s) \wedge Holds(Blocked2, s) \\ &\qquad \wedge \neg Holds(Stuffy, s) \\ &\qquad\qquad \rightarrow Occurs(Getstuffy, s), \\ &Holds(Stuffy, Result(Getstuffy, s)), \quad \text{and} \\ &Holds(Blocked1, Result(Block1, s)). \end{aligned} \tag{2}$$

$Getstuffy$ is an internal event and occurs all by itself when the vents are blocked.

We use circumscription to minimize occurrences, to minimize change (frame problem), and to minimize the fluents that prevent actions and other events (qualification problem).

Treating internal and external events by the same formalism admits elaborations that turn some instances of external events into internal events. Thus we can elaborate the stuffy room scenario by adjoining an occurrence axiom saying that when the room becomes stuffy, someone unblocks a vent, which makes the room unstuffy. The further elaboration that when a vent is unblocked, someone blocks it again, perhaps from feeling cold, causes the system to oscillate and never settle down.

An external event can create a situation in which the occurrence axiom for an internal event is satisfied. This leads to a new situation in which a new internal event can occur. When no more internal events occur the process *settles down*, and we can infer a statement about the resulting stable state. Stable states are usually characterized by *state constraints*. In physics these states often minimize potential energy.

The next three sections discuss examples, a buzzer which has only internal events, the stuffy room scenario, and the blocks world.

3 Formalizing a buzzer

Figure 1: A buzzer.

Figure 1 displays a buzzer consists of a relay connected to a battery by a switch that is opened when the relay operates. If the switch is on, the relay operates and opens the switch which turns off the relay which closes the switch. Thus the circuit oscillates and *never settles down* to a stable state.

The buzzer formalization has only internal events—at least once it is started, and this makes its operation easy to formalize.

State constraint axioms for formalizing a buzzer analogous to those often used for the stuffy room scenario would be immediately contradictory, asserting that the relay is on if and only if it is off. Our present situation calculus formalism follows human common sense reasoning directly and requires no special causal formalism or logic with implications not equivalent to their contrapositives.

There are effect axioms and occurrence axioms. The former are well known and give the effects of events. The latter assert that in situations in which certain fluents hold, certain events will occur.

We distinguish between the fluent $\mathrm{On}(Sw)$ asserting that the switch is on and the event $\mathrm{Onn}(Sw)$ that turns the switch on. The fluent holding in a situation is asserted by $Holds(\mathrm{On}(Sw), s)$. Likewise for the fluent $\mathrm{On}(R)$ and the event $\mathrm{Onn}(R)$ that concern the relay. We also have Off and Offf for the switch and the relay.

Effect axioms:

$$\begin{array}{l} Holds(On(R), Result(Onn(R), s)) \\ \neg Holds(On(R), Result(Offf(R), s)) \\ Holds(On(Sw), Result(Onn(Sw), s) \\ \neg Holds(On(Sw), Result(Offf(Sw), s)). \end{array} \tag{3}$$

Occurrence axioms:

$$\begin{array}{l} \neg Holds(On(Sw), s) \wedge Holds(On(R), s) \\ \qquad \rightarrow Occurs(Offf(R), s) \\ Holds(On(Sw), s) \wedge \neg Holds(On(R), s) \\ \qquad \rightarrow Occurs(Onn(R), s)) \\ Holds(On(R), s) \wedge Holds(On(Sw), s) \\ \qquad \rightarrow Occurs(Offf(Sw), s) \\ \neg Holds(On(R), s) \wedge \neg Holds(On(Sw), s) \\ \qquad \rightarrow Occurs(Onn(Sw), s) \end{array} \tag{4}$$

Note that each of the above occurrence axioms has a second term in the precondition. They are needed to avoid unwanted concurrent events.

Frame assertions—for now axioms:

$$\begin{array}{l} e = Onn(Sw) \vee e = Offf(Sw) \\ \qquad \rightarrow Holds(On(R), Result(e, s)) \\ \qquad\qquad \equiv Holds(On(R), s)). \end{array} \tag{5}$$

$$\begin{array}{l} e = Onn(R) \vee e = Offf(R) \\ \qquad \rightarrow Holds(On(Sw), Result(e, s)) \\ \qquad\qquad \equiv Holds(On(Sw), s)). \end{array} \tag{6}$$

These frame assertions tell what doesn't change. They are few enough in this case, since there are few actions and few fluents. In general it is more efficient to say what does change. In this case we have

$$\begin{array}{l} Changes(Onn(R), On(R), s), \\ Changes(Offf(R), On(R), s), \\ Changes(Onn(Sw), On(Sw), s), \quad \text{and} \\ Changes(Offf(Sw), On(Sw), s). \end{array} \tag{7}$$

In section 6 we describe how to get the frame assertions by circumscribing $Changes(e, f, s)$.

Let an initial situation, called $S0$, be given by

$$\neg Holds(On(Sw), S0) \wedge \neg Holds(On(R), S0) \tag{8}$$

We can proceed a step at a time. We have

$$Occurs(\mathrm{Onn}(Sw), S0) \tag{9}$$

in accordance with (4). Hence

$$Next(S0) = Result(Onn(Sw), S0), \tag{10}$$

and therefore, letting

$$S1 = Next(S0), \tag{11}$$

we have

$$\neg Holds(\mathrm{On}(R), S1) \wedge Holds(\mathrm{On}(Sw), S1). \tag{12}$$

Some elaborations of the buzzer axioms will be worth doing.

1. Allow the action of stopping the buzzer to occur at any situation.

2. Consider the action of stopping the buzzer as a concurrent event.

3. A concurrency elaboration along the lines of [McC92] and [MC98] might be to have two non synchronized buzzers B1 and B2 with no guaranteed temporal relation between the events involving B1 and B2.

4 The stuffy room scenario

A problem arises when the well-known stuffy room scenario is formalized with a state constraint that when both vents are blocked by pillows the room is stuffy and changes in fluents are minimized. This can lead to the unintended model that when one vent is already blocked the action of blocking the other event causes the blocked vent to become unblocked in order to minimize change. Some complication of the formalism is required to deal with the phenomenon. Direct formalization in terms of actions and events avoids the difficulty. Also it corresponds better to the way we humans think about the problem, i.e. we think about the room becoming stuffy.

We use fluents Blocked1, Blocked2, and Stuffy. We have the action events Block1, Unblock1, Block2, Unblock2 and the internal events Getstuffy and Ungetstuffy. [1]

Effect axioms:

$$\begin{array}{l} Holds(Blocked1, Result(Block1, s)) \\ Holds(Blocked2, Result(Block2, s)) \\ \neg Holds(Blocked1, Result(Unblock1, s)) \\ \neg Holds(Blocked2, Result(Unblock2, s)) \\ Holds(Stuffy, Result(Getstuffy, s)) \\ \neg Holds(Stuffy, Result(Ungetstuffy, s)) \end{array} \tag{14}$$

Occurrence axioms:

$$\begin{array}{l} Holds(Blocked1, s) \wedge Holds(Blocked2, s) \\ \qquad \wedge \neg Holds(Stuffy, s) \\ \qquad\qquad \rightarrow Occurs(Getstuffy, s) \quad \text{and} \\ (\neg Holds(Blocked1, s) \vee \neg Holds(Blocked2, s)) \\ \qquad\qquad \wedge Holds(Stuffy, s) \\ \qquad \rightarrow Occurs(Ungetstuffy, s) \end{array} \tag{15}$$

The frame axioms are

$$\begin{array}{l} Changes(Block1, Blocked1, s), \\ Changes(Block2, Blocked2, s), \\ Changes(Unblock1, Blocked1, s), \\ Changes(Unblock2, Blocked2, s), \\ Changes(Getstuffy, Stuffy, s), \quad \text{and} \\ Changes(Ungetstuffy, Stuffy, s). \end{array} \tag{16}$$

[1] One of the referees suggested that using the fluents *Blocked*1 and *Blocked*2 and the corresponding actions was too special, and we should say that the room is stuffy when all the vents are blocked. We can accommodate his preference by introducing the vents as objects and using the axiom

$$\begin{array}{l} (\forall vent)(Holds(Blocked(vent), s)) \\ \qquad \rightarrow Occurs(Getstuffy, s) \end{array} \tag{13}$$

and a corresponding axiom for the effect of unblocking a vent.

This is just a step towards a general commonsense theory of the effects of ventilation on stuffiness. Such a theory would have to take into account the fact that blocking the vents does not make the room stuffy under all circumstances. For now it's simpler to just consider the particular room with exactly two vents.

How they work is described in section 6.

We need to distinguish between *internal events* like $Getstuffy$ and *external events* like $Block1$. As we shall see, an external event may be an internal event of a more comprehensive narrative, e.g. one in which $Block1$ occurs when Mike is annoyed by cold air coming from $Vent1$.

We can tell a simple sequential story by first describing $S0$, e.g. by

$$\begin{array}{l} \neg Holds(Blocked1, S0) \wedge \neg Holds(Blocked2, S0) \\ \qquad \wedge \neg Holds(Stuffy, S0). \end{array} \tag{17}$$

We can now write the narrative

$$\begin{array}{l} S1 = Result^*(Block1, S0) \\ S2 = Result^*(Block2, S1) \\ S3 = Result^*(Unblock2, S2) \\ S4 = Result^*(Block2, S3), \quad \text{etc.} \end{array} \tag{18}$$

Here $Result^*(e, s)$ is like the Rr of [McC95]. It is the result of doing a followed by the occurrence of whatever internal events occur. The assumption is that some sequence of internal events will occur after which the situation remains the same until another external event occurs. $Result^*(e, s)$ is undefined in the buzzer example in which internal events occur forever.

$Result^*$ requires an induction axiom or schema. Here's one candidate:

$$\begin{array}{l} P(s) \wedge (\forall s\ e)(P(s) \wedge Occurs(e, s) \rightarrow P(Result(e, s))) \\ \rightarrow P(Result^*(e, s)). \end{array} \tag{19}$$

The function $Next^*$ has the same relation to $Result^*$ that $Next$ has to $Result$. It gives the next situation to which no occurrence assertion applies. $Next^*$ satisfies

$$\begin{array}{l} Result^*(e, s) = Next^*(Result(e, s)), \\ (\forall e)(\neg Occurs(e, s)) \rightarrow Next^*(s) = s, \quad \text{and} \\ Occurs(e, s) \rightarrow Next^*(s) = Next^*(Result(e, s)). \end{array} \tag{20}$$

In the present case we will have

$$S1 = Result^*(Block1, S0) = Result(Block1, S0).$$

because no internal event will occur in $Result(Block1, S0)$. However, we'll have

$$\begin{array}{l} S2 = Result^*(Block2, S1) \\ \qquad = Result(Getstuffy, Result(Block2, S1)), \end{array} \tag{21}$$

because now the internal event $Getstuffy$ will occur. Thus we'll have $\neg Holds(Stuffy, S1)$ but $Holds(Stuffy, S2)$, $\neg Holds(Stuffy, S3)$, and $Holds(Stuffy, S4)$.

We can write

$$\begin{aligned} S4 = &Result^*(Block2, Result^*(Unblock2, \\ &\quad Result^*(Block2, Result^*(Block1, S0)))) \\ = &Result(Getstuffy, Result(Block2, \\ &\quad Result(Ungetstuffy, Result(Unblock2, \\ &\quad Result(Getstuffy, Result(Block2, \\ &\quad\quad Result(Block1, S0))))))), \end{aligned} \tag{22}$$

which can also be written

$$\begin{aligned} S4 &= Result^*(Block1; Block2; Unblock2; Block2, S0) \\ &= Result(Block1; Block2; Getstuffy; Unblock2; \\ &\quad Ungetstuffy; Block2; Getstuffy, S0). \end{aligned} \tag{23}$$

Here we extend the meaning of $Result$ to allow a sequence of events as an argument.

4.1 Telling stories using $Occurs$ and $Next$

Another way of telling stories is to always use $Occurs$. An external event is axiomatized by asserting that it occurs.

The above story is then given by

$$\begin{aligned} &Occurs(Block1, S0) \\ &S1 = Next(S0) = Result(Block1, S0) \\ &Occurs(Block2, S1) \\ &S1' = Next(S1) = Result(Block2, S1) \\ &Occurs(\text{Getstuffy}, S1'), \text{by inference} \\ &S2 = Next(S1') = Result(Getstuffy, S1') \\ &Occurs(Unblock2, S2) \\ &S2' = Next(S2) = Result(Unblock2, S2) \\ &Occurs(Ungetstuffy, S2')\text{by inference} \\ &S3 = Next(S2') = Result(Ungetstuffy, S2') \\ &Occurs(Block2, S3) \\ &S3' = Next(S3) = Result(Block2, S3) \\ &Occurs(Getstuffy, S3')\text{by inference} \\ &S4 = Next(S3') = Result(Getstuffy, S3'). \end{aligned} \tag{24}$$

We can also write the story more briefly as

$$\begin{aligned} &Occurs(Block1, S0) \\ &S1 = Next^*(S0) = Result^*(Block1, S0) \\ &Occurs(Block2, S1) \\ &S2 = Next^*(S1) = Result^*(Block2, S1) \\ &Occurs(Unblock2, S2) \\ &S3 = Next^*(S2) = Result^*(Unblock2, S2) \\ &Occurs(Block2, S3) \\ &S4 = Next^*(S3) = Result^*(Block2, S3) \end{aligned} \tag{25}$$

Still more briefly

$$\begin{aligned} S4 &= Next^*(Next^*(Next^*(Next^*(S0)))) \\ &= Result^*(Block2, Result^*(Unblock2, \\ &\quad Result^*(Block2, Result^*(Block1, S0)))) \end{aligned} \tag{26}$$

4.2 Two elaborations of the stuffy room scenario

The first elaboration says that when Pat finds the room stuffy he unblocks vent2. We have

$$Holds(Stuffy, s) \rightarrow Occurs(Does(Pat, Unblock2), s), \tag{27}$$

or, more elaborately,

$$\begin{aligned} &Holds(Stuffy, s) \wedge \neg Holds(Uncomforable\text{-}Pat, s) \\ &\quad \rightarrow Occurs(Becomes\text{-}Uncomfortable(Pat), s), \\ &Holds(Uncomfortable, Pat, \\ &\quad Result(Becomes\text{-}Uncomforable(Pat), s)) \\ &Holds(Uncomfortable, Pat, s) \\ &\rightarrow Occurs(Does(Pat, Unblock\text{-}Vent2), s), \\ &\neg Holds(Blocked2, Result(Does(Pat, Unblock\text{-}Vent2), s)). \end{aligned} \tag{28}$$

(24) remains the same except that perhaps we should change the notation so that instead of $S3$ and $S3'$ we write $S2''$ and $S2'''$, since these are now intermediate situations. The situation $S4$ is now unstable.

Now let's add a second elaboration in which Mike finds the room cold when there is an unblocked vent and blocks vent2. It is expressed by adding

$$Holds(Unstuffy, s) \rightarrow Occurs(Does(Mike, Block2), s). \tag{29}$$

With both of these elaborations, we get an oscillation; Pat unblocks vent2 and Mike blocks it again. $Result^*$ and $Next^*$ are no longer defined.

5 The blocks world

Assume enough unique names axioms.

The blocks world involves the frame problem in a more significant way than do the buzzer and the stuffy room scenarios.

We use the predicate $Prevents(p, e, s)$ to say that a move is prevented by there being a block on top of the block to be moved or on the destination unless the destination is the table. We thereby skip the use of the fluent $Clear(x)$ prevalent in many blocks world sitcalc theories.

Here's the effect axiom for moving a block.

$$
\begin{array}{l}
(\forall p)(\neg(Prevents(p, Move(x, y), s) \wedge Holds(p, s))) \\
\quad \rightarrow \\
(Holds(On(x, y), Result(Move(x, y), s))) \\
\wedge \\
((Holds(On(x, z), s)) \wedge z \neq y \\
\qquad \rightarrow \neg Holds(On(x, z), Result(Move(x, y), s)))),
\end{array} \tag{30}
$$

and here are the axioms for prevention:

$$
\begin{array}{l}
Prevents(On(z, x), Move(x, y), s) \quad \text{and} \\
y \neq Table \rightarrow Prevents(On(z, y), Move(x, y), s).
\end{array} \tag{31}
$$

We adopt the usual way of emphasizing the frame problem by introducing the action of painting a block a certain color. Thus

$$
\begin{array}{l}
(\forall p)(\neg(Prevents(p, Paint(x, c), s) \wedge Holds(p, s))) \\
\quad \rightarrow Holds(Color(x, color), Result(Paint(x, color), s)),
\end{array} \tag{32}
$$

or, using object valued, i.e. non propositional, fluents,

$$
\begin{array}{l}
(\forall p)(\neg(Prevents(p, Paint(x, c), s) \wedge Holds(p, s))) \\
\quad \rightarrow Value(Color(x), \\
\qquad Result(Paint(x, color), s)) = color.
\end{array} \tag{33}
$$

The change axioms for the blocks world are

$$
\begin{array}{l}
Changes(Paint(x, c), Color(x), s), \\
Holds(On(x, z), s) \\
\quad \rightarrow Changes(Move(x, y), On(x, z), s) \\
\wedge Changes(Move(x, y), On(x, y), s).
\end{array} \tag{34}
$$

The nonmonotonic reasoning associated with the blocks world will be discussed after the section dealing with nonmonotonic reasoning in situation calculus in general.

6 Nonmonotonic reasoning—situation by situation

We use circumscription to minimize the events that occur in a situation, the fluents that might prevent an event from having its standard effect, and the changes in fluents. In contrast to the formalism of [McC86] which minimized predicates over all the arguments, we minimize for each successive situation separately. However, in doing this minimization in s we take as fixed the $Holds(f, s)$ sentences and the $Value(exp, s) = \ldots$ sentences inferred from the effects of the event that led to the situation. We are giving up the possibility of trading and abnormality in one situation for an abnormality in another.

Doing the nonmonotonic reasoning in situations successively corresponds to the way people predict the consequences of sequences of actions and events. It seems to give the same conclusions as Yoav Shoham's chronological minimization [Sho88] but is computationally more straightforward. Like chronological minimization, it avoids the Yale shooting problem and its friends.[2]

However, we advocate this only for projection problems, i.e. reasoning about the future from information about the past. The method is not appropriate for the *stolen car scenario* in which one has to reason from an assertion (that the car is missing) about a later situation. [3]

With the present formalism, the person or agent setting up the problem must know that projection forward in time is appropriate. It would be better if this were a consequence of the formalized facts.

Now let's consider circumscribing at each situation separately. The simplest case is when we have a predicate $Foo(x, y, s)$.

We write the axioms

$$
\begin{array}{l}
Foo' \leq_s Foo \equiv (\forall x\ y)(Foo'(x, y, s) \rightarrow Foo(x, y, s)), \\
(Foo' <_s Foo) \equiv (Foo' \leq_s Foo) \wedge \neg(Foo' =_s Foo), \\
Foo' =_s Foo \equiv (\forall x\ y)(Foo'(x, y, s) \equiv Foo(x, y, s)).
\end{array} \tag{35}
$$

Then the circumscription of $Foo(x, y, s)$ takes the form

$$
\begin{array}{l}
Axiom(Foo, vars, s) \wedge (\forall foo'\ vars')(Axiom(foo', vars') \\
\qquad \rightarrow \neg(foo' <_s Foo)).
\end{array} \tag{36}
$$

Here $vars$ stands for a list of the entities being varied as Foo is minimized.

[2] The ideas of internal and external events of the preceding sections are independent of the formalism used for nonmonotonic reasoning. For example, Golog [Rei01] or the Causal Calculator [aA01] could be used—perhaps with some modifications for the buzzer and the oscillating stuffy room.

[3] Actually part of the stolen car scenario can be treated provided we don't suppose that the car being missing is to be projected from information about the past. Certainly we can go forward from the situation in which the car is missing to further events in the future. Likewise, in the story of Junior's travels [McC92], we can assert that Junior loses his ticket to Moscow in London and reason forward from that fact.

This spells out to

$$\begin{aligned}
&Axiom(Foo, vars, s) \wedge (\forall foo' \; vars') \\
&(Axiom(foo', vars') \wedge ((\forall x \; y)(foo'(x, y, s) \\
&\qquad \rightarrow Foo(x, y, s)) \\
&\rightarrow (\forall xy)(Foo(x, y, s) \equiv foo'(x, y, s)))).
\end{aligned} \tag{37}$$

Call this formula $Circ(Axiom; Foo; vars; s)$. This is the notation of [Lif94] with the addition of the argument s to say that s is kept fixed.

The general frame axioms are

$$\begin{aligned}
&\neg Changes(e, p, s) \\
&\qquad \rightarrow (Holds(p, Result(e, s)) \equiv Holds(p, s))
\end{aligned} \tag{38}$$

for propositional fluents and

$$\begin{aligned}
&\neg Changes(e, f, s) \\
&\qquad \rightarrow Value(f, Result(e, s)) = Value(f, s).
\end{aligned} \tag{39}$$

for general fluents.

Suppose we allow complex fluents, say $p \; And \; q$ when p and q are propositional fluents. We then need an axiom

$$\begin{aligned}
&Changes(e, p, s) \vee Changes(e, q, s) \\
&\qquad \rightarrow Changes(e, p \; And \; q, s).
\end{aligned} \tag{40}$$

Similar axioms are required for the other propositional functions of fluents and for the compositions of non-propositional fluents.

[This leads to difficulties when we want to delimit what changes, since there are arbitrarily complex compositions of fluents. We'll confine ourselves to elementary fluents for now by not putting compositions in the language.]

In these circumscriptions we also minimize $Holds$.

This tolerates elaborations like

$$Holds(Weak, s) \rightarrow Prevents(Weak, Move(x, y), s). \tag{41}$$

If $Holds(Weak, s)$ isn't asserted, $Move(x, y)$ will not be prevented.

Lin and Shoham, [LS95] consider a theory of action to be *provably correct* if doing the nonmonotonic reasoning results in a complete nonmonotonic theory of the action. This seems like a worthy goal, but I don't know if the present theory achieves it.

7 Actions and other events

The previous sections presented a formalism adequate for the examples discussed. In this section we discuss the situation calculus in general and its connection with the real world. We also discuss relations between different situation calculus theories, e.g. theories at different levels of detail, [4] with actions by agents as a special case. Thus an action term a is considered an abbreviation of the event term $Does(person, a)$. Besides *effect axioms* formalizing $Result(e, s)$ [$do(e, s)$ in Canada and its colonies], there are *occurrence axioms* asserting that in situations satisfying certain expressions in the fluents, an event e occurs—written $Occurs(e, s)$. [5]

Before giving effect and occurrence axioms, we present some general considerations concerning situation calculus and its applications.

7.1 Situation calculus and the real world

There have been many formulations of situation calculus.

[MH69] regarded a situation as a snapshot of the world at some instant of time. Such a system could not be known and described completely, but a person or program could know facts about a situation, i.e. the values of some fluents, and could infer some consequences of some actions from these facts. Situations are examples of *rich entities*, i.e. entities involving more detail than can be specified. *Poor entities* have finitely describable structures.

However, theories of action and change[6] often use a more limited notion of situation. Thus Raymond Reiter [Rei01] and his colleagues regard situations as the nodes of a tree based at an initial situation $S0$ and whose edges branching from a situation s are the ac-

[4] [McC59] proposed mathematical logic as a tool for representing facts about the consequences of actions and using logical reasoning to plan sequences of actions that would achieve goals. Situation calculus as a formalism was proposed in [McC63] and elaborated in [MH69]. The name "situation calculus" was first used in [MH69] but wasn't defined there. [McC86] proposed to solve the frame and qualification problems by circumscription, but the proposed solution to the frame problem was incorrect. [Sha97] and [Rei01] describe several situation calculus formalisms and give references.

[5] I suspect I need to pound the table a little here. Actions are just a kind of event, and formalized *reasoning about actions and change* need to treat events as the general case and those events which are actions as special. This has long seemed obvious to me, but I find that many other researchers don't want to use the same formalism for events that are not actions of agents and those which are.

The consequence has been the introduction of extensions to logic for treating what are called domain constraints, most of which are better treated by formalizing events.

[6] "events and change" would be better terminology

tions that may be taken in s. Other researchers, including Murray Shanahan [Sha97] and myself, use $S0$ as just a name for some situation whose consequences are of interest.

The viewpoint of this article is that a situation s is arbitrary element of a space $Sits$ of situations, i.e. s bears the same relation to $Sits$ as a group element bears to a group. Situation calculus theories relate situations, fluents and action by axioms, i.e. are abstract structures satisfying the theory.

A robot can use a *poor* situation calculus theory T to decide what to do in a world of rich situations. For example, the robot's blocks world theory may only allow specifying that one block is on another, not where it is located on the other. Suppose we have a mapping $Observe$ from a subset of rich situations to poor situations. When the robot observes a world situation s to which the theory T applies, it obtains a poor situation $Observe(s) \in Sits(T)$. Using the theory T, the robot infers that a certain action a will advance its goal. It then performs an action $Execute(a)$ in the world. If the theory T corresponds to the world properly, $Result(Execute(a), s)$ will be an improved situation.

It isn't the purpose of this paper to develop a theory of the correspondence between rich real world situations and those of limited sitcalc domains. However, the way we formalize sitcalc is motivated by the hope of making these correspondences in a later theory.

Whether an event is external depends on the theory. If we can formulate when an event e will occur, then we can make our theory more powerful by including an occurrence axiom for that event. If we assume a deterministic world, the limiting case is a theory in which all events are internal.

8 Elaboration tolerance

An important feature of human common sense is that human knowledge of a phenomenon is often readily elaborated to take new information into account. It is important that logical theories of common sense phenomena also have this property. [McC99] has a detailed discussion.

Situation calculus theories benefit from several kinds of elaboration. Section 4 discusses elaborating the stuffy room theory by adding occurrence axioms for a person being motivated to open a vent when the room becomes stuffy. [McC92] constructs a theory of a persons travel planning which can be elaborated by adding a sentence asserting that he loses his airplane ticket at a certain point in his journey. Because the reasoning depends on minimizing occurrences, we can no longer conclude that the original travel plan will succeed.

In general, elaboration tolerance concerns making it easy to modify a theory, but the simplest kind of elaboration is to add one or more sentences to an existing theory. It is desirable that elaborations be doable in this way as much as possible. [McC99] discusses when this can and cannot be done for a given theory and how to make theories for which elaboration by conjoining sentences is possible. Theories expressed in natural language have this kind of elaboration tolerance to a high extent.

9 Extensions of the formalism and problems they present

The basic situation calculus admits many useful extensions. The ideas of this section are tentative.

9.1 Concurrency

There are two limiting cases of concurrency that can be treated in the situation calculus.

Easy concurrency:

Two or more events, say e_1 and e_2 occur in a situation s and result in the same next situation $Next(s)$. The fluents that hold in $Next(s)$ are those determined by the effect axioms for $e1$ and $e2$ separately. Thus if we move a block and paint it concurrently, it will have both the new location and the new color in $Next(s)$.

General concurrency:

Two *processes*, starting, say from initial situations $S0$ and $S0'$ take place and affect different sets of fluents. If nothing is said about the timing of the processes and no axioms of interaction are given, nothing can be inferred about the relative timing of the processes. Moreover, what can be inferred about the values of the fluents in successive situations is exactly what can be inferred by the processes taken separately. Thus Louis Pasteur was elected to the French Academy of Sciences in 1862 concurrently with certain battles of the American Civil War, but historians mention neither process in connection with the other. This is a limiting case, i.e. the case of zero interaction. Two theories of separate processes can be combined by taking the conjunction of their axioms. The combined theory is a *conservative extension* of each separate theory. It can be useful to elaborate the combined theory by giving axioms for the interaction. [McC95] and [MC98] treat elaborating theories of two non-interacting processes by adding axioms of interaction. Those articles treat Junior traveling in Europe and Daddy stacking gold blocks in New York. There is no interaction until we adjoin assertions about Junior losing an airplane ticket and asking Daddy for money, thus forcing Daddy to sell one of the blocks he was stacking.

We hope to combine the ideas of the two above-mentioned articles with those of this article in future work.

9.2 Events whose occurrence depends on the past

Suppose we want George to unblock both vents when the room becomes stuffy. When he has unblocked one vent, the room becomes unstuffy, so the physical situation is as it was when he blocked the first vent, so he needs to remember that the room was previously stuffy. We can make occurrences depend on past situations by adding for each event e an additional effect axiom

$$Past(Result(e, s)) = s. \tag{42}$$

Notice that $Past(Past, s))$ is the situation two events back.

We can have George unblock Vent1 after he has unblocked Vent2 and the room has become unstuffy by introducing the occurrence axiom

$$Stuffy(Past(Past(s)) \rightarrow Occurs(Unblock1, s). \tag{43}$$

The history as just described does not say what events occurred. This information is provided by having for each event e the axiom

$$Lastevent(Result(e, s)) = e. \tag{44}$$

Notice that this formalization is noncommittal as to whether the information is in an actor's memory.

This seems neat, and maybe it will be useful.

9.3 "Branching time" and "linear time"

We can tell a story by saying what occurs in each situation. In some situations what occurs is determined by an occurrence action and depends on the fluents holding in the situation. In other situations, we simply provide an axiom, e.g. $Occurs(Birth(\text{Benjamin-Franklin}), S1806)$. This is a linear time theory.

However, linear time and branching time are sometimes appropriately used together. Suppose we wish to say that the actor will take the low road or the high road according to which will get him to Scotland first. We can write

$$\begin{aligned}&\textbf{if } [\text{Arrival-time}(Result(\text{Take-Low-Road}, s))\\ &\qquad \leq \text{Arrival-time}(Result(\text{Take-High-Road}, s))]\\ &\textbf{then } Occurs(\text{Take-Low-Road}, s)\\ &\textbf{else } Occurs(\text{Take-High-Road}, s).\end{aligned} \tag{45}$$

Here we have used a branching time criterion for a linear time action.

This is not as elaborate as actual human behavior in which mental events occur calculating which route will lead to earliest arrival.

9.4 Induction in the situation calculus

Several kinds of mathematical induction seem to be required. For example, one may want to prove a proposition $P(Next^*(s))$ by showing that it is true for s and is preserved by the events that occur between s and $Next^*(S)$. A related kind of induction is needed to prove that something is true for all situations arising in the operation of a buzzer. The simplest case of the $Next^*$ induction might be to show that a block unmoved by each of a sequence of events is in the same position in $Next^*(s)$.

The simplest situation calculus is Reiter's [Rei01]. The formula is

$$[P(S0) \wedge ((\forall a\ s)(P(s) \rightarrow P(Result(a, s))))] \rightarrow (\forall s)P(s). \tag{46}$$

Here are two formulas

$$\begin{aligned}&[P(s) \wedge ((\forall e\ s)(P(s) \wedge Occurs(e, s) \rightarrow P(Next(s))))]\\ &\rightarrow P(Next^*(s)).\end{aligned} \tag{47}$$

(47) is appropriate when $Next^*(s)$ is defined.

When $Next^*(s)$ is not defined, as in the buzzer case, we can use $s \leq s'$ to mean that s' is a distant successor of s and have the axiom.

$$\begin{aligned}&[P(s) \wedge s \leq s'\\ &\qquad \wedge((\forall e\ s)(P(s) \wedge Occurs(e, s) \rightarrow P(Next(s))))]\\ &\rightarrow P(s').\end{aligned} \tag{48}$$

9.5 Formalizing Oscillations

The buzzer oscillates, i.e. the situation repeats again and again. So does the stuffy room scenario with the two elaborations that cause Vent2 to become blocked and unblocked repeatedly. However, we don't need a complete repetition of the situation to have oscillation. Suppose, or example, we add a clock to the buzzer, a natural number valued fluent that each event increments by 1. Then although the whole situation would not repeat, we would still want to consider the system as oscillatory.

This suggests a relative notion of oscillatory, i.e. oscillatory with respect to certain fluents.

Moreover, we would like to consider the buzzer as oscillating even if we provide for it stopping its oscillation by being turned off.

As we have described the buzzer, it cannot be turned off. Likewise the stuffy room process cannot be changed once we have added the elaborations about people blocking and unblocking the vent. See (27) and (29).

Here's a way of putting interventions into the formalism.

Let a be an action, e.g. stopping that damn buzzer. The following two axioms describe an elaboration that interpolates an action after a normal internal action. In the buzzer case it would be opening an additional switch in the circuit. The additional switch isn't in Fig. 1 or described in section 3.

$$\begin{aligned}&Occurs(a,s) \wedge External(a) \wedge Occurs(e,s)\\ &\rightarrow Next(s) = Result(a, Result(e,s))\end{aligned} \tag{49}$$

and

$$\begin{aligned}&Occurs(e,s) \wedge (\forall e')(Occurs(e',s) \rightarrow e' = e)\\ &\qquad \rightarrow Next(s) = Result(e,s).\end{aligned} \tag{50}$$

This is a limited kind of concurrency. Only certain kinds of interventions can be done this way.

9.6 State constraints after all

As was shown in Section 4, the condition for a room being stuffy is better formalized with effect axioms, occurrence axioms, and the events $Getstuffy$ and $Ungetstuffy$. Lin and Reiter [LR94] consider the Emperor's decree that no more than one object (block) be yellow, which may be regarded as a domain constraint. They point out that it is more efficient to encode the constraint as a precondition that a block may be painted yellow only if no block is already yellow. Their way of expressing this does not readily elaborate to require that no more than seven blocks be yellow.

I think logical AI needs a more complex treatment. It seems to me that efficiency conflicts with generality. It is bad or dangerous to have more than one yellow block, but perhaps only if one is not a special favorite of the emperor or if one is just about to die anyway. The point is that common sense (at least human level common sense) requires that such constraints tolerate elaboration. Human level common sense also allows the constraint to become an action precondition as a result of some inference. This inference should take place within the logical formalization.

Lin and Reiter include the following formula.

$$\begin{aligned}&(\forall x\ y\ s)(Poss(Paint(x,y),s)\\ &\equiv (Nearby(x,s) \wedge Haspaint(y,s)\\ &\wedge(\forall x_1)(Color(x_1, Yellow, s) \wedge y = Yellow \rightarrow x = x_1))).\end{aligned} \tag{51}$$

This formula is specialized to the emperor tolerating just one yellow block. If he tolerates 7 yellow blocks, we had better use set notation, i.e. refer to $card(\{x|Color(x, Yellow)\}) \leq 7$.[7]

There are some domain constraints that are not naturally formalized by internal actions. One is the blocks world constraint that a block may not be on top of itself. Formulas like

$$Above(Top(block), Bottom(block), s) \tag{52}$$

or even

$$\begin{aligned}&Height(Top(block), s) - Height(Bottom(block), s)\\ &\qquad \geq 1.0cm\end{aligned} \tag{53}$$

tell more about the world than the simple

$$\neg On(block, Top(block), s). \tag{54}$$

An important application for the direct use of state constraints is when an event starts a process that eventually leads to an equilibrium state. For example, if I drop a coin on the floor it will bounce around for a while and then settle down. It will reach equilibrium in a second or so, and I am interested in whether the coin ends up heads or tails rather than in the process of its settling down. In the case of the coin the equilibrium condition, at least what we want to know about it, is easy to state, namely

$$\begin{aligned}&On(coin, floor, Result^*(Drop(coin,s)))\\ &\wedge(Heads(coin, Result^*(s))\\ &\vee Tails(coin, Result^*(s))),\end{aligned} \tag{55}$$

where using $Result^*$ means that we are skipping by some internal events, in this case not formalized.

[7] I pound the table here because of some resistance to the idea that axiomatic set theory makes logical AI easier.

Another example may be concocted from the elaborated stuffy room scenario. While Pat and Mike disagree in their preferences, under normal circumstances we can suppose they will come to an agreement in some short time. One will defer to the other in the matter of the blocked vents. As with the coins, the theory of eventual agreement doesn't predict what the agreement will be.

More generally, Aarati Parmar suggests that internal events are evoked by any non-equilibrium situations.

9.7 Javier Pinto's formalism

The work closest to the present is [Pin98b], as one of the referees forcefully pointed out. There are substantial differences, both in approach and in the formalisms motivated by the different approaches.

Pinto uses the Reiter notion of situations as trees built from the initial situation $S0$ by iterations of forming $do(a, s)$ where a is an action and s a previously formed situation term.

Pinto (as does Reiter) builds time, represented by a real number, into his situation calculus formalism. It seems to me that making time fit the tree structure of situation terms leads to complications. Pinto has five different *occur* predicates, whereas we have only one. His occurrence axioms all have time parameters. Our occurrence axioms involve only situations and fluents and are therefore simpler. The examples of the present article do not involve time explicitly. When time must be explicit, we propose to treat the passage of time as an independent situation calculus process running concurrently with the processes we are treating.

Pinto includes the following interesting examples. We show how our method treats a few of them.

1. "The sun will rise tomorrow at 6:03 am." Here we have two concurrent processes: the passage of time and the path of the sun through the sky. The sentence describes an interaction.

We can represent the sentence by

$$\begin{aligned}&Value(Time, s) = Time(Tomorrow603am)\\ &\quad\rightarrow Occurs(Sunrise, s),\end{aligned}$$

where we are not taking into account the explicit indexical of tomorrow, and the implicit indexical that sunrise being a 6:03am must refer to a specific latitude and longitude.

2. "If you eat the forbidden fruit you will be expelled." Pinto treats this as one event causing another but remarks that it might be better formalized as a state, i.e. that of having eaten the fruit, giving rise to an event. That's how the present paper would treat it, i.e.

$$\begin{aligned}&Holds(Has\text{-}occurred(Eat(Forbidden\text{-}fruit)), s)\\ &\rightarrow Occurs(Does(God, Expel(Eater)), s),\end{aligned} \tag{56}$$

together with the general moving finger axioms

$$\begin{aligned}&Occurs(e, s) \rightarrow Holds(Has\text{-}occurred(e, Next(s))),\\ &\text{and}\quad Holds(Has\text{-}occurred(e, s) \wedge s < s'\\ &\qquad\rightarrow Holds(Has\text{-}occurred(e, s')).\end{aligned} \tag{57}$$

3. "The train to Ottawa leaves every day at 7 pm." where it is understood that this scheduled event may not occur under exceptional circumstances.

$$\begin{aligned}&Value(Time, s) = Time(7pm)\\ &\qquad\wedge\neg Prevented(Train\text{-}Leaves\text{-}for\text{-}Ottowa, s)\\ &\quad\rightarrow Occurs(Train\text{-}Leaves\text{-}for\text{-}Ottowa, s).\end{aligned}$$

4. "If my neighbor's burglar alarm goes off while I am at home, I will call the police." Pinto treats this example and the previous one by slightly different formalisms, one involving a predicate $occurs_{po}(action, time)$ and the other a predicate $occurs_{ct}(action, time, action2)$.

5. The Miller-Shanahan [RM94] example of the briefcase.

6. "My house has a burglar alarm. If the alarm is connected, I have exactly 60 seconds to deactivate it after opening the main door. If I am unable to disconnect the alarm, it will go off."

7. "Upon an insertion into EMP or an update to EMP, the new SAL is checked, and if it exceeds $100,000, then the JobTitle of this employee is added to HPAID, assuming it was not there already."

$$\begin{aligned}&Holds(Checksalary, Result(Insert(EMP, y, s)))\\ &\wedge Holds(Checksalary, Result(Update(EMP, y, s))).\end{aligned}$$

$$\begin{aligned}&Holds(Checksalary(employee, s) \rightarrow\\ &Occurs(Add(JobTitle(employee), HPAID), s)\end{aligned}$$

[Pin98a] introduces $occurs(a, s)$, where a is a "natural action". Natural actions partly correspond to internal events. The article is dedicated to concurrent events, to which I hope devote a separate article.

10 Concluding remarks

Events that are not actions have been previously used—at least by Fangzhen Lin [Lin98], Sheila McIlraith [McI00], and Javier Pinto.

Occurrence axioms are even more important in the treatment of concurrent events in situation calculus—to be the subject of another article.

This work benefited from discussions with Eyal Amir, Tom Costello, Ron Fadel, Hector Levesque, Vladimir Lifschitz, Fangzhen Lin, Sheila McIlraith, Leora Morgenstern, Aarati Parmar, Raymond Reiter, and Tran Son and the comments of three anonymous referees.

This research was partly supported by SRI Subcontract No. 34-000144 under SPAWAR Prime Contract No. N66001-00-C-8018.

References

[aA01] Texas Action Group at Austin. Causal calculator home page, 2001. http://www.cs.utexas.edu/users/tag/cc.

[GS88] Matthew L. Ginsberg and David E. Smith. Reasoning about action I: A possible worlds approach. *Artificial Intelligence*, 35(2):165–195, 1988.

[Lif94] Vladimir Lifschitz. Circumscription. In J. A. Robinson Dov M. Gabbay, C. J. Hogger, editor, *Handbook of logic in artificial intelligence and logic programmin*, volume 3, pages 297–352. Oxford, 1994.

[Lin98] Fangzhen Lin. On the relationships between static and dynamic causal rules in the situation calculus. In Charles L. Ortiz, Jr., editor, *Working Notes of the AAAI Spring Symposium on Prospects for a Commonsense Theory of Causation*, pages 38–43, Menlo Park, CA, 1998. American Association for Artificial Intelligence.

[LR94] Fangzhen Lin and Ray Reiter. State constraints revisited. *Journal of Logic and Computation*, 4:655–678, 1994.

[LS95] Fangzhen Lin and Yoav Shoham. Provably correct theories of action. *Journal of the ACM*, 42(2):293–320, March 1995.

[MC98] John McCarthy and Tom Costello. Combining narratives. In *Proceedings of Sixth Intl. Conference on Principles of Knowledge Representation and Reasoning*, pages 48–59. Morgan-Kaufman, 1998.

[McC59] John McCarthy. Programs with Common Sense[8]. In *Mechanisation of Thought Processes, Proceedings of the Symposium of the National Physics Laboratory*, pages 77–84, London, U.K., 1959. Her Majesty's Stationery Office. Reprinted in [McC90].

[McC63] John McCarthy. Situations, actions and causal laws. Technical Report Memo 2, Stanford University Artificial Intelligence Laboratory, Stanford, CA, 1963. Reprinted in [Min68].

[McC86] John McCarthy. Applications of Circumscription to Formalizing Common Sense Knowledge[9]. *Artificial Intelligence*, 28:89–116, 1986. Reprinted in [McC90].

[McC90] John McCarthy. *Formalizing Common Sense: Papers by John McCarthy.* Ablex Publishing Corporation, 1990.

[McC92] John McCarthy. Overcoming unexpected obstacles[10]. Web only, 1992.

[McC95] John McCarthy. Situation Calculus with Concurrent Events and Narrative[11]. 1995. Web only, partly superseded by [MC98].

[McC99] John McCarthy. Elaboration tolerance[12]. *web only for now*, 1999.

[McI00] Sheila A. McIlraith. An axiomatic solution to the ramification problem (sometimes). *Artificial Intelligence*, 116(1–2):87–121, 2000.

[MH69] John McCarthy and Patrick J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence[13]. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969. Reprinted in [McC90].

[Min68] Marvin Minsky, editor. *Semantic information processing.* MIT Press, 1968.

[Pin98a] Javier A. Pinto. Concurrent actions and interacting effects. In Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, editors, *KR'98: Principles of Knowledge Representation and Reasoning*, pages 292–303. Morgan Kaufmann, San Francisco, California, 1998.

[8] http://www-formal.stanford.edu/jmc/mcc59.html

[9] http://www-formal.stanford.edu/jmc/applications.html

[10] http://www-formal.stanford.edu/jmc/glasgow.html

[11] http://www-formal.stanford.edu/jmc/narrative.html

[12] http://www-formal.stanford.edu/jmc/elaboration.html

[13] http://www-formal.stanford.edu/jmc/mcchay69.html

[Pin98b] Javier A. Pinto. Occurrences and narratives as constraints in the branching structure of the situation calculus. *Journal of Logic and Computation*, 8(6):777–808, 1998.

[Rei01] Raymond Reiter. *Knowledge in Action*. M.I.T. Press, 2001.

[RM94] R.S.Miller and M.P.Shanahan. Narratives in the situation calculus. *Journal of Logic and Computation*, 4(5):513–530, 1994.

[Sha97] Murray Shanahan. *Solving the Frame Problem, a mathematical investigation of the common sense law of inertia*. M.I.T. Press, 1997.

[Sho88] Yoav Shoham. Chronological ignorance: Experiments in nonmonotonic temporal reasoning. *Artificial Intelligence*, 36(3):279–331, 1988.