Notes on Deductive and Abductive Planning in the Event Calculus

Rob Miller
Department of Computing, Imperial College,
180 Queen's Gate, London SW7 2BZ, U.K.
rsm@doc.ic.ac.uk
http://www.doc.ic.ac.uk/~rsm/

July 1996

The main purpose of these notes is to show how, at least from a semantic or conceptual point of view, and with a few simple modifications, recent versions of the Event Calculus can support both a deductive and an abductive view of planning in linear time. Specifically, two potential criticisms of planning with the Event Calculus are addressed; (i) the criticism that it is *necessary* to use 'meta-level' concepts such as abduction to formulate the notion of a plan in the context of the Event Calculus, and (ii) the criticism that planning with the Event Calculus will involve some kind of (computationally intractable) dynamic consistency checking.

1 Domain Independent Event Calculus Axioms

The axioms below are written in sorted predicate calculus, with sorts for actions (variables $a, a_1, a_2, \ldots$), fluents ($f, f_1, f_2, \ldots$) and timepoints ($t, t_1, t_2, \ldots$). A sort of domain objects (variables $x, x_1, x_2, \ldots$) is also assumed with which to deal with "non-temporal" aspects of the domain. Only models are considered in which the sort of timepoints is interpreted as the non-negative reals or as the naturals, with the symbol '$<$' referring to the usual ordering. All variables are assumed to be universally quantified with maximum scope unless otherwise stated. Axioms (EC1)–(EC6) below express the following: (i) Fluents which initially hold, or which have been initiated by an occurrence of an action, continue to hold until an occurrence of an action which terminates them. (ii) Fluents which do not initially hold, or which have been terminated by an occurrence of an action, continue not to hold until an occurrence of an action which initiates them.

$$HoldsAt(f,t) \leftarrow [InitialisedTrue(f) \wedge \neg Clipped(0,f,t)] \qquad \text{(EC1)}$$

$$\neg HoldsAt(f,t) \leftarrow [InitialisedFalse(f) \wedge \neg Declipped(0,f,t)] \qquad \text{(EC2)}$$

$$HoldsAt(f,t_2) \leftarrow [Happens(a,t_1) \wedge Initiates(a,f,t_1) \wedge t_1 < t_2 \wedge \neg Clipped(t_1,f,t_2)] \qquad \text{(EC3)}$$

$$\neg HoldsAt(f,t_2) \leftarrow [Happens(a,t_1) \wedge Terminates(a,f,t_1) \wedge t_1 < t_2 \wedge \neg Declipped(t_1,f,t_2)] \qquad \text{(EC4)}$$

$$Clipped(t_1,f,t_2) \leftrightarrow [[InitialisedFalse(f) \wedge t_1 = 0] \vee \exists a,t[Happens(a,t) \wedge t_1 \leq t < t_2 \wedge [Terminates(a,f,t) \vee Impossible(a,t)]]] \qquad \text{(EC5)}$$

$$Declipped(t_1,f,t_2) \leftrightarrow [[InitialisedTrue(f) \wedge t_1 = 0] \vee \exists a,t[Happens(a,t) \wedge t_1 \leq t < t_2 \wedge [Initiates(a,f,t) \vee Impossible(a,t)]]] \qquad \text{(EC6)}$$

Note that, like Kartha and Lifschitz's work in [8] and Sandewall's use of *occlusion* [22], this axiomatisation includes mechanisms for dynamically adding fluents to (and removing fluents from) the "frame" (i.e. the set of fluents subject to the "commonsense law of inertia"). Similar ideas were first introduced into the event calculus by Shanahan in [24]. At time 0, exactly those fluents which have been *initialised true* or *initialised false* belong to the frame (exclusive "or"). Other fluents are dynamically added to the frame either by being *initiated* or by being *terminated* by action occurrences at specific times. Simultaneously *initiating* and *terminating* a fluent does not cause an inconsistency, but causes the fluent to be removed from the frame, so that at subsequent times it is not possible to conclude either that it holds or that it does not hold. Analogously, the occurrence of an "impossible" action doesn't cause an inconsistency, but removes all fluents from the frame[1].

For the present purposes, it is convenient to distinguish between two forms of action occurrence from the perspective of a particular agent – (i) occurrences either known to have happened, or predicted to happen using knowledge about the laws of nature and about the intentions of other agents, etc., and (ii) future performances of actions planned by the agent itself. Accordingly, *Happens* is defined in terms of the predicates *Occurs* and *Perform* as follows:

$$Happens(a,t) \leftrightarrow [Occurs(a,t) \vee Perform(a,t)] \qquad \text{(EC7)}$$

(This distinction between the predicates *Occurs* and *Perform* is somewhat analogous to the distinction between the predicates *Triggers* and *Performed* in [12].)

[1] Reiter [19] deals with impossible actions in an analogous way within the context of the Situation Calculus. Every sequence of actions is included as a situation term in the language, but the logical mechanisms for predicting what holds when are blocked in those situations resulting from an action which is not possible.

2 A Basic Notion of Planning

Regarding formulating a notion of planning and of a plan, the basic idea is as follows. A domain specific theory $\mathcal{T}$ is written using the predicates *Impossible*, *Initiates*, *Terminates*, *Occurs*, *InitialisedTrue*, etc., but not mentioning the predicates *Happens* or *Perform*. In the theory $\mathcal{T}$, the predicates *Impossible*, *Initiates*, *Terminates* and *Occurs* are appropriately minimised to solve the qualification, frame and occurrence minimisation problems. A *goal* $\mathcal{G}$ is a conjunction of ground *Holds* literals. A *plan from time* τ *for* $\mathcal{G}$ is a sentence $\mathcal{P}$ of the form

$$Perform(a,t) \;\leftrightarrow\; [[a = \alpha_1 \wedge t = \tau_1] \;\vee \ldots \vee [a = \alpha_n \wedge t = \tau_n]] \tag{1}$$

where each meta-variable α_i is a ground action term, each meta-variable τ_i is a time point constant such that $\tau_i \geq \tau$, and $\mathcal{P}$ is such that

$$(\text{EC1}) \wedge \ldots \wedge (\text{EC7}) \wedge \mathcal{T} \wedge \mathcal{P} \quad \models \quad \mathcal{G} \tag{2}$$

or equivalently

$$(\text{EC1}) \wedge \ldots \wedge (\text{EC7}) \wedge \mathcal{T} \quad \models \quad \mathcal{P} \rightarrow \mathcal{G} \tag{3}$$

Hence, under this formulation, planning can either be viewed as *abducing* sentences of the form $\mathcal{P}$ to add to the domain theory [4] [5] [14] [15] [16], or can be viewed as *deducing* theorems (of the form $\mathcal{P} \rightarrow \mathcal{G}$), in the spirit of Green, Reiter and others [7] [20] [21]. In either case, it is necessary to ensure that the plan is consistent with the domain theory, i.e. that

$$(\text{EC1}) \wedge \ldots \wedge (\text{EC7}) \wedge \mathcal{T} \quad \not\models \quad \neg\mathcal{P} \tag{4}$$

However, this need not necessarily place any computational burden on the mechanism for generating plans. Instead, theorems can be formulated stating that provided $\mathcal{T}$ is of a certain general form, *any* $\mathcal{P}$ of the form (1) satisfies (4), thus making the need for dynamic consistency checking redundant. The remainder of these notes concentrate on formulating and proving such a theorem for a simple but general class of domain theories.

3 Domain Dependent Axioms

We will consider domain descriptions of the form

$$\mathcal{D} = Poss(\mathcal{D}) \wedge Eff(\mathcal{D}) \wedge Init(\mathcal{D}) \wedge Nar(\mathcal{D}) \wedge Una(\mathcal{D}) \wedge Dom(\mathcal{D}),$$

where:

- $Poss(\mathcal{D})$ contains a finite number of axioms of the form

$$Impossible(a,t) \leftarrow \Phi(a,t)$$

where a, and t are the only variables which appear free in the formulae $\Phi(a,t)$, and where the only timepoint terms in $\Phi(a,t)$ are free occurrences of the variable t inside the predicate $HoldsAt$.

- $Eff(\mathcal{D})$ contains a finite number of axioms either of the form

  $$Initiates(a,f,t) \leftarrow \Phi(a,f,t)$$

  or of the form

  $$Terminates(a,f,t) \leftarrow \Phi(a,f,t)$$

  where a, f and t are the only variables which appear free in the formulae $\Phi(a,f,t)$, and where the only timepoint terms in $\Phi(a,f,t)$ are free occurrences of the variable t inside the predicate $HoldsAt$.

- $Init(\mathcal{D})$ contains a finite number of axioms either of the form

  $$InitialisedTrue(f) \leftarrow \Phi(f)$$

  or of the form

  $$InitialisedFalse(f) \leftarrow \Phi(f)$$

  where f is the only variable free in $\Phi(f)$, and $\Phi(f)$ contains no action or timepoint terms.

- $Nar(\mathcal{D})$ contains a finite number of axioms of the form

  $$Occurs(a,\tau) \leftarrow \Phi(a)$$

  for some non-negative real number τ, where a is the only variable which appears free in the formulae $\Phi(a)$, and $\Phi(a)$ does not contain any timepoint or fluent terms.

- $Una(\mathcal{D})$ is a set of uniqueness-of-names axioms for all fluent and action symbols.

- $Dom(\mathcal{D})$ is a theory containing no action, fluent or timepoint terms.

3.1 An Example

The following example domain theory $\mathcal{D}_d$ concerns a room with a door which is initially locked. $Poss(\mathcal{D}_d)$ consists of the single rule

$$Impossible(a,t) \leftarrow [a = Enter \wedge \neg HoldsAt(Open,t)]$$

$Eff(\mathcal{D}_d)$ Contains the four rules

$$Initiates(a,f,t) \leftarrow [a = Enter \wedge f = InRoom \wedge HoldsAt(Open,t)]$$

$$Initiates(a,f,t) \leftarrow [a = Push \wedge f = Open \wedge \neg HoldsAt(Locked,t)]$$

$$Terminates(a,f,t) \leftarrow [a = Pull \wedge f = Open]$$

$$Terminates(a,f,t) \leftarrow [a = TurnKey \wedge f = Locked]$$

$Init(\mathcal{D}_d)$ Contains the three rules

$$InitialisedTrue(f) \leftarrow f = Locked$$

$$InitialisedFalse(f) \leftarrow f = Open$$

$$InitialisedFalse(f) \leftarrow f = InRoom$$

$Una(\mathcal{D}_d)$ is a set of uniqueness-of-names axioms for the fluents $Open$, $Locked$ and $InRoom$, and for the actions $TurnKey$, $Enter$, $Push$ and $Pull$:

$$Open \neq Locked \wedge Open \neq InRoom \wedge Locked \neq InRoom$$

$$\begin{aligned} & TurnKey \neq Enter \wedge TurnKey \neq Push \wedge TurnKey \neq Pull \\ & \quad \wedge Enter \neq Pull \wedge Enter \neq Push \wedge Push \neq Pull \end{aligned}$$

$Nar(\mathcal{D}_d)$ and $Dom(\mathcal{D}_d)$ are empty.

Given the goal $\mathcal{G}_d = HoldsAt(InRoom, 14)$, we would intuitively expect the following two sentences to both be plans from time 0 for $\mathcal{G}$:

$$\begin{aligned} Perform(a,t) \leftrightarrow\ & [[a = TurnKey \wedge t = 3] \vee [a = Push \wedge t = 7] \\ & \vee [a = Enter \wedge t = 11]] \end{aligned} \qquad (\mathcal{P}_1)$$

$$\begin{aligned} Perform(a,t) \leftrightarrow\ & [[a = TurnKey \wedge t = 3] \vee [a = Pull \wedge t = 5] \\ & \vee [a = Push \wedge t = 8] \vee [a = Enter \wedge t = 12]] \end{aligned} \qquad (\mathcal{P}_2)$$

However, it is neither the case that

$$\text{(EC1)} \wedge \ldots \wedge \text{(EC7)} \wedge \mathcal{D}_d \models \mathcal{P}_1 \rightarrow \mathcal{G}_d$$

nor that

$$\text{(EC1)} \wedge \ldots \wedge \text{(EC7)} \wedge \mathcal{D}_d \models \mathcal{P}_2 \rightarrow \mathcal{G}_d$$

$\mathcal{D}_d$ needs to be transformed into a stronger theory in order to solve (i) the occurrence minimisation problem (we need for example to be able to infer $\neg Occurs(Pull, 10)$ and thus $\neg Happens(Pull, 10)$), (ii) the frame problem (for example, in the second plan $\mathcal{P}_2$, we need to be able to infer that the redundant $Pull$ action doesn't lock the door again, i.e. $\neg Initiates(Pull, Locked, t)$), and (iii) the qualification problem (we need to be able to infer sentences such as $\neg Impossible(TurnKey, 3)$). The purpose of the circumscription policy described in the next section is to provide such a transformation.

4 A Circumscription Policy

As mentioned previously, to deal with the qualification, frame and occurrence minimisation problems, the predicates *Impossible*, *Initiates*, *Terminates* and *Occurs* are minimised. The circumscription policy used here is inspired by a solution to the frame problem described by Kartha and Lifschitz in [9]. The circumscription operator is not applied globally – rather, simple circumscription policies are applied to those components of the domain theories which can be regarded as suppling the (partial) definitions of the particular predicates to be minimised. Given a domain theory $\mathcal{D}$ as described above, the circumscription policy is:

$$\begin{array}{l} CIRC[Nar(\mathcal{D})\ ;\ Occurs] \ \wedge\ CIRC[Eff(\mathcal{D})\ ; Initiates, Terminates] \\ \quad \wedge\ CIRC[Poss(\mathcal{D})\ ; Impossible] \ \wedge\ Init(\mathcal{D}) \ \wedge\ Una(\mathcal{D}) \ \wedge\ Dom(\mathcal{D}) \end{array}$$

We will abbreviate this to $CIRC_{EC}[\mathcal{D}]$. The use of circumscription need not be cause for concern. In fact, using results from [11] it is straightforward to show that, for the class of domain descriptions we are considering, the three circumscibed conjuncts of $CIRC_{EC}[\mathcal{D}]$ are each equivalent to simple (first order) completion statements, so that we could equally have used specialised completion operators instead. (The circumscription operator $CIRC[-\ ; Initiates, Terminates]$ is somewhat analogous to the mechanism Reiter employs to transform "effect" axioms into "successor state" axioms.)

5 A Theorem about Plan Consistency

Theorem 1 Let $\mathcal{D}$ be a domain description as described above, such that the sentence $\mathcal{D}^- = Init(\mathcal{D}) \wedge Una(\mathcal{D}) \wedge Dom(\mathcal{D})$ is consistent. Let $\mathcal{P}$ be a sentence of the form

$$Perform(a, t) \ \leftrightarrow\ [[a = \alpha_1 \wedge t = \tau_1] \ \vee \ldots \vee [a = \alpha_n \wedge t = \tau_n]] \tag{1}$$

where each meta-variable α_i is a ground action term and each meta-variable τ_i is a time point constant. Then

$$\mathcal{D}^+_{\mathcal{P}} \ =\ (\mathrm{EC1}) \wedge \ldots \wedge (\mathrm{EC7}) \wedge CIRC_{EC}[\mathcal{D}] \wedge \mathcal{P}$$

is consistent.

Proof: (Sketch) Given a model M of $\mathcal{D}^-$, a model M' of $\mathcal{D}^+_\mathcal{P}$ can be constructed from M with identical interpretations of all symbols except the predicates *HoldsAt*, *Initiates*, *Terminates*, *Clipped*, *Declipped*, *Impossible*, *Happens*, *Occurs* and *Perform*, (all of which have a temporal argument). The interpretation of *Perform* in M' can be straightforwardly constructed from $\mathcal{P}$, and similarly the interpretation of *Occurs* in M' can be constructed from the completion statement $CIRC[Nar(\mathcal{D})\ ;\ Occurs]$ in the obvious way. The interpretation of *Happens* can be defined as the union of the interpretations of *Perform* and *Occurs*, so that (EC7) is satisfied. Hence, because of the form of $\mathcal{P}$ and $CIRC[Nar(\mathcal{D})\ ;\ Occurs]$, there are only a finite number of points $\tau_1, \ldots, \tau_n$ in M' at which actions happen.

It is now straightforward to add interpretations to M' for *HoldsAt*, *Initiates*, *Terminates*, *Clipped*, *Declipped* and *Impossible* which satisfy $CIRC_{EC}[\mathcal{D}]$ and (EC1)–(EC6), by working along each of the intervals $[0, \tau_1], (\tau_1, \tau_2], \ldots$, etc. in turn. For example, if $\tau \in [0, \tau_1]$, then $\langle \phi, \tau \rangle$ is in the interpretation of *HoldsAt* if and only if ϕ is in the interpretation of *InitialisedTrue* and not in the interpretation of *InitialisedFalse*. Given the interpretation of *HoldsAt* over such an interval $(\tau_i, \tau_{i+1}]$, the interpretations of *Initiates*, *Terminates* and *Impossible* are defined in the obvious way using the completion statements $CIRC[Eff(\mathcal{D})\ ;\ Initiates, Terminates]$ and $CIRC[Poss(\mathcal{D})\ ;\ Impossible]$ respectively. Finally, the interpretations of *Clipped* and *Declipped* for any pair of timepoints between 0 and τ_{i+1} are defined straightforwardly using (EC5) and (EC6). •

If time is interpreted as the naturals, the above theorem can easily be extended to cover domains with "triggered" events, i.e. to theories $\mathcal{D}$ such that $Nar(\mathcal{D})$ contains rules of the form

$$Occurs(\alpha, t) \leftarrow [\{\neg\}HoldsAt(\phi_1, t) \wedge \ldots \wedge \{\neg\}HoldsAt(\phi_n, t)]$$

The proof is analogous to that of Theorem 1, but in this case the model is built by counting along the time line, rather than working along the series of intervals $[0, \tau_1], (\tau_1, \tau_2], \ldots$. Again where time is interpreted only as the naturals, the theorem can be proved with respect to an expanded Event Calculus which includes "trajectory" mechanisms similar to those introduced by Shanahan [23], [24], [25]. (These mechanisms facilitate the description of the behaviours of parameters whose values can change between action occurrences according to mathematical functions of time.)

If we allow time to also be interpreted as the reals, things get more complicated. Investigations are currently underway to find syntactically or mathematically verifiable constraints on domain descriptions which guarantee plan consistency in the sense of Theorem 1, where the associated Event Calculus includes mechanisms for dealing with continuous change similar to those in [23], [24] or [13].

6 Discussion

These notes were written partly as a response to the remarks by Reiter in [20], about abduction and planning in linear time formalisms. In particular, the splitting of the *Happens* predicate into the predicates *Occurs* and *Perform* (see axiom (EC7)) is in order to address what Reiter calls the "premature minimisation problem". Although the present formalism still embodies the assumption that all "natural" action occurrences are inferrable from the axiomatization (the predicate *Occurs* is minimised), the agent retains "free will" as regards its own future actions (i.e. can pick its own closure axiom $\mathcal{P}$ for *Perform*). This allows for both a deductive and an abductive view of planning (the expressions (2) and (3) are equivalent).

Reiter lists five specific drawbacks of abductive planning. The first three are:

> *1. It is a metalevel task; the planner must leave the object language to generate a candidate collection of atoms of the form occurs*(A, T)*, test the consistency of these atoms with respect to the object level axiomatization of the domain, then return to the object level to prove the goal sentence relative to the enlarged axiom set.*
>
> *2. Because of the above consistency test, abductive plans are not even recursively enumerable for first order axiomatizations, in contrast to the deductive case.*
>
> *3. Even if we ignore the noncomputability of the consistency test, from a computational point of view there are at least two theorem proving tasks for abductive planners: the consistency test (which normally must be performed several times), together with the goal-entailment proof.*

As regards the need for consistency checking, Theorem 1 and analogous results can be used to eliminate these drawbacks. As regards metalevel reasoning, if an object-level formulation of planning is preferred, expression (3) rather than expression (2) may be used in the definition of a plan.

Reiter's fourth drawback is:

> *4. It is not difficult to imagine settings where a robot agent needs to establish that a plan does* not *exist. In the deductive case, this amounts to establishing that Axioms* $\models \neg(\exists s)G(s)$*, and, at least formally, is no more problematic than the planning problem. For abductive planners, it is not at all obvious what such a proof might look like, or how it could be constructed; the robot must show, again at the metalevel, that there is no finite set of atoms of the form occurs*(A, T)*, consistent with the background axioms, which entails the goal.*

By monotonicity of classical logic, in the framework presented in these notes it would certainly be sufficient to establish

$$\text{(EC1)} \wedge \ldots \wedge \text{(EC7)} \wedge \mathcal{T} \quad \models \quad \neg\mathcal{G}$$

in order to show that a plan does not exist for a goal $\mathcal{G}$. Again, formally this is no more problematic than the planning problem. However, in practice, one would probably have to settle for deducing something more like

$$\neg\exists a,t[t < \tau \ \wedge \ \mathit{Happens}(a,t) \ \wedge \ \mathit{Impossible}(a,t)] \ \rightarrow \ \neg\mathcal{G}(\tau)$$

which could be regarded as roughly analogous to

$$\neg(\exists s)(\mathit{possible}(s) \wedge \mathit{legal}(s) \wedge G(s))$$

in the Situation Calculus.

Reiter's last drawback concerns concurrency:

> *5. As observed by Pelavin [17], in the case of concurrent actions, abductive planning can yield incorrect plans in the presence of partial world descriptions.*

It is beyond the scope of these notes to investigate representing the effects of (possibly conflicting or combining) concurrent actions in detail. But notice that axioms (EC1)–(EC7) offer at least a partial solution to the "precondition interaction problem" [18], [20]. For example, if a domain description includes both the information $\mathit{Terminates}(\mathit{StartMoveRight}, \mathit{MovingLeft}, t)$ and the information $\mathit{Initiates}(\mathit{StartMoveLeft}, \mathit{MovingLeft}, t)$, performing both *StartMoveRight* and *StartMoveLeft* simultaneously doesn't cause an inconsistency, but simply makes the subsequent truth value of *MovingLeft* impossible to predict. In this respect the framework here is similar to and inspired by the *Language* $\mathcal{A}$–based framework of Bornscheuer and Thielscher [3], who argue that it is often possible to safeguard against potential sources of inconsistency within a domain dependent theory by introducing sufficient scope for non-determinism within the surrounding domain independent framework.

The general issue of consistency verification has to be addressed, one way or another, both by proponents of the Event Calculus and of the Situation Calculus (and indeed of any other logic based formalism). For either formalism to be taken seriously as a methodology for modelling and reasoning about reasonably complex dynamic domains, the potential user will have to be supplied with guidelines and support for ensuring that his or her domain dependent theory is consistent with the given framework. Theorem 1 is intended as a step towards this goal. Experience shows that the same issues tend to crop up in different guises in both the Event and the Situation Calculi. The need to verify that all potential plans are consistent with (other aspects of) the domain description is no exception. Situation Calculus approaches potentially suffer from what might be called the "premature projection problem". That is, all conceivable "plans" are already incorporated (as situation terms) in the domain description as a branch in the tree of situations, so that an inconsistency along any one branch is propagated throughout the theory, even if the associated "plan" can never be or is never put into action. Hence theorems are required to show that, given a domain description of a particular general class, inconsistency can never arise for any situation term – whether "possible", "impossible", "legal", "illegal", "hypothetical", "actual" or whatever. It seems likely that such theorems could have proofs analogous to that of Theorem 1, and, for each general class of domain, proofs would be roughly of the same difficulty or complexity. This is because intuitively a Situation Calculus domain description can be thought of as a causal theory plus all conceivable world histories, whereas an Event Calculus domain description together with an arbitrary plan (in the sense of these notes) can be thought of as a causal theory plus an arbitrary single world history.

The idea of a "plan from time τ for $\mathcal{G}$" is perhaps more general than necessary – in practice one would probably be interested only in plans from time "*Now*", where the interpretation of distinguished time point symbol *Now* was appropriately updated as time progressed. One could envisage an agent architecture similar to that in [10], [1] or [2] (note the use of the symbol "s_N" to represent the "situation now" in [1] and [2]), with *Occurs* facts appropriately asserted into the agent's knowledge base as it executed various actions (a minor extension to Theorem 1 would guarantee that such assertions could not result in inconsistency). In the present formulation, adding the constraint $\neg\exists t,a[t < Now \ \wedge \ Perform(a,t)]$ to the domain independent theory would effectively divide time into a known or linear past and an unknown or branching future. Some of these ideas are a little reminiscent of Gabbay's work [6] on "declarative past and imperative future".

References

[1] Chitta Baral, Michael Gelfond and Alessandro Provetti, *Representing Actions - I: Laws, Observations and Hypotheses (Short Version)*, in Working Notes of the AAAI Spring Symposium: Extending Theories of Action - Formal Theory and Practical Applications, Stanford University, California, USA, 1995.

[2] Chitta Baral, Michael Gelfond and Alessandro Provetti, *Representing Actions: Laws, Observations and Hypotheses*, Journal of Logic Programming, Special Issue on Reasoning about Actions (to appear), 1996.

[3] Sven-Erik Bornscheuer and Michael Thielscher, *Explicit and Implicit Indeterminism: Reasoning about Uncertain and Contradictory Specifications of Dynamic Systems*, `http://www.icsi.berkeley.edu/~michaelt/publications/reports/ordered-index.html`, Technical Report TR-96-009, ICSI, 1996.

[4] Marc Denecker, Lode Missiaen and Maurice Bruynooghe, *Temporal Reasoning with Abductive Event Calculus*, in proceedings of ECAI'92, 1992

[5] Kave Eshghi, *Abductive Planning with Event Calculus*, Proceedings of the 5th International Conference and Symposium on Logic Programming, ed.s Robert Kowalski and Kenneth Bowen, MIT Press, pages 562–579, 1988.

[6] Dov Gabbay, *The Declarative Past and Imperative Future*, Proceedings of the Temporal Logic in Specification: Altrincham Workshop 1987, LNCS 398, pages 409-448, Springer-Verlag, 1989.

[7] C.C. Green, *Theorem Proving by Resolution as a Basis for Question-answering Systems*, in B. Meltzer and D. Michie, ed.s, Machine Intelligence 4, Pub. American Elsevier, New York, pages 183-205, 1969.

[8] G. Neelakantan Kartha and Vladimir Lifschitz, *Actions with Indirect Effects (Preliminary Report)*, Proceedings KR'94, pages 341-350, 1994.

[9] G. Neelakantan Kartha and Vladimir Lifschitz, *A Simple Formalization of Actions Using Circumscription*, In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95), pages 1970-1975, 1995.

[10] Robert Kowalski, *Using Metalogic to Reconcile Reactive with Rational Agents*, in K. Apt and F. Turini, ed.s, Meta-logics and Logic Programming, MIT Press, 1995.

[11] Vladimir Lifschitz, *Circumscription*, in Handbook of Logic in Artificial Intelligence, ed.s D. Gabbay, C. Hogger and J.A. Robinson, Oxford University Press, pages 297-352, 1995.

[12] Rob Miller, *A Case Study in Reasoning about Actions and Continuous Change*, in Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96), Budapest, Hungary, Pub. John Wiley & Sons, Ltd., 1996.

[13] Rob Miller and Murray Shanahan, *Reasoning about Discontinuities in the Event Calculus*, to appear in proceedings of KR'96, 1996.

[14] Lode Missiaen, *Localized Abductive Planning for Robot Assembly*, in proceedings of the 1991 IEEE Conference on Robotics and Automation, IEEE Robotics and Automation Society, pages 605-610, 1991.

[15] Lode Missiaen, *Localized Abductive Planning with the Event Calculus*, Ph.D. Thesis, Department of Computer Science, K.U.Leuven, 1991.

[16] Lode Missiaen, Marc Denecker and Maurice Bruynooghe, *An Abductive Planning System based on Event Calculus*, Journal of Logic and Computation, vol. 5 no. 5. pages 579-602, 1995.

[17] R.N. Pelavin, *Planning with Simultaneous Actions and External Events*, in J.F. Allen, H.A. Kautz, R.N. Pelavin and J.D. Tenenberg (ed.s), Reasoning about Plans, pages 127-211, Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[18] Javier Pinto, *Temporal Reasoning in the Situation Calculus*, PhD. Thesis, University of Toronto, 1994.

[19] Ray Reiter, *The Frame Problem in the Situation Calculus: a Simple Solution (Sometimes) and a Completeness Result for Goal Regression*, in Artificial Intelligence and Mathematical Theory of Computation: Papers in Honour of John McCarthy, ed. V. Lifschitz, Academic Press, page 418, 1991.

[20] Ray Reiter, *Natural Actions, Concurrency and Continuous Time in the Situation Calculus*, `http://www.cs.toronto.edu/~cogrobo/`, 1996. (Previous version in proceedings of Common Sense'96, page 176, 1996.)

[21] Ray Reiter, *A Formal Account of Planning with Concurrency, Continuous Time and Natural Actions (Extended Abstract)*, Proceedings of the Workshop at FAPR'96, Reasoning about Actions and Planning in Complex Environments, Technical Report AIDA-96-11, Darmstadt, June 1996.

[22] Erik Sandewall, *Features and Fluents*, Oxford University Press, 1994.

[23] Murray Shanahan, *Representing Continuous Change in the Event Calculus*, In proceedings ECAI'90, pages 598-603, 1990.

[24] Murray Shanahan, *A Circumscriptive Calculus of Events*, Artificial Intelligence, vol 77, Elsevier Science Publishers, pages 249-284, 1995.

[25] Murray Shanahan, *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*, MIT Press, 1996 (to appear).