Reasoning about Actions, Events and Causality*

Antonis Kakas
Department of Computer Science,
University of Cyprus,
75 Kallipoleos Street, P.O. Box 537,
CY-1678 Nicosia, CYPRUS
antonis@turing.cs.ucy.ac.cy
http://www.ucy.ac.cy/ucy/cs/webPages/PEOPLE/faculty/kakas.htm

Rob Miller
Department of Computer Science,
Queen Mary & Westfield College,
University of London,
Mile End Road, London E1 4NS, U.K.
rsm@dcs.qmw.ac.uk
http://www.dcs.qmw.ac.uk/~rsm/

Abstract

The Language $\mathcal{E}$ is a simple declarative language for describing the effects of action occurrences within a given narrative, using an ontology of actions, time points and fluents (i.e. properties which can change their truth values over time). This paper shows how $\mathcal{E}$ may be extended to deal with ramifications. More precisely, we show how Language $\mathcal{E}$ domain descriptions can include statements describing permanent relationships or constraints between fluents, and how the model theoretic semantics of $\mathcal{E}$ can be extended in an intuitive way to ensure that the effects of actions are appropriately propagated via such statements, whilst retaining $\mathcal{E}$'s simple approach to the frame problem. This results in a simple model of causality appropriate for many domains involving actions and change.

Introduction

The idea of *action description languages* was first introduced by Gelfond and Lifschitz [6, 7], with their proposal for the Language $\mathcal{A}$. The intention was that such languages would have a specialised syntax and semantics, and would be at a high enough level of abstraction so as to be easily readable, understandable and intuitive. They could then be used as specifications for theories of action and change written in more general purpose or computation oriented formalisms. Since then, many variants and extensions of $\mathcal{A}$ have been proposed. The majority of these, like $\mathcal{A}$ itself, are based on an ontology inherited from the Situation Calculus.

The Language $\mathcal{E}$ [9] was introduced partly to illustrate that Gelfond and Lifschitz's methodology can be applied using ontologies other than that of the Situation Calculus. In particular, it showed that for *narrative reasoning* (that is, reasoning about actions which actually occur at various times, and reasoning about the properties that hold or do not hold at different times as a consequence of such occurrences), it is appropriate to use an ontology where the flow of time is independent of the action occurrences embedded within it. $\mathcal{E}$'s ontology of actions, time points and fluents facilitates a particularly simple and intuitive syntax and semantics for describing the effects of such occurrences. Our approach has been inspired by Kowalski and Sergot's Event Calculus [13] and its extensions (see [27] for a comprehensive overview), in much the same way that $\mathcal{A}$ was inspired by the Situation Calculus. $\mathcal{E}$ might therefore be characterised as the first of a family of *event description languages*.

A key characteristic of the $\mathcal{E}$'s semantics is that it is *modular*, and thus easily extendible. To illustrate what is meant by this more precisely, two extensions of the "core" semantics are briefly described in an appendix of [9], to deal with *qualifications* and *ramifications* respectively. The purpose of the present paper, which summarises the first part of [10], is to expand on the theme of ramifications and causality. We argue that the inclusion of explicit time and events (action occurrences) in our ontology enables a simple and intuitive characterisation of cause and effect for a wide class of domains. Central to our approach are the semantic notions of an *initation point* and a *termination point*, which are timepoints at which particular events take place.

For generality, the semantics of $\mathcal{E}$ described in this paper (and in [9] and [10]) is in terms of an arbitrary structure of time. All definitions and results apply whether time is linear, branching, continuous or discrete. Indeed, as shown in [9], as a special case time may be defined as a forward branching structure of situations, each characterised by a unique sequence of actions, as in various versions of the situation calculus. In this case, as shown in [9], the semantics of $\mathcal{E}$ is equivalent to that of $\mathcal{A}$ (and thus, by the results of Kartha [11], equivalent to the core Situation Calculi of Reiter [23], Pednault [22] and Baker [1]). Hence both the representational and the computational mechanisms reported here and in [10] to deal with ramifications are, like those in [9], readily adaptable for use

* This paper is a summary of the first part of [10].

in the context of the Language $\mathcal{A}$ and the Situation Calculus.

A Review of the Basic Language $\mathcal{E}$

Syntax

Like the Language $\mathcal{A}$, the Language $\mathcal{E}$ is really a collection of languages. The particular vocabulary of each language depends on the domain being represented, but always includes a set of *fluent constants*, a set of *action constants*, and a partially ordered set of *time-points*. A *fluent literal* may either be a fluent constant or its negation, as shown in the following definitions.

Definition 1 [Domain Language] A *domain language* is a tuple $\langle \Pi, \preceq, \Delta, \Phi \rangle$, where $\preceq$ is a partial (possibly total) ordering defined over the non-empty set Π of *time points*, Δ is a non-empty set of *action constants*, and Φ is a non-empty set of *fluent constants*. □

Definition 2 [Fluent literal] A *fluent literal* of $\mathcal{E}$ is an expression either of the form F or of the form $\neg F$, where $F \in \Phi$. □

As in [9], for the remainder of this paper we assume a particular domain language $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$. We will often write $T_1 \prec T_2$ to mean $T_1 \preceq T_2$ and $T_1 \neq T_2$. Three types of statements are used to describe domains without ramifications; *h-propositions* ("h" for "happens"), *t-propositions* ("t" for "time point") and *c-propositions* ("c" for "causes"). Their intended meanings are clear from their definitions:

Definition 3 [h-proposition] An *h-proposition* in $\mathcal{E}$ is an expression of the form

$$A \textbf{ happens-at } T$$

where $A \in \Delta$ and $T \in \Pi$. □

Definition 4 [t-proposition] A *t-proposition* in $\mathcal{E}$ is an expression of the form

$$L \textbf{ holds-at } T$$

where L is a fluent literal of $\mathcal{E}$ and $T \in \Pi$. □

Definition 5 [c-proposition] A *c-proposition* in $\mathcal{E}$ is an expression either of the form

$$A \textbf{ initiates } F \textbf{ when } C$$

or of the form

$$A \textbf{ terminates } F \textbf{ when } C$$

where $F \in \Phi$, $A \in \Delta$, and C is a set of fluent literals of $\mathcal{E}$. □

"A **initiates** F **when** C" should be regarded as meaning "C is a minimally sufficient set of conditions for an occurrence of A to have an initiating effect on F". C-propositions of the form "A **initiates** F **when** $\emptyset$" and "A **terminates** F **when** $\emptyset$" can be written more simply as "A **initiates** F" and "A **terminates** F" respectively. For domains without indirect effects, a *domain description* in $\mathcal{E}$ is a triple $\langle \gamma, \eta, \tau \rangle$, where γ is a set of c-propositions, η is a set of h-propositions and τ is a set of t-propositions.

As a simple example domain, consider taking a photograph. Suppose that we need to refer to the actions of loading the camera with film, looking at the view and taking a picture, so that $\Delta = \{Load, Look, Take\}$, and refer to the properties of the camera having film in it and a picture having been taken, so that $\Phi = \{Loaded, Picture\}$. For Π and $\preceq$ we will pick the real numbers with the usual ordering relation. Given this vocabulary, an example of a t-proposition is

$$\neg Picture \textbf{ holds-at } 1 \qquad \text{(tsp1)}$$

Examples of h-propositions are

$$Load \textbf{ happens-at } 2 \qquad \text{(tsp2)}$$
$$Look \textbf{ happens-at } 5 \qquad \text{(tsp3)}$$
$$Take \textbf{ happens-at } 8 \qquad \text{(tsp4)}$$

and examples of c-propositions are

$$Load \textbf{ initiates } Loaded \qquad \text{(tsp5)}$$
$$Take \textbf{ initiates } Picture \textbf{ when } \{Loaded\} \qquad \text{(tsp6)}$$

Semantics

The semantics of $\mathcal{E}$ is based on simple definitions of interpretations and models. Since we are primarily interested in inferences about what holds at particular time-points in Π, it is sufficient to define an interpretation as a mapping of fluent/time-point pairs to $true$ or $false$ (i.e. a "*holds*" relation). An interpretation *satisfies* a fluent literal or set of fluent literals at a particular time-point if it assigns the relevant truth values to each of the corresponding fluent constants:

Definition 6 [Interpretation] An *interpretation* of $\mathcal{E}$ is a mapping

$$H : \Phi \times \Pi \mapsto \{true, false\}$$

□

Definition 7 [Point satisfaction] Given a set of fluent literals C of $\mathcal{E}$ and a time point $T \in \Pi$, an interpretation H *satisfies* C *at* T iff for each fluent constant $F \in C$, $H(F,T) = true$, and for each fluent constant F' such that $\neg F' \in C$, $H(F',T) = false$. □

The definition of a model in $\mathcal{E}$ is parametric on the definitions of an *initiation point* and a *termination point*. For the basic $\mathcal{E}$ (without ramification statements), these definitions are particularly straightforward. Initiation and termination points are simply time-points where a c-proposition and an h-proposition combine to describe a direct effect on a particular fluent:

Definition 8 [Initiation/termination point (without ramifications)] Let H be an interpretation of $\mathcal{E}$, let $D = \langle \gamma, \eta, \tau \rangle$ be a domain description, let $F \in \Phi$ and let $T \in \Pi$. T is an *initiation-point* (respectively *termination-point*) *for* F *in* H *relative to* D iff there is an $A \in \Delta$ such that (i) there is both an h-proposition in η of the form "A **happens-at** T" and a c-proposition in γ of the form "A **initiates** F **when** C" (respectively "A **terminates** F **when** C") and (ii) H satisfies C at T. □

Having defined these preliminary notions, it is straightforward to describe which interpretations are models of a particular domain description. Three basic properties need to be satisfied; (1) fluents change their truth values only via occurrences of initiating or terminating actions, (2) initiating a fluent establishes its truth value as $true$, and (3) terminating a fluent establishes its truth value as $false$. In addition, (4) every model must match with each of the t-propositions in the domain description:

Definition 9 [Model (without ramifications)] Given a domain description $D = \langle \gamma, \eta, \tau \rangle$ in $\mathcal{E}$, an interpretation H of $\mathcal{E}$ is a *model* of D iff, for every $F \in \Phi$ and $T, T', T_1, T_3 \in \Pi$ such that $T_1 \prec T_3$, the following properties hold:

1. If there is no initiation-point or termination-point T_2 for F in H relative to D such that $T_1 \preceq T_2 \prec T_3$, then $H(F, T_1) = H(F, T_3)$.
2. If T_1 is an initiation-point for F in H relative to D, and there is no termination-point T_2 for F in H relative to D such that $T_1 \prec T_2 \prec T_3$, then $H(F, T_3) = true$.
3. If T_1 is a termination-point for F in H relative to D, and there is no initiation-point T_2 for F in H relative to D such that $T_1 \prec T_2 \prec T_3$, then $H(F, T_3) = false$.
4. For all t-propositions in τ of the form "F **holds-at** T", $H(F, T) = true$, and for all t-propositions of the form "$\neg F$ **holds-at** T'", $H(F, T') = false$.

□

Definition 10 [Consistency] A domain description is *consistent* iff it has a model. □

Definition 11 [Entailment] A domain description D *entails* the t-proposition "F **holds-at** T", written "$D \models F$ **holds-at** T", iff for every model H of D, $H(F, T) = true$. D entails the t-proposition "$\neg F$ **holds-at** T" iff for every model H of D, $H(F, T) = false$. □

It is easy to see from the above definitions that, for example, the collection of t-, h- and c-propositions (tsp1)–(tsp6) entails the t-proposition

$$Picture \textbf{ holds-at } 10$$

Hence the Language $\mathcal{E}$ solves this (non-violent) version of the Yale Shooting Problem.

The framework described above allows for various modes of reasoning within various structures of time. As another example, we can construct a set Π_Δ of time-points out of the set $\Delta = \{Load, Look, Take\}$ of action constants used above. Π_Δ includes all sequences of these constants, e.g. $[Load, Look, Take]$ and $[Load, Load]$. We denote the empty sequence by S_0. It also includes auxiliary "start" points, $Start([Load, Look, Take])$, $Start([Load, Load])$, etc, for each such sequence other than S_0. The partial ordering $\preceq_\Delta$ on this set is the obvious one. Given any sequence $[\alpha_1, \ldots, \alpha_m]$, where each $\alpha_i \in \Delta$, it is such that

$$S_0 \prec_\Delta Start([\alpha_1, \ldots, \alpha_{m-1}]) \prec_\Delta$$
$$[\alpha_1, \ldots, \alpha_{m-1}] \prec_\Delta Start([\alpha_1, \ldots, \alpha_m])$$

Clearly, this gives a branching structure of time analogous to that used in many versions of the Situation Calculus. Now consider the domain description which includes the set η_Δ of all h-propositions of the form

$$\alpha_n \textbf{ happens-at } Start([\alpha_1, \ldots, \alpha_n])$$

and, in addition, the following c-propositions and t-propositions

$$Take \textbf{ initiates } Picture \textbf{ when } \{Loaded\}$$
$$\neg Picture \textbf{ holds-at } S_0$$
$$Picture \textbf{ holds-at } [Look, Take]$$

Again, it is easy to see that this domain entails the t-proposition

$$Loaded \textbf{ holds-at } S_0$$

Thus the semantics of $\mathcal{E}$ facilitates reasoning both forwards and backwards within various (branching or linear) structures of time. A general result in [9] shows that for domain descriptions with time structures such as $\langle \Pi_\Delta, \preceq_\Delta \rangle$ above and the set of h-propositions η_Δ, $\mathcal{E}$'s semantics is equivalent to that of the Language $\mathcal{A}$, so that $\mathcal{A}$ can be regarded as a special case of $\mathcal{E}$.[1] Note that proponents of the Situation Calculus generally regard reasoning with this type of branching structure of time as "hypothetical" reasoning about (possible future) actions (as opposed to reasoning about

[1] A slightly different notation is used in [9] to refer to the time-points within Π_Δ.

the "actual" time-line). Hence, viewed in this light, the h-propositions in η_Δ refer to hypothetical or possible future action occurrences. (This is in contrast to the very different use of the syntactically similar "occurrence facts" in [3], which refer to "actual" action occurrences.)

More generally, [9] also discusses how $\mathcal{E}$ can be used to formulate an intuitive notion of *planning*, and how various notions of *explanation* can be defined to deal with inconsistency within $\mathcal{E}$ domain descriptions. It also describes how Event Calculus style logic programs can be used to compute consequences (t-propositions) of $\mathcal{E}$ domain descriptions of the basic type described above, even in the absence of complete information about which fluents hold at some initial time-point.

Ramifications in $\mathcal{E}$

The *ramification problem* arises in domains whose description most naturally includes permanent constraints or relationships between fluents. In formalisms which allow for such statements, the effects of actions may sometimes be propagated via groups of these constraints. The problem is to adequately describe these propagations of effects, whilst retaining a solution to the frame problem – that is, the problem of succinctly expressing that most actions leave most fluents unchanged.

As briefly described in the appendix of [9], $\mathcal{E}$ domain descriptions can include ramification statements, called *r-propositions*, to describe constraints between fluents:

Definition 12 [r-proposition] An *r-proposition* in $\mathcal{E}$ is an expression of the form

$$L \textbf{ whenever } C$$

where L is a fluent literal and C is a set of fluent literals of $\mathcal{E}$. □

For the remainder of the paper, we will assume that domain descriptions are of the form $\langle \gamma, \eta, \tau, \rho \rangle$, where γ, η and τ are as before, and ρ is a set of r-propositions. The intended meaning of "L **whenever** C" is "C is a minimally sufficient cause for L", so that at every time-point that C holds, L holds, and every action occurrence (i.e. event) that brings about C also brings about L.

As a simple illustration of the use of r-propositions, consider the canonical "stuffy room" example. Imagine a room with a window and a vent. We can open and close both the window and the vent, with the obvious effects:

$$\begin{array}{ll} \mathit{CloseWindow} \textbf{ initiates } \mathit{WindowClosed} & \text{(sr1)} \\ \mathit{CloseVent} \textbf{ initiates } \mathit{VentClosed} & \text{(sr2)} \\ \mathit{OpenWindow} \textbf{ terminates } \mathit{WindowClosed} & \text{(sr3)} \\ \mathit{OpenVent} \textbf{ terminates } \mathit{VentClosed} & \text{(sr4)} \end{array}$$

Now suppose that we also want to assert that the room becomes stuffy whenever both the vent and the window are closed. We could of course attempt to describe this fact using extra c-propositions:

$$\begin{array}{c} \mathit{CloseWindow} \textbf{ initiates } \mathit{Stuffy} \textbf{ when } \{\mathit{VentClosed}\} \\ \mathit{CloseVent} \textbf{ initiates } \mathit{Stuffy} \textbf{ when } \{\mathit{WindowClosed}\} \\ \mathit{OpenWindow} \textbf{ terminates } \mathit{Stuffy} \\ \mathit{OpenVent} \textbf{ terminates } \mathit{Stuffy} \end{array}$$

However, there are problems with this approach. First, given the semantics of the previous section, these c-propositions fail to capture the "static" aspect of the intended constraint. In a domain description with no h-propositions or t-propositions at all, it would be possible to construct a model where, for example, *WindowClosed* and *VentClosed* were true at all time-points, but *Stuffy* was false. Second, this representation would fail to deal with simultaneous action occurrences. If both the vent and the window were initially open, closing both of them simultaneously would fail to make the room stuffy. If the vent was initially closed and the window initially open, simultaneously opening the vent and closing the window would cause an inconsistency.

The alternative is to use three r-propositions:

$$\begin{array}{ll} \mathit{Stuffy} \textbf{ whenever} & \text{(sr5)} \\ \qquad \{\mathit{VentClosed}, \mathit{WindowClosed}\} & \\ \neg\mathit{Stuffy} \textbf{ whenever } \{\neg\mathit{VentClosed}\} & \text{(sr6)} \\ \neg\mathit{Stuffy} \textbf{ whenever } \{\neg\mathit{WindowClosed}\} & \text{(sr7)} \end{array}$$

and to extend the semantics of $\mathcal{E}$ appropriately. As we shall see, the extension described below deals with the two problems described above, as well as with other problems raised by different examples. It is also easy to see that this approach has the additional advantage of being more succinct – imagine a slightly larger example where the room has a small window, a large window, a vent and a door, or where a number of different actions have the same effect of closing the vent.

In fact, we need make only very few modifications to the semantics of $\mathcal{E}$ in order to deal with r-propositions. The first modification is to extend condition (4) in Definition 9 of a model in the obvious way to cover the static aspect of r-propositions. Conditions (1), (2) and (3) stay the same (since they are parametric on the definitions of an initiation- and a termination-point, which are extended below), so that the new definition is:

Definition 13 [Model] Given a domain description $D = \langle \gamma, \eta, \tau, \rho \rangle$ in $\mathcal{E}$, an interpretation H of $\mathcal{E}$ is a *model* of D iff, for every $F \in \Phi$ and $T, T', T_1, T_3 \in \Pi$ such that $T_1 \prec T_3$, the following properties hold:

1. If there is no initiation-point or termination-point T_2 for F in H relative to D such that $T_1 \preceq T_2 \prec T_3$, then $H(F, T_1) = H(F, T_3)$.

2. If T_1 is an initiation-point for F in H relative to D, and there is no termination-point T_2 for F in H relative to D such that $T_1 \prec T_2 \prec T_3$, then $H(F, T_3) = true$.

3. If T_1 is a termination-point for F in H relative to D, and there is no initiation-point T_2 for F in H relative to D such that $T_1 \prec T_2 \prec T_3$, then $H(F, T_3) = false$.

4. For all t-propositions in τ of the form "F **holds-at** T", $H(F, T) = true$, for all t-propositions of the form "$\neg F$ **holds-at** T'", $H(F, T') = false$, and for all r-propositions in ρ of the form "L **whenever** C", if H satisfies C at T then H satisfies $\{L\}$ at T.

□

Using this new definition, and given an arbitrary $T \in \Pi$, we can, for example, now infer the t-proposition

$\neg WindowClosed$ **holds-at** T

from the two t-propositions

$\neg Stuffy$ **holds-at** T
$VentClosed$ **holds-at** T

by considering the "contrapositive", static effect of the r-proposition (sr5).

It remains only to extend the definitions of an initiation point and a termination point to accommodate the possibility of fluents being indirectly initiated and terminated via r-propositions. Again the modification is straightforward. Intuitively, in order to find time-points at which the fluent literal L is established via the r-proposition "L **whenever** C", we need to look for time-points at which one or more of the conditions in C become established, and at which the remaining conditions are already and continue to be satisfied (up to some time-point beyond the point in question). This intuition translates directly into the following fixed-point definition:

Definition 14 [Initiation/termination point] Let H be an interpretation of $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, and $D = \langle \gamma, \eta, \tau, \rho \rangle$ be a domain description. Let $\mathcal{W}$ be the set $2^{\Phi \times \Pi} \times 2^{\Phi \times \Pi}$ and let the operator $\mathcal{F} : \mathcal{W} \mapsto \mathcal{W}$ be defined as follows. For each, $\langle \mathcal{In}, \mathcal{Te} \rangle \in \mathcal{W}$ denote $\mathcal{F}(\langle \mathcal{In}, \mathcal{Te} \rangle)$ by $\langle \mathcal{In}', \mathcal{Te}' \rangle$. Then for any $F \in \Phi$ and $T \in \Pi$, (F, T) is in $\mathcal{In}'$ (respectively in $\mathcal{Te}'$) iff one of the following two conditions holds.

1. There is an $A \in \Delta$ such that (i) there is both an h-proposition in η of the form "A **happens-at** T" and a c-proposition in γ of the form "A **initiates** F **when** C" (respectively "A **terminates** F **when** C") and (ii) H satisfies C at T.

2. There is an r-proposition in ρ of the form "F **whenever** C" (respectively "$\neg F$ **whenever** C") and a partition $\{C_1, C_2\}$ of C such that (i) C_1 is non-empty, for each fluent constant $F' \in C_1$, $(F', T) \in \mathcal{In}$, and for each fluent literal $\neg F' \in C_1$, $(F', T) \in \mathcal{Te}$, and (ii) there is some $T_2 \in \Pi$, $T \prec T_2$, such that for all T_1, $T \preceq T_1 \preceq T_2$, H satisfies C_2 at T_1.

Let $\langle \mathcal{In}^f, \mathcal{Te}^f \rangle$ be the least fixed point of the (monotonic) operator $\mathcal{F}$ starting from the empty tuple $\langle \emptyset, \emptyset \rangle$. T is an *initiation-point* (respectively *termination-point*) *for F in H relative to D* iff $(F, T) \in \mathcal{In}^f$ (respectively $(F, T) \in \mathcal{Te}^f$). □

It is useful to note that any initiation or termination point T relative to D, defined in this way, must refer to at least one known h-proposition at T in the domain D. Therefore a domain which contains a finite number of h-propositions will necessarily contain only a finite number of initiation and termination points.

We can see how the new definition of initiation and termination points works by extending the stuffy room example with some t- and h-propositions. For the sake of variety, suppose that the set Π of time-points is the set of natural numbers with the usual ordering relation, and that in our domain description $D_{sr} = \langle \gamma_{sr}, \eta_{sr}, \tau_{sr}, \rho_{sr} \rangle$, γ_{sr} consists of c-propositions (sr1)–(sr4), ρ_{sr} consists of r-propositions (sr5)–(sr7), η_{sr} contains the h-propositions

$CloseWindow$ **happens-at** 2 (sr8)
$CloseVent$ **happens-at** 2 (sr9)

and τ_{sr} contains the t-propositions

$\neg WindowClosed$ **holds-at** 0 (sr10)
$\neg VentClosed$ **holds-at** 0 (sr11)

Intuitively, we can see that at any given time-point after 2, say 4, the room should be stuffy. To show that the semantics matches with this intuition, we need to show that

$$D_{sr} \models Stuffy \textbf{ holds-at } 4$$

and that D_{sr} is consistent. It is sufficient to observe that the following interpretation H_{sr} is the only model of D_{sr}. For all time-points (naturals) $\pi \in \Pi$ and $\zeta \in \{Stuffy, WindowClosed, VentClosed\}$,

$$H_{sr}(\zeta, \pi) = false \quad \text{if } \pi \leq 2$$
$$H_{sr}(\zeta, \pi) = true \quad \text{if } \pi > 2$$

According to Definition 14, the only possible initiation point or termination point for any of the fluents in H_{sr} relative to D_{sr} is 2, since this is the only point to which an h-proposition refers. The combination of (sr1) with (sr8) and the combination of (sr2) with (sr9) ensure that (in any model) 2 is an initiation point for $WindowClosed$ and $VentClosed$, by condition (1) of Definition 14. Hence, since $\{\{VentClosed, WindowClosed\}, \emptyset\}$ is a partition of $\{VentClosed, WindowClosed\}$, (sr5) ensures that (in any model) 2 is an initiation point for $Stuffy$ by condition (2) of Definition 14. It is now straightforward to check that H_{sr} satisfies all four conditions of Definition 13, and easy to see that there can be no other model of this domain. (Variations of this stuffy room example also behave correctly when represented in $\mathcal{E}$. In particular, if we replace (sr9) with "$CloseVent$ **happens-at** 3" our semantics does not give rise to the type of anomalous model which is problematic for some other approaches, e.g. [1], in which a change at 3 from $\neg Stuffy$ to $Stuffy$ is avoided by incorporating an intuitively unjustified change from $WindowClosed$ to $\neg WindowClosed$.)

Our second example is taken from [28], and is problematic for what the author terms as "categorization-based" approaches to ramification such as [14] and [25] (see [28] for a detailed discussion of why this is so). An electric circuit is arranged as in Figure 1, so that, when activated, the relay instantaneously pulls switch $Switch2$ open (i.e. disconnects it). We will use the fluent constants $Switch1$, $Switch2$ and $Switch3$ to represent that the respective switches are closed (i.e. making a connection), and the symbols $Relay$ and $Light$ respectively to indicate that these components are activated.

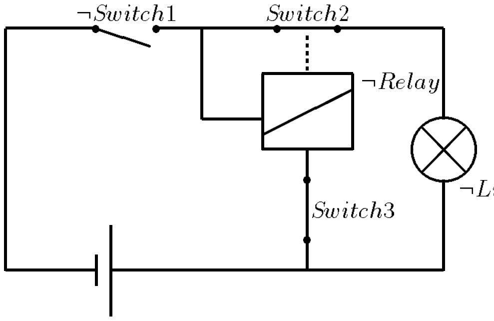

Figure 1

The representation of and reasoning about this domain is unproblematical for $\mathcal{E}$. The various permanent constraints on the state of the circuit can be represented with 7 r-propositions:

$Light$ **whenever** $\{Switch1, Switch2\}$ (c1)
$\neg Light$ **whenever** $\{\neg Switch1\}$ (c2)
$\neg Light$ **whenever** $\{\neg Switch2\}$ (c3)
$Relay$ **whenever** $\{Switch1, Switch3\}$ (c4)
$\neg Relay$ **whenever** $\{\neg Switch1\}$ (c5)
$\neg Relay$ **whenever** $\{\neg Switch3\}$ (c6)
$\neg Switch2$ **whenever** $\{Relay\}$ (c7)

The direct effects of opening (disconnecting) and closing (connecting) each of the three switches are captured in 6 c-propositions:

$CloseSwitch1$ **initiates** $Switch1$ (c8)
$OpenSwitch1$ **terminates** $Switch1$ (c9)
$CloseSwitch2$ **initiates** $Switch2$ (c10)
when $\{\neg Relay\}$
$OpenSwitch2$ **terminates** $Switch2$ (c11)
$CloseSwitch3$ **initiates** $Switch3$ (c12)
$OpenSwitch3$ **terminates** $Switch3$ (c13)

Taking the set Π of time-points to be the non-negative reals, we can describe the initial configuration of the circuit, pictured in Figure 1, as follows:

$\neg Switch1$ **holds-at** 0 (c14)
$Switch2$ **holds-at** 0 (c15)
$Switch3$ **holds-at** 0 (c16)
$\neg Light$ **holds-at** 0 (c17)
$\neg Relay$ **holds-at** 0 (c18)

(In fact, given condition (4) of Definition 13, only (c15) and (c16) are needed to adequately describe this initial configuration.)

If we add an occurrence of the action $CloseSwitch1$, say at 0.2, to this domain description:

$CloseSwitch1$ **happens-at** 0.2 (c19)

it should follow from $\mathcal{E}$'s semantics that, at points after 0.2, the relay is activated and $Switch2$ is disconnected, so that the light remains off. So, for example, taking D_c to consist of propositions (c1)–(c19), it should be the case that

$$D_c \models \neg Light \textbf{ holds-at } 0.4$$

and that D_c is consistent. As in the previous example, it is sufficient to show that there is a single model H_c for D_c, where H_c is defined as follows. For all $\zeta \in \{Switch1, Switch2, Switch3, Relay, Light\}$ and time-points (non-negative reals) $\pi \in \Pi$,

$$\begin{array}{ll} H_c(\zeta,\pi) = true & \text{if } \pi \leq 0.2 \text{ and} \\ & \zeta \in \{Switch2, Switch3\} \\ H_c(\zeta,\pi) = true & \text{if } \pi > 0.2 \text{ and} \\ & \zeta \in \{Switch1, Switch3, Relay\} \\ H_c(\zeta,\pi) = false & \text{otherwise} \end{array}$$

Clearly, the only possible initiation point or termination point for any of the fluents in any interpretation relative to D_c is 0.2. The combination of (c8) with (c19) ensures that (in any model) 0.2 is an initiation point for $Switch1$, by condition (1) of Definition 14. Hence, by condition (2) of Definition 14, 0.2 is an initiation point for $Relay$ in H_c, because $\{\{Switch1\}, \{Switch3\}\}$ is a partition of the set $\{Switch1, Switch3\}$ which appears on the righthand side of (c4), and any time-point $T_2 > 0.2$ is such that, for all T_1 in the closed interval $[0.2, T_2]$, $H_c(Switch3, T_1) = true$. Therefore, again by condition (2) of Definition 14, 0.2 is a termination point for $Switch2$ in H_c, since $\{\{Relay\}, \emptyset\}$ is a partition for the set $\{Relay\}$ which appears on the righthand side of (c7). Notice that 0.2 is not a termination point for $Switch3$ in H_c, because $Switch3$ does not appear in the left-hand side of any r-proposition in D_c, and there is no h-proposition mentioning the action $OpenSwitch3$ in D_c. Neither is 0.2 an initiation point for $Light$ in H_c – although $\{\{Switch1\}, \{Switch2\}\}$ is a partition of the set $\{Switch1, Switch2\}$ which appears on the righthand side of (c1), condition (2) of Definition 14 is not applicable because there is no $T_2 > 0.2$ such that $H_c(Switch2, T_2) = true$.

From this argument it is clear that in every model of D_c, 0.2 must be an initiation point for $Switch1$ and $Relay$ and a termination point for $Switch2$, and that $Switch3$ and $Light$ cannot have any initiation or termination points. Hence H_c is the only possible model of D_c.

Note that variations of D_c involving simultaneous action occurrences also give the intuitively expected results. For example, consider the domain description D_c^+, which is the same as D_c except that it also includes the h-proposition

$$OpenSwitch3 \ \textbf{happens-at} \ 0.2 \qquad \text{(c20)}$$

It is straightforward to check that the only model of D_c^+ is H_c^+, where for all $\zeta \in \{Switch1, Switch2, Switch3, Relay, Light\}$ and time-points (non-negative reals) $\pi \in \Pi$,

$$\begin{array}{ll} H_c^+(\zeta,\pi) = true & \text{if } \pi \leq 0.2 \text{ and} \\ & \zeta \in \{Switch2, Switch3\} \\ H_c^+(\zeta,\pi) = true & \text{if } \pi > 0.2 \text{ and} \\ & \zeta \in \{Switch1, Switch2, Light\} \\ H_c^+(\zeta,\pi) = false & \text{otherwise} \end{array}$$

Thus, for example,

$$D_c^+ \ \models \ Light \ \textbf{holds-at} \ 0.4$$

Discussion and Related Work

In this paper, which is a summary of the first part of [10], we have shown how the Language $\mathcal{E}$, extended with ramification statements or r-propositions, provides a simple, intuitive framework for representing and reasoning about direct and indirect effects of action occurrences embedded in an independent time structure. Time may be linear or branching, continuous or discrete. In [10] we have also given a general method of translating domain descriptions with r-propositions into ordinary (Prolog-like) Event Calculus style logic programs, and shown that this translation is sound.

In our formalisation, r-propositions play a double role. On the one hand, the proposition "L **whenever** C" acts as a static domain constraint – Definition 13 allows us to reason that whenever everything in C holds, L holds, and whenever L doesn't hold, something in C doesn't hold. On the other hand, it generates new causal information as far as the dynamic behaviour of the domain is concerned, through indirect initiation and termination. Roughly, when C is initiated, L is also initiated. This dymanic aspect is unidirectional; it is not the case that when L is terminated, C is terminated.

The issue of ramifications and causality has recently received widespread attention in the context of reasoning about actions and change. As has been noted by authors such as Sandewall [26] and Thielscher [28], most recent approaches to ramification can be classed as either *categorization-based* or *causal-based.* (Not all approaches readily fit into this classification, however. For example the recent approach of McIlraith [18, 19] has both a causal and categorization flavour.) Categorization-based approaches overcome problems with more naïve applications of the principle of minimal change by (temporarily or permanently) categorizing fluents into two or more classes. The principle of minimisation of change can then be applied more stringently to some of these classes than to others. Variations on this theme can be found, for example, in the approaches of Lifschitz [14] (in his use of *frame* and *non-frame* fluents), in Sandewall [24] [26] and Gustafsson & Doherty [8] (in their use of *occlusion*), and in Kartha & Lifschitz [12] and Shanahan [27] (in their use of a *releases* mechanism or predicate). Such approaches deal easily with domains such as the stuffy room example (for example, *Stuffy* would simply be a non-frame fluent in [14], or released in [12] and [27]). However, as pointed out in [28], these approaches cope less easily with domains such as Thielscher's circuit example.

The approach to ramification described here bears a stronger resemblance to causal-based approaches, such as Thielscher's [28], McCain & Turner's [16, 17], Lin's [15] and Turner's [29]. This is because the effects of action occurrences cannot be propagated "backwards" through r-propositions, and this type of unidirectionality of indirect effects is a key characteristic of causal-based techniques. However, there are difficulties in forming detailed comparisons between $\mathcal{E}$ and some of these approaches. This is because, whereas $\mathcal{E}$ is a narrative-based formalism in which the notion of an action occurrence within some explicit time structure is central, these formalisms are state-based and often do not incorporate an explicit notion of time or of an occurrence. The emphasis is instead on the type of "hypothetical reasoning" about sequences of actions supported by the Situation Calculus. Although $\mathcal{E}$ supports this type of reasoning (as illustrated in the second section of this paper), it can also deal with more arbitrary, independent time structures. (In [17], McCain & Turner do introduce explicit time, but the time structure must be linear and discrete.) In our semantics, change is characterised by a series or collection of initiation and termination points, which are time-points at which particular actions occur. In other words, change is "caused" by explicitly represented, temporally embedded events. Explicit reference to the flow of time has made it easy to generalise the semantics so that it covers ramifications, i.e. indirect cause and effect. The unidirectionality of r-propositions is initimately related to the unidirectionality of this independent flow of time. Note that the idea of initiation and termination points is already implicit in the original Event Calculus [13]. Our contribution has been to make these notions explicit, and hence amenable to generalisation, in the semantics.

For Thielscher's approach [28] at least, the extension to deal with explicit time and narrative information seems likely to be a source of difficulty. Thielscher computes ramifications by applying an *action description* to an initial state, and then generating a sequence of temporary states using *causal relationships* until a *successor state* is reached which satisfies all the *domain constraints* of the system. If explicit time is added to this schema, the temporal status of these temporary states is questionable. If the temporary states are somehow assigned to successive time-points, the domain constraints may be violated at these time-points, and extra mechanisms will have to be introduced to deal with interruptions to the propagation of effects due to other action occurrences. If on the other hand these temporary states are "outside time", so that the overall change from initial state to successor state happens instantaneously or in a single time step, it seems that Thielscher's framework will sometimes give rise to anomalous models. For example, taking Thielscher's light-detector example (from [28], where a light detector is permanently activated whenever a light bulb emits light) we would, as explained in [28], be able to generate a model where the detector becomes activated even though there is no time-point at which the light bulb is on.

In their Language $\mathcal{ER}$ [30, 31, 32], Van Belleghem, Denecker & Dupré introduce some interesting extensions and variations to the ideas presented in the present paper and in [9] and [10]. Roughly speaking (and among numerous other innovations), they extend the expressivity given by r-propositions in $\mathcal{E}$ as follows. When re-expressing the r-proposition "F **whenever** C" as a *derived effect rule* in $\mathcal{ER}$, the syntax allows the user to stipulate that, when constructing a model, only some partitions of the set C may be applied within a definition analogous to condition (2) of Definition 14. This extra expressivity is motivated by an example involving the activation of an alarm system. Domain descriptions written in $\mathcal{ER}$ can also include simple *state constraints*, further adding to its expressivity. However, $\mathcal{ER}$ is arguably less general than $\mathcal{E}$ in the sense that it uses only a particular linear (real number) model of time. Aside from its specific contributions in terms of representational techniques, the development of $\mathcal{ER}$ illustrates that, as for action description languages such as $\mathcal{A}$ and $\mathcal{AR}$, "event description languages" (e.g. $\mathcal{E}$ and $\mathcal{ER}$) can form part of a spectrum, in which each individual language incorporates trade-offs between simplicity, succinctness and range of applicability. (There are also some interesting technical differences between $\mathcal{E}$'s and $\mathcal{ER}$'s semantics, which are discussed in [30] and deserve further investigation. These regard certain combinations of ramification statements which give rise to non-determinism in $\mathcal{E}$ but inconsistency in $\mathcal{ER}$. See *http://www.dcs.qmw.ac.uk/˜rsm/abstract15.html* for one such example.)

The techniques discussed in [9] for *explanation* and *planning* in terms of $\mathcal{E}$ are equally applicable where domains include ramifications described by r-propositions. In particular, suppose that we wish to express narrative information as a sequence of observations of fluent values at different time-points (i.e. as a set of t-propositions), rather than as a sequence of action occurrences (i.e. a set of h-propositions). In these circumstances we may use the notion of *h-entailment*, defined in [9] as a generalisation of $\mathcal{E}$'s ordinary entailment, to compute other consequences of our domain theory. To illustrate, we take (and slightly modify) a simple example of Sandewall, which he introduces in [26] to illustrate a limitation of current causal approaches to ramification. Sandewall asks us to consider the case of a lamp LA which is turned on independently by two different switches. Sandewall describes this with the classical constraint $(LA \leftrightarrow (SW_1 \vee SW_2))$. In $\mathcal{E}$ we would use three r-propositions:

$$\neg LA \textbf{ whenever } \{\neg SW_1, \neg SW_2\} \quad \text{(la1)}$$
$$LA \textbf{ whenever } \{SW_1\} \quad \text{(la2)}$$
$$LA \textbf{ whenever } \{SW_2\} \quad \text{(la3)}$$

Closing either of the switches activates that switch, and (taking time as the naturals) initially the lamp is off (note that the semantics of $\mathcal{E}$ then implies that neither switch is initially closed):

$$CloseSW_1 \textbf{ initiates } SW_1 \quad \text{(la4)}$$
$$CloseSW_2 \textbf{ initiates } SW_2 \quad \text{(la5)}$$
$$\neg LA \textbf{ holds-at } 0 \quad \text{(la6)}$$

Sandewall's point is essentially that at some later timepoint, say 5, we might wish to simply state that LA becomes true, without explicitly stating that one of the switches has been closed. In terms of $\mathcal{E}$, we simply want to state:

$$LA \textbf{ holds-at } 5 \quad \text{(la7)}$$

We should then be able to infer that at 5 at least one of the switches is closed. Using the definitions of h-models, h-consistency and h-entailment in [9], it is straightforward to verify that the domain description consisting of (la1)-(la7) is h-consistent, and that in all h-models of this domain, either SW_1 is true or SW_2 is true at all times greater than or equal to 5, as intended by the example.[2] Note that for all other examples given in this paper, h-entailment coincides exactly with ordinary entailment.

There is considerable scope for further work on the themes of this paper. First, it would be instructive to investigate more formally and systematically the correspondence between $\mathcal{E}$ and other formalisms which deal with ramifications. Much of the discussion here has centered around a few benchmark problems. Sandewall [24] and others have argued the benefits of a more systematic methodology for assessing and comparing different formalisms for reasoning about actions. As regards ramifications, Sandewall has already undertaken an extensive analysis of various categorization-based formalisms [26], comparing them both pairwise and against his own "semantics-level" template. It would certainly be beneficial to apply this same methodology to assess the range of applicability of $\mathcal{E}$. (In Sandewall's terminology, $\mathcal{E}$ might best be described as a "semantics-level" formalism, since by itself it offers no proof theory. Like Sandewall, our primary intention has been to give a natural and intuitive definition of a model-theoretic entailment relation, against which we have measured the correctness of a class of logic programs.)

We would also like to investigate the translation of $\mathcal{E}$ domains into other general-purpose formalisms, such as classical logic, abductive logic programming, and argumentation frameworks. In [10] the translation is into logic programs with negation-as-failure (NAF). In analysing properties of this translation [10] uses only a basic (and limited) understanding of NAF, in terms of the the SLDNF proof procedure. Any of the more sophisticated semantics of logic programs subsumes this very procedural understanding. The translation and analysis therefore provides and verifies a baseline implementation but could almost certainly be improved in several ways. In particular, the incompleteness of the translation could probably be narrowed down if more general forms of NAF were to be used. In fact, Dimopoulos & Kakas [4] have already developed a translation of the Language $\mathcal{A}$ into an argumentation framework, and shown this to be complete for consistent domains. We are currently investigating analogous translations of the Language $\mathcal{E}$ with similar properties. Turner's work [29] in providing a sound and complete translation of a causal extension of $\mathcal{A}$ into logic programs is also relevant here.

There are many ways in which the expressiveness of $\mathcal{E}$ might be extended (as has already been partly illustrated in [30], [31] and [32]). The syntax of $\mathcal{E}$ already allows for *concurrent* action occurrences, and it seems likely that it could be extended to enable the description of interactions between, and combinations of effects of, concurrent actions. Since in $\mathcal{E}$ the flow of time may be continuous, it is a good candidate formalism for dealing with *continuous change*, perhaps by adapting some of the ideas in [21]. Related to this is the issue of *triggered events* (or *natural actions*). Preliminary investigations have already shown that, since $\mathcal{E}$ is narrative-based, it is relatively straightforward to extend its syntax and semantics to include and assign meaning to statements such as "C **triggers** A" – i.e. "at time points when the set of fluent literals C holds, action A occurs". Finally, we believe that narrative based formalisms such as $\mathcal{E}$ can offer interesting perspectives on issues relating to the *qualification problem*.

[2] A trivial extension to the syntax of $\mathcal{E}$ and to condition (4) in Definition 13 of a model would also allow us to write and assign meaning to t-propositions with disjunctions, such as "$(SW_1 \vee SW_2)$ **holds-at** 5". These kinds of extensions are discussed more generally in [31].

For example, they enable a distinction to be made between circumstances under which it is impossible for an action to occur, and circumstances under which is it impossible to predict the effects of an (attempt to perform an) action. Some preliminary ideas for qualification statements in $\mathcal{E}$ are included in the appendix of [9], but much further work needs to be done.

Acknowledgements

Thanks to Kristof Van Belleghem, Tom Costello, Michael Gelfond, Michael Thielscher, Hudson Turner and Pascal Walheim for helpful comments relating to various aspects of this paper and/or its longer version [10]. This research was funded by the EPSRC and the University of Cyprus.

References

[1] Andrew Baker, *Nonmonotonic Reasoning in the Framework of the Situation Calculus*, Artificial Intelligence, Vol 49(5-23), 1991.

[2] Chitta Baral and Michael Gelfond, *Representing Concurrent Actions in Extended Logic Programming*, Proceedings IJCAI 93, Morgan Kaufmann, page 866, 1993.

[3] Chitta Baral, Michael Gelfond and Alessandro Provetti, *Representing Actions: Laws, Observations and Hypotheses*, The Journal of Logic Programming, Vol 31(1-3) (Special Issue on Reasoning about Action and Change), Elsevier Science, 1997.

[4] Yannis Dimopoulos and Antonis Kakas, *Logic Programming without Negation as Failure*, Proceedings of the 1995 International Logic Programming Symposium (ILPS'95), pages 369-383, MIT Press, 1995.

[5] Phan Minh Dung, *Representing Actions in Logic Programming and its Applications in Database Updates*, Proceedings of the Tenth International Conference on Logic Programming, ed David S. Warren, MIT Press, pages 222-238, 1993.

[6] Michael Gelfond and Vladimir Lifschitz, *Representing Actions in Extended Logic Programming*, Proceedings of the Joint International Conference and Symposium on Logic Programming, ed. Krzysztof Apt, MIT Press, page 560, 1992.

[7] Michael Gelfond and Vladimir Lifschitz, *Representing Action and Change by Logic Programs*, Journal of Logic Programming, volume 17 numbers 2, 3 and 4, North-Holland, pages 301-322, 1993.

[8] Joakim Gustafsson and Patrick Doherty, *Embracing Occlusion in Specifying the Indirect Effects of Actions*, in Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR'96), ed.s Aiello, Doyle and Shapiro, Morgan Kaufmann, 1996.

[9] Antonis Kakas and Rob Miller, *A Simple Declarative Language for Describing Narratives with Actions*, The Journal of Logic Programming, Vol 31(1-3) (Special Issue on Reasoning about Action and Change), pages 157-200, Elsevier Science, 1997.

[10] Antonis Kakas and Rob Miller, *Reasoning about Actions, Narratives and Ramifications*, Linkoping Electronic Articles in Computer and Information Science, Vol. 2(1997): nr 12. http://www.ep.liu.se/ea/cis/1997/012/. Also posted and under public review in the News Journal of Electronic Transactions on Artificial Intelligence (http://www.ida.liu.se/ext/etai/), 1997.

[11] G. Neelakantan Kartha, *Soundness and Completeness Theorems for Three Formalizations of Action*, Proceedings IJCAI 93, page 724, 1993.

[12] G. Neelakantan Kartha and Vladimir Lifschitz, *Actions with Indirect Effects (Preliminary Report)*, in Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning, pages 341-350, 1994.

[13] Robert Kowalski and Marek Sergot, *A Logic-Based Calculus of Events*, New Generation Computing, vol 4, page 267, 1986.

[14] Vladimir Lifschitz, *Frames in the Space of Situations*, Artificial Intelligence Journal, Volume 46, pages 365-376, 1990.

[15] Fangzhen Lin, *Embracing Causality in Specifying the Indirect Effects of Actions*, In C. S. Mellish, ed., Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95), pages 1985-1991, 1995.

[16] Norman McCain and Hudson Turner, *A Causal Theory of Ramifications and Qualifications*, In C. S. Mellish, ed., Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95), pages 1978-1983, 1995.

[17] Norman McCain and Hudson Turner, *Causal Theories of Action and Change*, In Proceedings of AAAI'97, 1997.

[18] Sheila McIlraith, *A Closed-Form Solution to the Ramification Problem (Sometimes)*, In working notes of the IJCAI'97 workshop on Nonmonotonic Reasoning, Action and Change, 1997.

[19] Sheila McIlraith, *Representing Actions and State Constraints in Model-Based Diagnosis*, In Proceedings of AAAI'97, 1997.

[20] Rob Miller, *Situation Calculus Specifications for Event Calculus Logic Programs*, in Proceedings of the Third International Conference on Logic Programming and Non-monotonic Reasoning, Lexington, KY, USA, Springer Verlag, 1995.

[21] Rob Miller and Murray Shanahan, *Reasoning about Discontinuities in the Event Calculus*, in Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96), ed.s L. C. Aiello, J. Doyle & S. C. Shapiro, Morgan Kaufmann, 1996.

[22] Edwin Pednault, *ADL: Exploring the Middle Ground between STRIPS and the Situation Calculus*, In Ronald Brachman, Hector Levesque and Ray Reiter, ed.s, Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, pages 324-332, 1989.

[23] Ray Reiter, *The Frame Problem in the Situation Calculus: a Simple Solution (sometimes) and a Completeness Result for Goal Regression*, in Vladimir Lifschitz, ed., Artificial Intelligence and Mathematical Theory of Computation, pages 359-380, Academic Press, 1991.

[24] Erik Sandewall, *Features and Fluents: A Systematic Approach to the Representation of Knowledge about Dynamical Systems*, Oxford University Press, 1994.

[25] Erik Sandewall, *Reasoning about Actions and Change with Ramification*, in Computer Science Today, volume 1000 of LNCS, Springer, 1995.

[26] Erik Sandewall, *Assessments of Ramification Methods that Use Static domain Constraints*, in Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning (KR'96), ed.s Aiello, Doyle and Shapiro, Morgan Kaufmann, 1996.

[27] Murray Shanahan, *Solving the Frame Problem: A Mathematical Investigation of the Commonsense Law of Inertia*, MIT Press, 1997.

[28] Michael Thielscher, *Ramification and Causality*, Artificial Intelligence, 1997.

[29] Hudson Turner, *Representing Actions in Logic Programs and Default Theories: a Situation Calculus Approach*, The Journal of Logic Programming, Vol 31(1-3) (Special Issue on Reasoning about Action and Change), Elsevier Science, 1997.

[30] Kristof Van Belleghem, *Open Logic Programming as a Knowledge Representation Language for Time-dependent Problem Domains*, Ph.D. thesis, K. U. Leuven, to be submitted, 1997.

[31] Kristof Van Belleghem, Marc Deneker and Daniele Theseider Dupré, *Ramifications and Dependencies in an Event-based Language*, submitted for publication, 1997.

[32] Kristof Van Belleghem, Marc Deneker and Daniele Theseider Dupré, *Ramifications in an Event-based Language*, in Proceedings of the 9th Dutch Artificial Intelligence Conference (NAIC 97), 1997.