Available online at www.sciencedirect.com

SCIENCE DIRECT®

Artificial Intelligence 170 (2006) 1017–1029

Artificial Intelligence

www.elsevier.com/locate/artint

Event calculus and temporal action logics compared

Erik T. Mueller

IBM Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

Received 11 November 2005; received in revised form 8 May 2006; accepted 8 May 2006

Available online 12 June 2006

Abstract

We compare the event calculus and temporal action logics (TAL), two formalisms for reasoning about action and change. We prove that, if the formalisms are restricted to integer time, inertial fluents, and relational fluents, and if TAL action type specifications are restricted to definite reassignment of a single fluent, then the formalisms are not equivalent. We argue that equivalence cannot be restored by using more general TAL action type specifications. We prove however that, if the formalisms are further restricted to single-step actions, then they are logically equivalent.

Keywords: Commonsense reasoning; Reasoning about action and change; Event calculus; Temporal action logics (TAL)

1. Introduction

Reasoning about action and change is a fundamental area of research within artificial intelligence. This is an important area because action and change are pervasive aspects of the world in which intelligent agents operate. Over the years, a number of formalisms and frameworks for reasoning about action and change have been developed. Among them are the situation calculus [25,33], the event calculus [19,36], features and fluents [34,35], action languages [7–9], and the fluent calculus [12,41,42].

Although there has been some cross-pollination, the various formalisms have been developed in relative isolation, and the relationship between them is not always well understood. But understanding the relationship between the formalisms is important for the following reasons:

- It helps to advance the field. An understanding of the space of possible formalisms and where each formalism is situated in this space is essential to their refinement.
- It enables sharing of reasoning tools. A number of reasoning tools are available, as shown in Table 1. If problems in one formalism can be translated into another formalism, they can be solved using reasoning tools for the other formalism.
- It enables sharing of problem libraries developed for each of the formalisms and reasoning tools.
- It facilitates collaboration. Researchers working using one formalism can understand and build on the results of researchers using another formalism.

E-mail address: etm@us.ibm.com (E.T. Mueller).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2006.05.001

Table 1
Tools for reasoning about action and change

Formalism	Tool
Situation calculus	KM [3]
	http://www.cs.utexas.edu/users/mfkb/km.html
Event calculus	Event calculus planner [39]
	http://www.iis.ee.ic.ac.uk/~mpsha/planners.html
	Discrete Event Calculus Reasoner [28]
	http://decreasoner.sourceforge.net/
TAL	VITAL [20]
	http://www.ida.liu.se/~jonkv/vital/
$\mathcal{C}+$	CCALC [9,23]
	http://www.cs.utexas.edu/users/tag/cc/
$\mathcal{E}$	$\mathcal{E}$-RES [14,15]
	http://www.ucl.ac.uk/~uczcrsm/LanguageE/
Fluent calculus	FLUX [43]
	http://www.fluxagent.org/

Two major streams of research in reasoning about action and change are temporal action logics (TAL) [4–6,21], which has its origins in the features and fluents framework, and the event calculus [29,37]. TAL and the event calculus appear to be similar because they both have characterizations in classical logic and both use linear time. But their exact relationship has been unclear.

In this paper, we compare the event calculus with support for events with duration [26,37] and TAL 1.0 [4,5].[1] We start by restricting the event calculus and TAL 1.0 to integer time, inertial fluents, and relational fluents. We further restrict TAL 1.0 action type specifications to definite reassignment of a single fluent. We then prove that these restricted versions are not equivalent. We show that equivalence cannot be restored even if more general TAL action type specifications are used. We then further restrict the two formalisms to single-step actions and prove that these versions are logically equivalent.

2. Past work

In the past, four approaches have been used to compare formalisms for reasoning about action and change:

(1) Two formalisms are proved to be logically equivalent.
(2) A syntactic translation is defined from a domain description in one formalism to a domain description in another formalism, and the two domain descriptions are proved to entail the same results. Translations may be provided in one or both directions.
(3) Semantic (model theoretic) conditions are defined under which a domain description in one formalism matches a domain description in another formalism, and matching domain descriptions are proved to entail the same results.
(4) A general formalism is defined, and formalisms are shown to be special cases of the general formalism.

In order to ease comparison, the formalisms are often extended or restricted in various ways.

The first approach is used by Kowalski and Sadri [17,18], who consider a version of the event calculus extended with branching time, but without concurrent events, continuous change, and release from the commonsense law of inertia. They show that this version of the event calculus is logically equivalent to a version of the situation calculus similar to that of Reiter [32]. The first approach is also used by Mueller [27], who proves that, if the domain of the timepoint sort is restricted to the integers, the continuous event calculus is logically equivalent to a discrete version of the event calculus.

We use the first approach in this paper.

[1] We use the variant of TAL 1.0 in which actions are treated as first-class citizens [5, pp. 19–20].

The second approach is used by a number of researchers. Kartha [16] defines a translation from domain descriptions of action language $\mathcal{A}$ [8] into three versions of the situation calculus [1,30,31]. He asserts that for any sequence of events, the $\mathcal{A}$ domain description and the situation calculus translations entail the same truth values of fluents.

Thielscher [40] restricts $\mathcal{A}$ to a single sequence of actions, and restricts ego world semantics [35] to inertial fluents, relational fluents, and single-step actions. He defines a translation from $\mathcal{A}$ domain descriptions to ego world semantics domain descriptions, and defines a translation from ego world semantics domain descriptions to $\mathcal{A}$ domain descriptions. He sketches proofs that in both cases, the models of the domain descriptions entail the same event occurrences and fluent truth values.

Giunchiglia and Lifschitz [11] define a translation from unrestricted domain descriptions of action language $\mathcal{C}$ [10] into the situation calculus, and define a classical logic translation of the transition semantics of $\mathcal{C}$ domain descriptions. They prove that for any domain description, the two translations are logically equivalent. They also define a translation from restricted $\mathcal{C}$ domain descriptions to TAL domain descriptions, and define another classical logic translation of the transition semantics of $\mathcal{C}$ domain descriptions. They prove that for any domain description, the first translation is a conservative extension of the second translation.

The third approach is used by Miller and Shanahan [26], who consider a version of the $\mathcal{E}$ action language [13] and a version of the event calculus without release from the commonsense law of inertia, continuous change, and state constraints. They define semantic conditions under which an $\mathcal{E}$ domain description matches an event calculus domain description. They prove that, if an $\mathcal{E}$ domain description matches an event calculus domain description, the domain descriptions entail the same event occurrences and fluent truth values.

The fourth approach is used by Van Belleghem, Denecker, and De Schreye [44], who define a general formalism that encompasses both the situation calculus and a version of the event calculus without concurrent events, continuous change, and release from the commonsense law of inertia. They describe how the situation calculus and this version of the event calculus are obtained by restricting the general formalism.

Bennett and Galton [2] define a versatile event logic (VEL) whose semantics includes a number of formalisms for temporal reasoning, and present ways of describing the situation calculus and the event calculus within VEL. They consider a version of the event calculus without continuous change and release from the commonsense law of inertia.

A related approach is that of Sandewall [35], who defines ontological families and the intended models of a domain description of a given family. The correctness of any particular formalism is then assessed against these formal specifications.

3. The event calculus and TAL

How shall we go about proving logical equivalence of the event calculus with events with duration and TAL 1.0? As shown in Table 2, the formalisms do not support the same features. In addition, the formalisms do not address indirect effects and nondeterministic effects using the same language features. Indirect effects are represented in the event calculus using causal constraints and effect constraints [38], whereas they are represented in TAL 1.0 using dependency constraints [5, pp. 16–18]. Nondeterministic effects are represented in the event calculus using determining fluents [37, pp. 419–420], whereas they are represented in TAL 1.0 using disjunctions in reassignment operators [4, pp. 35–36].

At this point, we have two choices. We can either extend the formalisms with their missing features, or we can restrict the formalisms to their common features. We choose the second approach. We disallow causal constraints, continuous change, continuous time, and effect constraints in the event calculus, and we disallow dependency constraints, disjunctions in reassignment operators, durational fluents, and functional fluents in TAL 1.0.

In order to prove logical equivalence, we would like to characterize the event calculus and TAL 1.0 using the same classical language. TAL 1.0 domain descriptions are written in a specialized language $\mathcal{L}(ND)$ and then translated into a classical language $\mathcal{L}(FL)$. The important translation rules are provided in Appendix A. Event calculus domain descriptions are also expressed in a classical language. But TAL 1.0 $\mathcal{L}(FL)$ and the event calculus use different symbol sets. TAL 1.0 $\mathcal{L}(FL)$ uses *Holds*, *Occurs*, and *Occlude*, whereas the event calculus as we have restricted it uses *HoldsAt*, *InitiallyP*, *InitiallyN*, *Happens*, *Initiates*, *Terminates*, *Clipped*, and *Declipped*. How shall we reconcile these two languages?

In TAL 1.0, the effects of events are specified using $\mathcal{L}(ND)$ action type specifications of the form

$$[\tau, \tau']\alpha \rightsquigarrow \beta$$

Table 2
Features of the event calculus and TAL 1.0

Feature	Event calculus	TAL 1.0
Causal constraints	✓	
Concurrent events	✓	✓
Context-sensitive effects	✓	✓
Continuous change	✓	
Continuous time	✓	
Dependency constraints		✓
Disjunctions in reassignment operators		✓
Durational fluents		✓
Effect constraints	✓	
Events with duration	✓	✓
Functional fluents		✓
Inertial fluents	✓	✓
State constraints	✓	✓

where τ and τ' are timepoints, α is an action, and β is a formula that specifies the preconditions and postconditions of α. Postconditions are defined using the R, X, and I reassignment operators. The R operator specifies that a fluent is released from the commonsense law of inertia during a time interval and is constrained to have a particular value at the end of the interval. The I operator specifies that a fluent is released from the commonsense law of inertia during a time interval and is constrained to have a particular value during the interval. The X operator specifies that a fluent is released from the commonsense law of inertia during a time interval. Note that a fluent released from the commonsense law of inertia by a reassignment operator may be further constrained by other parts of the same action type specification or by different axioms such as state constraints.

The effects of actions are often[2] specified using $\mathcal{L}(ND)$ action type specifications of the form

$$[\tau, \tau']\alpha \leadsto [\tau]\gamma \rightarrow R((\tau, \tau'](\neg)\beta) \tag{1}$$

where τ and τ' are timepoints, α is an action, γ is a formula, and β is a fluent. This represents that, if α occurs from τ to τ', and γ is true at τ, then β will be of indeterminate truth value from $\tau + 1$ to $\tau' - 1$ inclusive, and will be true (false) starting at τ'. A specification of this form is translated into an $\mathcal{L}(FL)$ formula

$$Occurs(\tau, \tau', \alpha) \rightarrow (\gamma' \rightarrow (\neg)Holds(\tau', \beta) \wedge \forall t(\tau < t \wedge t \leqslant \tau' \rightarrow Occlude(t, \beta))) \tag{2}$$

where γ' is the $\mathcal{L}(FL)$ translation of $[\tau]\gamma$.[3] Notice that (2) is logically equivalent to the conjunction of the formulas:

$$Occurs(\tau, \tau', \alpha) \wedge \gamma' \rightarrow (\neg)Holds(\tau', \beta)$$
$$Occurs(\tau, \tau', \alpha) \wedge \gamma' \wedge \tau < t \wedge t \leqslant \tau' \rightarrow Occlude(t, \beta)$$

In order to reconcile the two languages, we assume that all TAL 1.0 action type specifications are of the form (1). Given this assumption we show in Sections 4 and 6 that the restricted TAL 1.0 is not equivalent to the restricted event calculus. We argue in Section 7 that, even if we relax this assumption and allow more general action type specifications, we still cannot obtain equivalence.

Given this assumption we can express $[\tau, \tau']\alpha \leadsto [\tau]\gamma \rightarrow R((\tau, \tau']\beta)$ as $\gamma' \rightarrow Initiates(\alpha, \beta, \tau)$, and $[\tau, \tau']\alpha \leadsto [\tau]\gamma \rightarrow R((\tau, \tau']\neg\beta)$ as $\gamma' \rightarrow Terminates(\alpha, \beta, \tau)$, just as in the event calculus. We must then add the following domain-independent axioms to TAL:

$$Occurs(t_1, t_2, e) \wedge Initiates(e, f, t_1) \rightarrow Holds(t_2, f) \tag{3}$$

$$Occurs(t_1, t_2, e) \wedge Terminates(e, f, t_1) \rightarrow \neg Holds(t_2, f) \tag{4}$$

[2] 46 of the 68 unique action type specifications provided with the latest release of VITAL, version 2.999.910 alpha of October 8, 2003, are of this form or can be rewritten as several specifications of this form. 8 specifications are of the form $[\tau, \tau']\alpha \leadsto [\tau]\gamma \rightarrow I((\tau, \tau'](\neg)\beta)$ where β is a durational fluent, 5 specifications involve $R([\tau'](\neg)\beta)$, 3 specifications use disjunctions in reassignment operators, and the remaining 6 specifications use other combinations of reassignment operators. Current TAL domains are more complex, but have not yet been added to VITAL.

[3] For example, the $\mathcal{L}(FL)$ translation of $[\tau]\beta_1 \wedge \neg\beta_2$ is $Holds(\tau, \beta_1) \wedge \neg Holds(\tau, \beta_2)$. See Appendix A for more details.

$$(\mathit{Initiates}(e, f, t_1) \vee \mathit{Terminates}(e, f, t_1)) \wedge \mathit{Occurs}(t_1, t_2, e) \wedge t_1 < t \wedge t \leqslant t_2 \rightarrow \mathit{Occlude}(t, f) \tag{5}$$

Furthermore, we treat the TAL 1.0 $\mathcal{L}(FL)$ predicates $\mathit{Holds}(t, f)$, $\mathit{Occurs}(t_1, t_2, e)$, and $\mathit{Occlude}(t, f)$, and the event calculus predicates $\mathit{InitiallyP}(f)$, $\mathit{InitiallyN}(f)$, $\mathit{Clipped}(t_1, f, t_2)$, and $\mathit{Declipped}(t_1, f, t_2)$ as abbreviations.

We use a many-sorted language with equality, with sorts for events, fluents, and timepoints. The domain of the timepoint sort is the integers. The language has the following predicates:

- $\mathit{HoldsAt}(f, t)$: Fluent f is true at timepoint t.
- $\mathit{Happens3}(e, t_1, t_2)$: Event e occurs from timepoint t_1 to timepoint t_2.
- $\mathit{Initiates}(e, f, t_1)$: If event e occurs from timepoint t_1 to timepoint t_2, then fluent f will be true after t_2.
- $\mathit{Terminates}(e, f, t_1)$: If event e occurs from timepoint t_1 to timepoint t_2, then fluent f will be false after t_2.

3.1. TALA axiomatization

We use the following axiomatization of the restricted TAL 1.0, which we call TALA. We start with definitions of *Holds* and *Occurs*.

TALA1 $\mathit{Holds}(t, f) \stackrel{\text{def}}{\equiv} \mathit{HoldsAt}(f, t)$

TALA2 $\mathit{Occurs}(t_1, t_2, e) \stackrel{\text{def}}{\equiv} \mathit{Happens3}(e, t_1, t_2 - 1)$

Note the difference in the ending timepoint t_2 between *Occurs* and *Happens3*.

TAL uses circumscription [22,24] of the *Occlude* and *Occurs* predicates [21, p. 26], just as the event calculus uses circumscription of *Initiates*, *Terminates*, and *Happens* [37, p. 417]. We continue with a definition based on the circumscription of *Occlude* in (5). We compute the circumscription using Proposition 2 of Lifschitz [22], which reduces circumscription to predicate completion.

TALA3 $\mathit{Occlude}(t, f) \stackrel{\text{def}}{\equiv} \exists e, t_1, t_2\, ((\mathit{Initiates}(e, f, t_1) \vee \mathit{Terminates}(e, f, t_1)) \wedge \mathit{Occurs}(t_1, t_2, e) \wedge t_1 < t \wedge t \leqslant t_2)$

The TAL 1.0 nochange axiom $\neg \mathit{Occlude}(t + 1, f) \rightarrow (\mathit{Holds}(t + 1, f) \leftrightarrow \mathit{Holds}(t, f))$ [4, p. 30] is logically equivalent to the conjunction of the axioms TALA4 and TALA5.

TALA4 $\mathit{Holds}(t, f) \wedge \neg \mathit{Occlude}(t + 1, f) \rightarrow \mathit{Holds}(t + 1, f)$

TALA5 $\neg \mathit{Holds}(t, f) \wedge \neg \mathit{Occlude}(t + 1, f) \rightarrow \neg \mathit{Holds}(t + 1, f)$

We continue with formulas (3) and (4).

TALA6 $\mathit{Occurs}(t_1, t_2, e) \wedge \mathit{Initiates}(e, f, t_1) \rightarrow \mathit{Holds}(t_2, f)$

TALA7 $\mathit{Occurs}(t_1, t_2, e) \wedge \mathit{Terminates}(e, f, t_1) \rightarrow \neg \mathit{Holds}(t_2, f)$

We finish with a constraint on the starting and ending timepoints of an event occurrence.

TALA8 $\mathit{Occurs}(t_1, t_2, e) \rightarrow t_1 < t_2$

Let TALA be the formula generated by conjoining axioms TALA4 through TALA8 and then expanding the predicates *Holds*, *Occurs*, and *Occlude* using definitions TALA1 through TALA3.

3.2. ECA axiomatization

There are two versions of the event calculus that support events with duration [26,37]. Both are candidates for equivalence with TAL 1.0. We start by using the following axiomatization of the event calculus. It is obtained from the version of the event calculus of Shanahan [37, p. 416] by eliminating *Releases*, and replacing *InitiallyP*, *InitiallyN*, *Clipped*, and *Declipped* with definitions.

ECA1 $InitiallyP(f) \stackrel{\text{def}}{\equiv} HoldsAt(f, 0)$
ECA2 $InitiallyN(f) \stackrel{\text{def}}{\equiv} \neg HoldsAt(f, 0)$
ECA3 $Clipped(t_1, f, t_4) \stackrel{\text{def}}{\equiv} \exists e, t_2, t_3\, (Happens3(e, t_2, t_3) \wedge t_1 < t_3 \wedge t_2 < t_4 \wedge Terminates(e, f, t_2))$
ECA4 $Declipped(t_1, f, t_4) \stackrel{\text{def}}{\equiv} \exists e, t_2, t_3\, (Happens3(e, t_2, t_3) \wedge t_1 < t_3 \wedge t_2 < t_4 \wedge Initiates(e, f, t_2))$
ECA5 $InitiallyP(f) \wedge \neg Clipped(0, f, t) \rightarrow HoldsAt(f, t)$
ECA6 $InitiallyN(f) \wedge \neg Declipped(0, f, t) \rightarrow \neg HoldsAt(f, t)$
ECA7 $Happens3(e, t_1, t_2) \wedge Initiates(e, f, t_1) \wedge t_2 < t_3 \wedge \neg Clipped(t_1, f, t_3) \rightarrow HoldsAt(f, t_3)$
ECA8 $Happens3(e, t_1, t_2) \wedge Terminates(e, f, t_1) \wedge t_2 < t_3 \wedge \neg Declipped(t_1, f, t_3) \rightarrow \neg HoldsAt(f, t_3)$
ECA9 $Happens3(e, t_1, t_2) \rightarrow t_1 \leqslant t_2$

Let ECA be the formula generated by conjoining axioms ECA5 through ECA9 and then expanding the predicates *InitiallyP*, *InitiallyN*, *Clipped*, and *Declipped* using definitions ECA1 through ECA4.

We can now proceed to our first result.

4. Lack of equivalence between TALA and ECA

In this section, we expose two differences between TALA and ECA. The first difference involves an occurrence of an event that initiates a fluent, followed by another occurrence of an event that initiates the same fluent. In TALA, the fluent is of indeterminate truth value within the durations of both event occurrences, whereas in ECA, it is only of indeterminate truth value within the duration of the first event occurrence. Within the duration of the second event occurrence, the fluent is true, because it was previously initiated and has not been clipped.

Theorem 1. TALA $\not\models$ ECA.

Proof. Consider the following structure S:

$$\mathbf{Initiates} = \{\langle \mathbf{I}, \mathbf{F}, 1\rangle, \langle \mathbf{I}, \mathbf{F}, 4\rangle\} \tag{6}$$

$$\mathbf{Terminates} = \emptyset \tag{7}$$

$$\mathbf{Happens3} = \{\langle \mathbf{I}, 1, 2\rangle, \langle \mathbf{I}, 4, 5\rangle\} \tag{8}$$

$$\mathbf{HoldsAt} = \{\langle \mathbf{F}, 3\rangle, \langle \mathbf{F}, 4\rangle, \langle \mathbf{F}, 6\rangle, \langle \mathbf{F}, 7\rangle, \langle \mathbf{F}, 8\rangle, \ldots\} \tag{9}$$

We can show $S \models$ TALA but $S \not\models$ ECA. It is straightforward to verify $S \models$ TALA. In order to show $S \not\models$ ECA, it is sufficient to show $S \not\models$ ECA7. Moreover, we need only show

$$S \not\models Happens3(I, 1, 2) \wedge Initiates(I, F, 1) \wedge 2 < 5 \wedge \neg Clipped(1, F, 5) \rightarrow HoldsAt(F, 5) \tag{10}$$

From (7), we have

$$S \models \neg\exists e, t_2, t_3 (Happens3(e, t_2, t_3) \wedge 1 < t_3 \wedge t_2 < 5 \wedge Terminates(e, F, t_2))$$

From this and definition ECA3, we have $S \models \neg Clipped(1, F, 5)$. From this, (8), and (6), we have

$$S \models Happens3(I, 1, 2) \wedge Initiates(I, F, 1) \wedge 2 < 5 \wedge \neg Clipped(1, F, 5)$$

But from (9), we have $S \models \neg HoldsAt(F, 5)$. Therefore, we have (10). □

The second difference between TALA and ECA involves an occurrence of an event that initiates a fluent, which overlaps in time an occurrence of an event that terminates the same fluent. In TALA, the fluent is true at the end of the initiating event occurrence, whereas in ECA, the fluent is of indeterminate truth value at the end of the initiating event occurrence.

Theorem 2. ECA $\not\models$ TALA.

Proof. Consider the following structure S:

$$\mathbf{Initiates} = \{\langle \mathbf{I}, \mathbf{F}, 1\rangle\} \tag{11}$$

$$\mathbf{Terminates} = \{\langle \mathbf{T}, \mathbf{F}, 2\rangle\} \tag{12}$$

$$\mathbf{Happens3} = \{\langle \mathbf{I}, 1, 2\rangle, \langle \mathbf{T}, 2, 3\rangle\} \tag{13}$$

$$\mathbf{HoldsAt} = \emptyset \tag{14}$$

We can show $S \models$ ECA but $S \not\models$ TALA. It is straightforward to verify $S \models$ ECA. In order to show $S \not\models$ TALA, it is sufficient to show $S \not\models$ TALA6. Furthermore, we need only show

$$S \not\models \mathit{Occurs}(1, 3, I) \wedge \mathit{Initiates}(I, F, 1) \rightarrow \mathit{Holds}(3, F) \tag{15}$$

From (13), definition TALA2, and (11), we have $\mathit{Occurs}(1, 3, I) \wedge \mathit{Initiates}(I, F, 1)$. But from (14) and definition TALA1, we have $S \models \neg \mathit{Holds}(3, F)$. Therefore, we have (15). □

Thus we have lack of equivalence.

Corollary 3. TALA *is not logically equivalent to* ECA.

Proof. This follows from either Theorem 1 or Theorem 2. □

5. ECB axiomatization

A second axiomatization of the event calculus that supports events with duration is provided by Miller and Shanahan [26, pp. 470–471]. After rewriting it in the style of ECA, it is as follows.

ECB1 $\mathit{Clipped}'(t_1, f, t_4) \stackrel{\text{def}}{\equiv} \exists e, t_2, t_3(\mathit{Happens3}(e, t_2, t_3) \wedge t_1 \leqslant t_3 \wedge t_2 < t_4 \wedge \mathit{Terminates}(e, f, t_2))$
ECB2 $\mathit{Declipped}'(t_1, f, t_4) \stackrel{\text{def}}{\equiv} \exists e, t_2, t_3(\mathit{Happens3}(e, t_2, t_3) \wedge t_1 \leqslant t_3 \wedge t_2 < t_4 \wedge \mathit{Initiates}(e, f, t_2))$
ECB3 $\mathit{Clipped}(t_1, f, t_4) \stackrel{\text{def}}{\equiv} \exists e, t_2, t_3(\mathit{Happens3}(e, t_2, t_3) \wedge t_1 < t_3 \wedge t_2 < t_4 \wedge \mathit{Terminates}(e, f, t_2))$
ECB4 $\mathit{Declipped}(t_1, f, t_4) \stackrel{\text{def}}{\equiv} \exists e, t_2, t_3(\mathit{Happens3}(e, t_2, t_3) \wedge t_1 < t_3 \wedge t_2 < t_4 \wedge \mathit{Initiates}(e, f, t_2))$
ECB5 $\mathit{HoldsAt}(f, t_1) \wedge t_1 < t_2 \wedge \neg \mathit{Clipped}'(t_1, f, t_2) \rightarrow \mathit{HoldsAt}(f, t_2)$
ECB6 $\neg \mathit{HoldsAt}(f, t_1) \wedge t_1 < t_2 \wedge \neg \mathit{Declipped}'(t_1, f, t_2) \rightarrow \neg \mathit{HoldsAt}(f, t_2)$
ECB7 $\mathit{Happens3}(e, t_1, t_2) \wedge \mathit{Initiates}(e, f, t_1) \wedge t_2 < t_3 \wedge \neg \mathit{Clipped}(t_1, f, t_3) \rightarrow \mathit{HoldsAt}(f, t_3)$
ECB8 $\mathit{Happens3}(e, t_1, t_2) \wedge \mathit{Terminates}(e, f, t_1) \wedge t_2 < t_3 \wedge \neg \mathit{Declipped}(t_1, f, t_3) \rightarrow \neg \mathit{HoldsAt}(f, t_3)$
ECB9 $\mathit{Happens3}(e, t_1, t_2) \rightarrow t_1 \leqslant t_2$

ECB differs from ECA in the following ways:

- It eliminates the definitions of *InitiallyP* and *InitiallyN*.
- It add definitions of *Clipped*′ and *Declipped*′.
- It adds the axioms of inertia ECB5 and ECB6.

Let ECB be the formula generated by conjoining axioms ECB5 through ECB9 and then expanding the predicates *Clipped*′, *Declipped*′, *Clipped*, and *Declipped* using definitions ECB1 through ECB4.
We then have our second result.

6. Lack of equivalence between TALA and ECB

ECB is not equivalent to TALA, for the two reasons previously given for ECA, as well as the following reason. Consider a single occurrence of an event that initiates a fluent. Within the duration of the event occurrence, the fluent is of indeterminate truth value in TALA, whereas in ECB a fluent that is true persists in each model because of the frame axiom ECB5. That is, if in a given model the fluent is true at any given timepoint within an event occurrence, then it is true for all remaining timepoints within the event occurrence.

Theorem 4. TALA $\not\models$ ECB.

Proof. Consider the following structure S:

$$\mathbf{Initiates} = \{\langle \mathbf{I}, \mathbf{F}, 1\rangle\} \tag{16}$$

$$\mathbf{Terminates} = \emptyset \tag{17}$$

$$\mathbf{Happens3} = \{\langle \mathbf{I}, 1, 3\rangle\} \tag{18}$$

$$\mathbf{HoldsAt} = \{\langle \mathbf{F}, 2\rangle, \langle \mathbf{F}, 4\rangle, \langle \mathbf{F}, 5\rangle, \langle \mathbf{F}, 6\rangle, \ldots\} \tag{19}$$

We can show $S \models$ TALA but $S \not\models$ ECB. It is straightforward to verify $S \models$ TALA. In order to show $S \not\models$ ECB, it is sufficient to show $S \not\models$ ECB5. Moreover, we need only show

$$S \not\models \mathit{HoldsAt}(F, 2) \wedge 2 < 3 \wedge \neg \mathit{Clipped}'(2, F, 3) \rightarrow \mathit{HoldsAt}(F, 3) \tag{20}$$

From (17), we have

$$S \models \neg \exists e, t_2, t_3 (\mathit{Happens3}(e, t_2, t_3) \wedge 2 \leqslant t_3 \wedge t_2 < 3 \wedge \mathit{Terminates}(e, F, t_2))$$

From this and definition ECB1, we have $S \models \neg \mathit{Clipped}'(2, F, 3)$. From this and (19), we have

$$S \models \mathit{HoldsAt}(F, 2) \wedge 2 < 3 \wedge \neg \mathit{Clipped}'(2, F, 3)$$

But from (19), we have $S \models \neg \mathit{HoldsAt}(F, 3)$. Therefore, we have (20). □

Corollary 5. TALA *is not logically equivalent to* ECB.

Proof. This follows from Theorem 4. □

7. General action type specifications

We have shown that, if TAL action type specifications are of the form (1), then TAL and the event calculus are not equivalent. What if we use a more general form of action type specification? Can we restore equivalence with the event calculus?

The answer is no. It is sufficient to show that the difference highlighted in the proof of Theorem 2 cannot be erased. ECA7 (as well as ECB7, which is identical) entails that, if a fluent is not clipped during an event occurrence that initiates the fluent, then the fluent is true at the end of the event occurrence. Action type specifications of the form (1) entail that the fluent is always true at the end of the event. We would like to add to the action type specification the condition that the fluent is not clipped:

$$[t_1, t_2] I \rightsquigarrow \neg \exists t_3, t_4 [[t_3, t_4] T \wedge t_1 < t_4 - 1 \wedge t_3 < t_2] \rightarrow R([t_2] F)$$

(I and T are actions, and F is a fluent.) Unfortunately, the consequent of an action type specification cannot contain action occurrence statements such as $[t_3, t_4]T$ [4, p. 28].

The underlying difficulty is that TAL does not have the notions of clipped and declipped. In order for TAL to emulate the event calculus with events with duration, it would have to be extended with these notions.

8. Restriction to single-step actions

We now consider what happens if we restrict TAL and the event calculus to single-step actions. We add TALA9 to TALA:

TALA1 $\mathit{Holds}(t, f) \stackrel{\text{def}}{\equiv} \mathit{HoldsAt}(f, t)$

TALA2 $\mathit{Occurs}(t_1, t_2, e) \stackrel{\text{def}}{\equiv} \mathit{Happens3}(e, t_1, t_2 - 1)$

TALA3 $Occlude(t, f) \stackrel{\text{def}}{\equiv} \exists e, t_1, t_2((Initiates(e, f, t_1) \vee Terminates(e, f, t_1)) \wedge Occurs(t_1, t_2, e) \wedge t_1 < t \wedge t \leqslant t_2)$
TALA4 $Holds(t, f) \wedge \neg Occlude(t+1, f) \rightarrow Holds(t+1, f)$
TALA5 $\neg Holds(t, f) \wedge \neg Occlude(t+1, f) \rightarrow \neg Holds(t+1, f)$
TALA6 $Occurs(t_1, t_2, e) \wedge Initiates(e, f, t_1) \rightarrow Holds(t_2, f)$
TALA7 $Occurs(t_1, t_2, e) \wedge Terminates(e, f, t_1) \rightarrow \neg Holds(t_2, f)$
TALA8 $Occurs(t_1, t_2, e) \rightarrow t_1 < t_2$
TALA9 $Occurs(t_1, t_2, e) \rightarrow t_2 = t_1 + 1$

Let TALAS be the formula generated by conjoining axioms TALA4 through TALA9 and then expanding the predicates *Holds*, *Occurs*, and *Occlude* using definitions TALA1 through TALA3.

For single-step actions, the version of the event calculus that appears to be the most similar to TAL is the discrete event calculus (DEC) [27,29]. DEC was developed to improve the efficiency of automated reasoning in the event calculus. It improves efficiency by limiting time to the integers, and eliminating triply quantified time from many of the axioms. Mueller [27] proves that, for integer time and single-step events, DEC is logically equivalent to an extended version of ECB [26].

The following axiomatization is obtained from the full version of DEC by eliminating *Trajectory*, *AntiTrajectory*, *Releases*, and *ReleasedAt*.

DECA1 $Happens(e, t) \stackrel{\text{def}}{\equiv} Happens3(e, t, t)$
DECA2 $HoldsAt(f, t) \wedge \neg\exists e(Happens(e, t) \wedge Terminates(e, f, t)) \rightarrow HoldsAt(f, t+1)$
DECA3 $\neg HoldsAt(f, t) \wedge \neg\exists e(Happens(e, t) \wedge Initiates(e, f, t)) \rightarrow \neg HoldsAt(f, t+1)$
DECA4 $Happens(e, t) \wedge Initiates(e, f, t) \rightarrow HoldsAt(f, t+1)$
DECA5 $Happens(e, t) \wedge Terminates(e, f, t) \rightarrow \neg HoldsAt(f, t+1)$
DECA6 $Happens3(e, t_1, t_2) \rightarrow t_1 = t_2$

Let DECA be the formula generated by conjoining axioms DECA2 through DECA6 and then expanding the predicate *Happens* using definition DECA1.

We then obtain our final result.

9. Equivalence of TALAS and DECA

We can show that TALAS and DECA are logically equivalent. First, we prove a number of lemmas.

Lemma 6. TALAS $\models$ DECA2.

Proof. Suppose TALAS. Let f be an arbitrary fluent and t be an arbitrary timepoint. We must show $HoldsAt(f, t) \wedge \neg\exists e(Happens(e, t) \wedge Terminates(e, f, t)) \rightarrow HoldsAt(f, t+1)$. Suppose

$$HoldsAt(f, t) \tag{21}$$

$$\neg\exists e(Happens(e, t) \wedge Terminates(e, f, t)) \tag{22}$$

We consider two cases.

Case 1: $\exists e(Happens(e, t) \wedge Initiates(e, f, t))$.

From definition DECA1, TALA2, and existential instantiation, we get $Occurs(t, t+1, E) \wedge Initiates(E, f, t)$ for some E. From this, TALA6, and TALA1, we have $HoldsAt(f, t+1)$, as required.

Case 2: $\neg\exists e(Happens(e, t) \wedge Initiates(e, f, t))$.

From (22), definition DECA1, and TALA2, we get

$$\neg\exists e((Initiates(e, f, t) \vee Terminates(e, f, t)) \wedge Occurs(t, t+1, e))$$

Therefore,

$$t_1 = t \wedge t_2 = t_1 + 1 \rightarrow \neg((Initiates(e, f, t_1) \vee Terminates(e, f, t_1)) \wedge Occurs(t_1, t_2, e)) \tag{23}$$

From TALA9, by contraposition, we get $t_2 \neq t_1 + 1 \rightarrow \neg Occurs(t_1, t_2, e)$. Hence,

$$t_2 \neq t_1 + 1 \rightarrow \neg((Initiates(e, f, t_1) \vee Terminates(e, f, t_1)) \wedge Occurs(t_1, t_2, e))$$

From this and (23), we have

$$t_2 \neq t_1 + 1 \vee t_1 = t \rightarrow \neg((Initiates(e, f, t_1) \vee Terminates(e, f, t_1)) \wedge Occurs(t_1, t_2, e)) \tag{24}$$

We can show

$$t_1 < t + 1 \wedge t + 1 \leqslant t_2 \rightarrow t_2 \neq t_1 + 1 \vee t_1 = t \tag{25}$$

which is logically equivalent to $t_1 < t + 1 \wedge t + 1 \leqslant t_2 \wedge t_2 = t_1 + 1 \rightarrow t_1 = t$. To see this, suppose

$$t_1 < t + 1 \tag{26}$$

$$t + 1 \leqslant t_2 \tag{27}$$

$$t_2 = t_1 + 1 \tag{28}$$

From (27) and (28), we get $t \leqslant t_1$. From this and (26), we have $t_1 = t$, as required.

From (25) and (24), we have

$$t_1 < t + 1 \wedge t + 1 \leqslant t_2 \rightarrow \neg((Initiates(e, f, t_1) \vee Terminates(e, f, t_1)) \wedge Occurs(t_1, t_2, e)) \tag{29}$$

From this and TALA3, we get $\neg Occlude(t + 1, f)$. From this, (21), TALA1, and TALA4, we have $HoldsAt(f, t + 1)$, as required. □

Lemma 7. TALAS $\models$ DECA3.

Proof. The proof is identical to that of Lemma 6, except that $\neg HoldsAt$ is substituted for *HoldsAt*, *Initiates* and *Terminates* are swapped, TALA7 is substituted for TALA6, and TALA5 is substituted for TALA4. □

Lemma 8. TALAS $\models$ DECA4.

Proof. Suppose TALAS. Let e be an arbitrary event, f be an arbitrary fluent, and t be an arbitrary timepoint. We must show $Happens(e, t) \wedge Initiates(e, f, t) \rightarrow HoldsAt(f, t + 1)$. Suppose $Happens(e, t) \wedge Initiates(e, f, t)$. From $Happens(e, t)$, definition DECA1, and TALA2, we get $Occurs(t, t + 1, e)$. From this, $Initiates(e, f, t)$, TALA6, and TALA1, we have $HoldsAt(f, t + 1)$. □

Lemma 9. TALAS $\models$ DECA5.

Proof. The proof is identical to that of Lemma 8, except that $\neg HoldsAt$ is substituted for *HoldsAt*, *Terminates* is substituted for *Initiates*, and TALA7 is substituted for TALA6. □

Lemma 10. TALAS $\models$ DECA6.

Proof. This follows from TALA9 and TALA2. □

Now we consider the other direction.

Lemma 11. DECA $\models$ TALA4.

Proof. Suppose DECA. Let f be an arbitrary fluent and t be an arbitrary timepoint. We must show $Holds(t, f) \wedge \neg Occlude(t + 1, f) \rightarrow Holds(t + 1, f)$. Suppose

$$Holds(t, f) \tag{30}$$

$$\neg Occlude(t + 1, f) \tag{31}$$

From (31) and definition TALA3, we have

$$\neg\exists e, t_1, t_2((\mathit{Initiates}(e, f, t_1) \vee \mathit{Terminates}(e, f, t_1)) \wedge \mathit{Occurs}(t_1, t_2, e) \wedge t_1 < t+1 \wedge t+1 \leqslant t_2)$$

Hence, $\neg\exists e(\mathit{Terminates}(e, f, t) \wedge \mathit{Occurs}(t, t+1, e))$. From this, definition TALA2, and DECA1, we get $\neg\exists e$ $(\mathit{Happens}(e, t) \wedge \mathit{Terminates}(e, f, t))$. From this, (30), definition TALA1, and DECA2, we have $\mathit{Holds}(t+1, f)$, as required. □

Lemma 12. DECA $\models$ TALA5.

Proof. The proof is identical to that of Lemma 11, except that $\neg$*Holds* is substituted for *Holds*, *Initiates* and *Terminates* are swapped, and DECA3 is substituted for DECA2. □

Lemma 13. DECA $\models$ TALA6.

Proof. Suppose DECA. Let e be an arbitrary event, f be an arbitrary fluent, and t_1 and t_2 be arbitrary timepoints. We must show $\mathit{Occurs}(t_1, t_2, e) \wedge \mathit{Initiates}(e, f, t_1) \rightarrow \mathit{Holds}(t_2, f)$. Suppose $\mathit{Occurs}(t_1, t_2, e) \wedge \mathit{Initiates}(e, f, t_1)$. From $\mathit{Occurs}(t_1, t_2, e)$ and definition TALA2, we have $\mathit{Happens3}(e, t_1, t_2 - 1)$. From this and DECA6, we get $t_2 = t_1 + 1$. From this, $\mathit{Happens3}(e, t_1, t_2 - 1)$, and DECA1, we have $\mathit{Happens}(e, t_1)$. From this, $\mathit{Initiates}(e, f, t_1)$, DECA4, and definition TALA1, we get $\mathit{Holds}(t_1 + 1, f)$. From this and $t_2 = t_1 + 1$, we have $\mathit{Holds}(t_2, f)$. □

Lemma 14. DECA $\models$ TALA7.

Proof. The proof is identical to that of Lemma 13, except that $\neg$*Holds* is substituted for *Holds*, *Terminates* is substituted for *Initiates*, and DECA5 is substituted for DECA4. □

Lemma 15. DECA $\models$ TALA8.

Proof. This follows from definition TALA2 and DECA6. □

Lemma 16. DECA $\models$ TALA9.

Proof. This follows from definition TALA2 and DECA6. □

Now we proceed to the equivalence theorem.

Theorem 17. TALAS *is logically equivalent to* DECA.

Proof. We prove the two directions separately.

(TALAS $\models$ DECA) Suppose TALAS. Then DECA2, DECA3, DECA4, DECA5, and DECA6 follow from Lemmas 6, 7, 8, 9, and 10, respectively.

(DECA $\models$ TALAS) Suppose DECA. Then TALA4, TALA5, TALA6, TALA7, TALA8, and TALA9 follow from Lemmas 11, 12, 13, 14, 15, and 16, respectively. □

10. Conclusions

We have investigated the relationship between the event calculus and TAL. We started by restricting both formalisms to their common features, and found that the resulting versions of the formalisms are not equivalent. We then further restricted the event calculus and TAL to single-step actions, and proved that these versions are logically equivalent.

Some areas for further work are the following:

- Lesser restrictions than the restriction to single-step actions could be explored. It may be possible to show a form of equivalence between TAL and the event calculus with events with duration if there are no overlapping events.

- Hybrids of the event calculus and TAL could be created.
- The relationships between other pairs of formalisms for reasoning about action and change could be explored. Correspondences could be developed between the event calculus and action language $\mathcal{C}+$, and between the situation calculus and temporal action logics.

Acknowledgements

I am grateful to three anonymous referees for helpful suggestions.

Appendix A. Translation from TAL 1.0 $\mathcal{L}(ND)$ to $\mathcal{L}(FL)$

The important translation rules from TAL 1.0 $\mathcal{L}(ND)$ to $\mathcal{L}(FL)$ are as follows [5, pp. 6–9]:

- If α is an action occurrence statement, then $\mathit{Trans}(\alpha \rightsquigarrow \beta) = \mathit{Trans}(\alpha) \rightarrow \mathit{Trans}(\beta)$. If disjunction ($\vee$) is used in β, it must be in the scope of an R, I, or X reassignment operator so that circumscription of *Occlude* can be computed using predicate completion [22].
- If α is an action, then $\mathit{Trans}([\tau, \tau']\alpha) = \mathit{Occurs}(\tau, \tau', \alpha)$.
- $\mathit{Trans}(R((\tau, \tau']\alpha)) = \mathit{Trans}(X((\tau, \tau']\alpha) \wedge [\tau']\alpha)$.
- $\mathit{Trans}(R([\tau]\alpha)) = \mathit{Trans}(X([\tau]\alpha) \wedge [\tau]\alpha)$.
- $\mathit{Trans}(I((\tau, \tau']\alpha)) = \mathit{Trans}(X((\tau, \tau']\alpha) \wedge (\tau, \tau']\alpha)$.
- $\mathit{Trans}(X((\tau, \tau']\alpha)) = \forall t(\tau < t \wedge t \leqslant \tau' \rightarrow \mathit{Trans}(X([t]\alpha)))$.
- $\mathit{Trans}(X([\tau]\neg\alpha)) = \mathit{Trans}(X([\tau]\alpha))$.
- If $\otimes \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then $\mathit{Trans}(X([\tau]\alpha \otimes \beta)) = \mathit{Trans}(X([\tau]\alpha)) \wedge \mathit{Trans}(X([\tau]\beta))$.
- If $\mathcal{Q} \in \{\exists, \forall\}$, then $\mathit{Trans}(X([\tau]\mathcal{Q}\nu[\alpha])) = \mathcal{Q}\nu(\mathit{Trans}(X([\tau]\alpha)))$.
- If α is a fluent, then $\mathit{Trans}(X([\tau]\alpha)) = \mathit{Occlude}(\tau, \alpha)$.
- $\mathit{Trans}([\tau]\neg\alpha) = \neg\mathit{Trans}([\tau]\alpha)$.
- If $\otimes \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, then $\mathit{Trans}([\tau]\alpha \otimes \beta) = \mathit{Trans}([\tau]\alpha) \otimes \mathit{Trans}([\tau]\beta)$.
- If $\mathcal{Q} \in \{\exists, \forall\}$, then $\mathit{Trans}([\tau]\mathcal{Q}\nu[\alpha]) = \mathcal{Q}\nu(\mathit{Trans}([\tau]\alpha))$.
- If α is a fluent, then $\mathit{Trans}([\tau]\alpha) = \mathit{Holds}(\tau, \alpha)$.

References

[1] A.B. Baker, Nonmonotonic reasoning in the framework of situation calculus, Artificial Intelligence 49 (1–3) (1991) 5–23.
[2] B. Bennett, A.P. Galton, A unifying semantics for time and events, Artificial Intelligence 153 (1–2) (2004) 13–48.
[3] P. Clark, B. Porter, KM—The Knowledge Machine 1.4.0: KM's situation mechanism, Department of Computer Science, University of Texas at Austin, Austin, TX, 2000.
[4] P. Doherty, PMON$^+$: A fluent logic for action and change, Formal specification, version 1.0, Technical Report LiTH-IDA-R-96-33, Department of Computer and Information Science, Linköping University, Linköping, Sweden, 1996.
[5] P. Doherty, J. Gustafsson, L. Karlsson, J. Kvarnström, TAL: Temporal Action Logics language specification and tutorial, Linköping Electronic Articles in Computer and Information Science 3 (015) (1998).
[6] P. Doherty, W. Łukaszewicz, Circumscribing features and fluents, in: D.M. Gabbay, H.J. Ohlbach (Eds.), Temporal Logic, in: Lecture Notes in Computer Science, vol. 827, Springer, Berlin, 1994, pp. 82–100.
[7] R.E. Fikes, N.J. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (1971) 189–208.
[8] M. Gelfond, V. Lifschitz, Representing action and change by logic programs, Journal of Logic Programming 17 (2–4) (1993) 301–321.
[9] E. Giunchiglia, J. Lee, V. Lifschitz, N.C. McCain, H. Turner, Nonmonotonic causal theories, Artificial Intelligence 153 (2004) 49–104.
[10] E. Giunchiglia, V. Lifschitz, An action language based on causal explanation: Preliminary report, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Conference on Innovative Applications of Artificial Intelligence, AAAI Press, Menlo Park, CA, 1998, pp. 623–630.
[11] E. Giunchiglia, V. Lifschitz, Action languages, temporal action logics and the situation calculus, Linköping Electronic Articles in Computer and Information Science 4 (040) (1999).
[12] S. Hölldobler, J. Schneeberger, A new deductive approach to planning, New Generation Computing 8 (3) (1990) 225–244.
[13] A.C. Kakas, R. Miller, A simple declarative language for describing narratives with actions, Journal of Logic Programming 31 (1–3) (1997) 157–200.
[14] A.C. Kakas, R. Miller, F. Toni, An argumentation framework for reasoning about actions and change, in: M. Gelfond, N. Leone, G. Pfeifer (Eds.), Proceedings of the Fifth International Conference on Logic Programming and Nonmonotonic Reasoning, in: Lecture Notes in Computer Science, vol. 1730, Springer, Berlin, 1999, pp. 78–91.

[15] A.C. Kakas, R. Miller, F. Toni, E-RES—A system for reasoning about actions, events and observations, in: Proceedings of the Eighth International Workshop on Non-Monotonic Reasoning, 2000.
[16] G. Neelakantan Kartha, Soundness and completeness theorems for three formalizations of action, in: R. Bajcsy (Ed.), Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1993, pp. 724–729.
[17] R.A. Kowalski, F. Sadri, The situation calculus and event calculus compared, in: M. Bruynooghe (Ed.), Logic Programming: The 1994 International Symposium, MIT Press, Cambridge, MA, 1994, pp. 539–553.
[18] R.A. Kowalski, F. Sadri, Reconciling the event calculus with the situation calculus, Journal of Logic Programming 31 (1–3) (1997) 39–58.
[19] R.A. Kowalski, M.J. Sergot, A logic-based calculus of events, New Generation Computing 4 (1) (1986) 67–95.
[20] J. Kvarnström, VITAL: Visualization and implementation of temporal action logic, Department of Computer and Information Science, Linköping University, Linköping, Sweden, 2001.
[21] J. Kvarnström, TALplanner and other extensions to temporal action logic, PhD thesis, Department of Computer and Information Science, Linköping University, Linköping, Sweden, 2005.
[22] V. Lifschitz, Circumscription, in: D.M. Gabbay, C.J. Hogger, J.A. Robinson (Eds.), Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning, Oxford University Press, Oxford, 1994, pp. 298–352.
[23] N.C. McCain, Causality in commonsense reasoning about actions, Technical Report CS-TR-97-25, Department of Computer Sciences, University of Texas at Austin, Austin, TX, 1997.
[24] J. McCarthy, Circumscription—a form of non-monotonic reasoning, Artificial Intelligence 13 (1–2) (1980) 27–39.
[25] J. McCarthy, P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence 4, Edinburgh University Press, Edinburgh, 1969, pp. 463–502.
[26] R. Miller, M. Shanahan, Some alternative formulations of the event calculus, in: A.C. Kakas, F. Sadri (Eds.), Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski, Part II, in: Lecture Notes in Computer Science, vol. 2408, Springer, Berlin, 2002, pp. 452–490.
[27] E.T. Mueller, Event calculus reasoning through satisfiability, Journal of Logic and Computation 14 (5) (2004) 703–730.
[28] E.T. Mueller, A tool for satisfiability-based commonsense reasoning in the event calculus, in: V. Barr, Z. Markov (Eds.), Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, Menlo Park, CA, 2004, pp. 147–152.
[29] E.T. Mueller, Commonsense Reasoning, Morgan Kaufmann, San Francisco, CA, 2006.
[30] E.P.D. Pednault, ADL: Exploring the middle ground between STRIPS and the situation calculus, in: R.J. Brachman, H.J. Levesque, R. Reiter (Eds.), Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, CA, 1989, pp. 324–332.
[31] R. Reiter, The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression, in: V. Lifschitz (Ed.), Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy, Academic Press, San Diego, CA, 1991, pp. 359–380.
[32] R. Reiter, Proving properties of states in the situation calculus, Artificial Intelligence 64 (2) (1993) 337–351.
[33] R. Reiter, Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, MIT Press, Cambridge, MA, 2001.
[34] E. Sandewall, Filter preferential entailment for the logic of action in almost continuous worlds, in: N.S. Sridharan (Ed.), Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1989, pp. 894–899.
[35] E. Sandewall, Features and Fluents: The Representation of Knowledge about Dynamical Systems, vol. I, Oxford University Press, Oxford, 1994.
[36] M. Shanahan, Solving the Frame Problem, MIT Press, Cambridge, MA, 1997.
[37] M. Shanahan, The event calculus explained, in: M.J. Wooldridge, M.M. Veloso (Eds.), Artificial Intelligence Today: Recent Trends and Developments, in: Lecture Notes in Computer Science, vol. 1600, Springer, Berlin, 1999, pp. 409–430.
[38] M. Shanahan, The ramification problem in the event calculus, in: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, CA, 1999, pp. 140–146.
[39] M. Shanahan, An abductive event calculus planner, Journal of Logic Programming 44 (1–3) (2000) 207–240.
[40] M. Thielscher, An analysis of systematic approaches to reasoning about actions and change, in: P. Jorrand, V. Sgurev (Eds.), Proceedings of the Sixth International Conference on Artificial Intelligence: Methodology, Systems, Applications, World Scientific, River Edge, NJ, 1994, pp. 195–204.
[41] M. Thielscher, From situation calculus to fluent calculus: State update axioms as a solution to the inferential frame problem, Artificial Intelligence 111 (1999) 277–299.
[42] M. Thielscher, The qualification problem: A solution to the problem of anomalous models, Artificial Intelligence 131 (2001) 1–37.
[43] M. Thielscher, Reasoning Robots: The Art and Science of Programming Robotic Agents, Springer, Dordrecht, 2005.
[44] K. Van Belleghem, M. Denecker, D. De Schreye, On the relation between situation calculus and event calculus, Journal of Logic Programming 31 (1–3) (1997) 3–37.