Variants of the Event Calculus

2 authors:

Fariba Sadri
Imperial College London
114 PUBLICATIONS **2,761** CITATIONS

Robert Kowalski
Imperial College London
154 PUBLICATIONS **12,446** CITATIONS

Variants of the Event Calculus

ICLP 95

Fariba Sadri and Robert Kowalski
Department of Computing, Imperial College of Science, Technology and Medicine, 180, Queens Gate, London SW7 2BZ
fs@doc.ic.ac.uk rak@doc.ic.ac.uk

Abstract

The event calculus was proposed as a formalism for reasoning about time and events. Through the years, however, a much simpler variant (SEC) of the original calculus (EC) has proved more useful in practice.

We argue that EC has the advantage of being more general than SEC, but the disadvantage of being too complex and in some cases erroneous. SEC has the advantage of simplicity, but the disadvantage of being too specialised.

This paper has two main objectives. The first is to show the formal relationship between the two calculi. The second is to propose a new variant (NEC) of the event calculus, which is essentially SEC in iff-form augmented with integrity constraints, and to argue that NEC combines the generality of EC with the simplicity of SEC.

We argue that NEC also demonstrates the more general potential of using theories consisting of iff-definitions and integrity constraints as a new logic programming paradigm.

1 Introduction

The original event calculus (EC) [10] was formulated as a logic program for representing and reasoning about the occurrence of events, the properties that events initiate and terminate, and the maximal time periods for which those properties hold. It also contained rules to derive the existence of implied events from incomplete information about explicitly given events.

Subsequent to the original paper [10], most further development of the EC focused on a variant [15, 7, 5, 16, 4] which employed time points instead of time periods. This simplified event calculus (SEC) was applied to such problems as database updates [7], planning [5, 12], explanation [16], hypothetical reasoning [14] and air traffic management [17]. A further special case of SEC, where time points are identified with global situations, has been shown to be equivalent to the situation calculus with induction [9].

In this paper we present two sets of results which help to explain why SEC has replaced EC in practice and to justify our proposal for NEC. The first shows that, in the special case in which all event occurrences are explicitly and completely given, the two calculi derive equivalent consequences about what properties hold at what time points. The second shows that in those cases where information about explicitly given events is incomplete the if-and-only-if form (iff-form) of SEC, augmented with appropriate integrity constraints, can be used, in place of EC, to derive the existence of implied events. We call this new variant of SEC the new event calculus (NEC). We also show that, in certain cases of incomplete information about events, both EC and SEC give incorrect results, which are avoided in NEC.

To prove our first set of results, we use the Clark completion [2] (i.e. iff-form together with the Clark equality theory) as the semantics of the EC and SEC logic programs. To prove our second set of results about NEC, we use the completion augmented with integrity constraints. Elsewhere, we are investigating other uses of the completion with integrity constraints as a computational paradigm [8, 6, 11].

The remainder of the paper is structured as follows. In sections 2 and 3 we introduce and discuss SEC and EC, respectively. In section 4 we discuss some of the shortcomings of EC. In section 5 we show how EC and SEC are formally related. In sections 6 and 7 we introduce NEC and argue that it subsumes most of the functionality of EC.

2 The simplified event calculus (SEC)

2.1 Axioms

Throughout this paper we use the notational conventions that variables start in the upper case and that constant, function and predicate symbols start in the lower case. All variables, which are not otherwise explicitly quantified, are implicitly bound by universal quantifiers in front of the formula in which they occur.

SEC consists of one core axiom, S, and any number of auxiliary domain dependent definitions. The core axiom in conventional logic programming if-form is:

$$\begin{aligned}\text{holdsAt}(P, T) \leftarrow\ & \text{happens}(E1, T1) \wedge \text{initiates}(E1, P) \wedge T1 < T \wedge \\ & \neg \exists E2, T2\ [\ \text{happens}(E2, T2) \wedge \text{terminates}(E2, P) \wedge T1 < T2 \wedge T2 < T] \qquad S\end{aligned}$$

This states that a property P which is initiated by an event *persists* until it is terminated by a subsequent event.

In addition to S, further axioms are needed to define the <, *initiates*, *terminates* and *happens* predicates. The exact definition of < is not directly relevant for the purposes of this paper, provided the definition satisfies such integrity constraints as transitivity and anti-symmetry.

Domain dependent axioms are needed to define the *initiates* and *terminates* predicates. For example:

initiates(E, has(X, Y)) ← act(E, give(Z, Y, X))
initiates(E, has(X, Y)) ← act(E, steal(X, Y, Z))
terminates(E, has(Z, Y)) ← act(E, give(Z, Y, X))
terminates(E, has(Z, Y)) ← act(E, steal(X, Y, Z))

which state that the property of possession is initiated and terminated by giving and stealing events.

Note that in this formulation of SEC the event variables represent event tokens, i.e. occurrences of events. Some other formulations [e.g. 9] use event types instead of tokens. We employ the event token formulation in this paper to facilitate the comparison with EC later.

Problem dependent axioms define event occurrences, e.g.:

happens(e1, 1) act(e1, give(bob, book, mary))
happens(e2, 10) act(e2, give(mary, book, tom))

If we assume that such axioms provide a complete description of event happenings, then their semantics can be expressed in iff-form, e.g.:

$$happens(X, Y) \leftrightarrow [(X = e1 \wedge Y = 1) \vee (X = e2 \wedge Y = 10)]$$

$$act(X, Y) \leftrightarrow [(X = e1 \wedge Y = give(bob, book, mary)) \vee (X = e2 \wedge Y = give(mary\ book, tom))]$$

Given such axioms, and the completion of *terminates* and <, it is possible to derive conclusions of the form *holdsAt(p, t)* for concrete properties p and specific time points t. However, it is also possible to derive more general conclusions about what properties hold for what time intervals.

2.2 Derivation of properties holding over time intervals

Suppose we are given a complete history of all the events that initiate or terminate a given property p. Pictorially:

```
e1     e2      e3     e4               e2i-1      e2i     en
o----------o   o---------o     ...      o-----------o  ... o
t1     t2      t3     t4               t2i-1      t2i     tn
```

where, for each $i \geq 1$, e_i happens at time t_i, $t_i < t_{i+1}$ and e_i initiates p if i is odd and terminates p if i is even. Then, assuming a total ordering on time points, the completion of S together with the completed definitions of the *initiates*, *terminates* and *happens* predicates implies:

$$holdsAt(p,T) \leftrightarrow [\ t_1 < T \leq t_2 \ \vee\ t_3 < T \leq t_4 \ \vee \ldots\ t_{2i-1} < T \leq t_{2i} \ \vee \ldots\ t_n < T] \qquad S^*$$

in case e_n initiates p. If e_n terminates p, the last condition is replaced by $t_{n-1} < T \leq t_n$, where $T \leq t_k$ abbreviates $T < t_k \vee T = t_k$.

Literally, S* defines when a property holds at a time point. However, if the inequality predicate is treated as a constraint in a constraint logic programming context, the use of S* simulates reasoning about properties holding for maximal time intervals.

2.3 SEC may give incorrect results

SEC was formulated to deal with complete information about events. It can give incorrect results if event information is incomplete.

Example: Suppose that event e1 at time 1 is an act of Bob giving a book to Mary, and e2 at time 10 is an act of John giving the book to Tom. Given the descriptions of these events, the definitions of *initiates* and *terminates* and S, and using the completion of such definitions, we can derive

holdsAt(has(mary, book), T) $\leftrightarrow$ 1< T
holdsAt(has(tom, book), T) $\leftrightarrow$ 10< T

which imply incorrectly that after time 10 both Mary and Tom have the book, and that at no time at all does John have the book.

The first incorrect implication can be corrected by revising the definition of *terminates*:

terminates(E, has(Z', Y)) ← act(E, give(Z, Y, X)) ∧ ¬ Z'= X
terminates(E, has(Z', Y)) ← act(E, steal(X, Y, Z)) ∧ ¬ Z'= X

Correcting the second incorrect implication is both more difficult and more interesting. We shall explain how it can be corrected by using NEC in section 6.

3 The original event calculus (EC)

One of the main reasons for the greater complexity of EC compared with SEC is that EC attempts to deal with incomplete information. Another is that its vocabulary is more complex, arguably because its ontology is concerned primarily with maximal time periods rather than with time points. Below we present the domain-independent axioms of EC.

holds(after(E, P)) ← initiates(E, P) O1
holds(before(E, P)) ← terminates(E, P) O2
start(after(E, P), E) O3
end(before(E, P), E) O4
start(before(E2, P), E1) ← equal(after(E1, P), before(E2, P)) O5
end(after(E1, P), E2) ← equal(after(E1, P), before(E2, P)) O6
equal(after(E1, P), before(E2, P)) ← holds(after(E1, P)) ∧
    holds(before(E2, P)) ∧ E1 < E2 ∧ ¬ broken(E1, P, E2) O7
broken(E1, P, E2) ← holds(after(E*, P*)) ∧ exclusive(P, P*) ∧
    E1 < E* ∧ E* < E2 O8
broken(E1, P, E2) ← holds(before(E*,P*)) ∧ exclusive(P, P*) ∧
    E1 < E* ∧ E* < E2 O9
exclusive(P, P) O10
exclusive(P, P*) ← incompatible(P, P*) O11
holdsAt(P, T) ← holds(after(E, P)) ∧ in(T, after(E, P)) O12
holdsAt(P, T) ← holds(before(E, P)) ∧ in(T, before(E, P)) O13
in(T, Period) ← start(Period, E1) ∧ end(Period, E2) ∧ time(E1, T1) ∧
    time(E2, T2) ∧ T1< T ∧ T< T2 O14
in(T, Period) ← start(Period, E1) ∧ time(E1, T1) ∧ T1 < T ∧
    ¬ end(Period, E2) O15
[start(before(E2, P2), init(before(E2, P2))) ∧
lequal(E1, init(before(E2, P2)))] ←
    holds(before(E1, P1)) ∧ holds(before(E2, P2)) ∧
    exclusive(P1, P2) ∧ E1 < E2 ∧ ¬ broken(E1, P2, E2) O16

$$
\begin{aligned}
&[\mathrm{end}(\mathrm{after}(E1, P1),\ \mathrm{fin}(\mathrm{after}(E1, P1))) \wedge \\
&\ \mathrm{lequal}(\mathrm{fin}(\mathrm{after}(E1, P1)),\ E2)] \leftarrow \\
&\qquad\qquad \mathrm{holds}(\mathrm{after}(E1, P1)) \wedge\ \mathrm{holds}(\mathrm{after}(E2, P2)) \wedge \\
&\qquad\qquad \mathrm{exclusive}(P1, P2) \wedge\ E1 < E2 \wedge \neg\, \mathrm{broken}(E1, P1, E2) \qquad \text{O17}
\end{aligned}
$$

$$
\begin{aligned}
&[\mathrm{end}(\mathrm{after}(E1, P1),\ \mathrm{fin}(\mathrm{after}(E1, P1))) \wedge \\
&\ \mathrm{start}(\mathrm{before}(E2, P2),\ \mathrm{init}(\mathrm{before}(E2, P2))) \wedge \\
&\ \mathrm{lequal}(\mathrm{fin}(\mathrm{after}(E1, P1)),\ \mathrm{init}(\mathrm{before}(E2, P2)))\ ] \leftarrow \\
&\qquad \mathrm{holds}(\mathrm{after}(E1, P1)) \wedge\ \mathrm{holds}(\mathrm{before}(E2, P2)) \wedge \\
&\qquad \mathrm{incompatible}(P1, P2) \wedge\ E1 < E2 \wedge \neg\, \mathrm{broken}(E1, P1, E2) \qquad \text{O18}
\end{aligned}
$$

O1-O4: An event E initiating a property P *starts* a maximal period, named *after(E, P)*, during which P holds. Similarly, an event E terminating P *ends* a maximal period *before(E, P)* during which P holds. The predicates *initiates* and *terminates* are defined by domain-dependent axioms as in SEC, with *act* given as part of the problem-specific input.

O5-O9: Because one event can initiate a property and a subsequent event can terminate it, the same period can be named both as an *after(E1, P)* and as a *before(E2, P)* period. It is necessary to determine, therefore, when two such periods are equal. This is the case (by default) when there is no event E* which happens between the two events and breaks the holding of P by initiating or terminating some property P* which excludes P. The equality of two periods *after(E1, P)* and *before(E2, P)* allows us to derive an end for the first period and a start for the second.

Note that in [10] the symbol = was used instead of the predicate *equal*. We use *equal* here to avoid confusion when we consider the completion of EC where we use = as the identity predicate. The Clark equality theory, CET [2], holds for =, but not for *equal*.

O10-O11: Properties P and P* are *exclusive* if they are identical or incompatible with one another. Incompatibility is defined by domain dependent rules, such as

$$\mathrm{incompatible}(\mathrm{has}(X, Y),\ \mathrm{has}(Z\ Y)) \leftarrow \neg\, X{=}Z.$$

O12-O13: A property holds at a time point when the time point is in a time period for which the property holds.

O14-O15: A time point is in a period if it is between the start and end of the period, or when the period has no end and the time point is after its start.

O16-O18: These axioms derive implied events from incomplete event descriptions. This ensures that periods are maximal and that incompatible properties do not hold for overlapping periods.

Axiom O16 deals with the case

```
P1          E1              P2          E2
<----------o                <---------o
```

where two events terminate exclusive properties, and no explicitly given event breaks the holding of P2 between the two events. The axiom derives an implicit event which initiates P2 and occurs after E1 or is the same as E1.

lequal is defined as

$$lequal(X, Y) \leftarrow X < Y$$
$$lequal(X, Y) \leftarrow equal(X, Y)$$

We use the predicate *lequal* instead of $\leq$ used in [10] to ensure that equality of events is not confused with identity (=) used in the completion.

Axiom O17 deals analogously with the case

```
E1        P1     E2   P2
o-------->     o-------->
```

where two events initiate exclusive properties, and no explicitly given event breaks the holding of P1 between the two events.

Axiom O18 deals with the case

```
E1        P1          P2       E2
o----------->     <----------o
```

where one event initiates a property incompatible with a property terminated by a later event, and no explicitly given event breaks the holding of P1 between the two events.

Note that EC does not have a predicate that corresponds directly to *happens* in SEC. But the intended meaning of *initiates(E, P)* is that event E happens and initiates property P. Similarly the intended meaning of *terminates(E, P)* is that E happens and terminates P. Also the intended meaning of *time(E, T)* is that event E happens at time T. Similarly to SEC, EC also contains suitable definitions for < on time points. In addition, it may contain problem-specific facts about the < relation between events.

4 Some problems with EC

Most of the problems with EC arise from the complexity of its vocabulary, which includes many different predicates which are not always properly related to one another. For example, none of the following properties can be shown from EC or its completion:

a) The start of a period is before its end. That is

$$start(Period, E1) \wedge end(Period, E2) \rightarrow E1 < E2$$

b) An event that ends/starts a period terminates/initiates the property that holds for that period. That is

$$end(after(E1, P), E2) \rightarrow terminates(E2, P)$$
$$start(before(E2, P), E1) \rightarrow initiates(E1, P)$$

c) If a property is terminated after it has been initiated then the period of time for which it holds due to that initiation must have an end, i.e.

$$holds(after(E1, P)) \wedge terminates(E3, P) \wedge E1 < E3 \rightarrow \exists\, E2\ end(after(E1, P), E2)$$

Property (c) fails to hold because it is consistent with EC for an event e to initiate a property p which is then terminated infinitely many times without there being an earliest termination. This can be eliminated by insisting that the ordering relation on events be well-ordered, or that the set of events be finite.

A somewhat less serious problem [13] is the inaccuracy of axioms O3 and O4. Axiom O3, for example, states that

start(after(E, P), E)

even when E does not affect P. This problem does not lead to other, more serious consequences, and can be avoided by adding extra conditions *holds(after(E, P))* and *holds(before(E, P))* to O3 and O4, respectively.

One final problem is that the treatment of incompatible properties, in axioms O11 and O16-O18, caters for only two incompatible properties.

The fact that two properties p1 and p2 are incompatible, i.e.

incompatible(p1, p2)

can be expressed by an integrity constraint

$\neg$ [holdsAt(p1, T) $\wedge$ holdsAt(p2, T)].

By extending the language to include such integrity constraints explicitly we can deal with any number of incompatible properties. In NEC given the constraint

$\neg$ [holdsAt(p1, T) $\wedge$ holdsAt(p2, T) $\wedge$ ... $\wedge$ holdsAt(pn, T)]

and a description of n events

```
e1   p1
o------------------
     e2          p2
     o------------------
                 .
                 .
                 .        en     pn
                          o------------------
```

violating the constraint, we have two options. Either we reject one of the event descriptions, or we restore integrity by deriving an event E that ends one of the periods before e_n. The axioms O16-O18 of EC build in the latter option. Explicit use of integrity constraints has the added advantage of allowing both alternative ways of dealing with violations of integrity.

5 The relationship between EC and SEC

Perhaps the most prominent difference between the two calculi is that EC is concerned with properties holding for time periods as well as at time points, whereas SEC is concerned explicitly only with time points. It is not clear that the EC concern with time periods has any advantages. For this reason, in our comparison of the two calculi, we will focus on the *holdsAt* facts that can be derived in both cases.

One rather trivial difference between the two calculi is that in EC properties hold neither at their initiation nor at their termination, whereas in SEC they hold at termination but not at initiation. The advantage of the SEC convention is that it makes it possible to express the integrity constraints which state that preconditions of events and properties terminated by events must hold at the time of the event occurrences. We will exploit this possibility later in our formulation of NEC.

Except for the conventions about properties holding at end points, the two calculi imply equivalent *holdsAt* facts when the explicitly given event occurrences are complete. When they are incomplete, SEC derives at least the same *holdsAt* facts as EC. In the remainder of this section we express these claims more precisely. The proofs are omitted for lack of space.

Let	Oi, $1 \leq i \leq 18$, be the axioms of EC
	KER = {Oi : $1 \leq i \leq 15$}
	DER = {Oi : $16 \leq i \leq 18$}
Input:	a finite set of facts about event happenings and their types, times and ordering in the vocabulary of EC, i.e. facts of the form *time(E, T)*, *act(E, A)*, *E1< E2*.
Input':	the same facts as in Input, but in the vocabulary of SEC, i.e. facts of the form *happens(E, T)*, *act(E, A)*.
Domain:	the set of domain dependent axioms defining *initiates*, *terminates* and *incompatible*.
Domain':	same as Domain without any rules for *incompatible*.
Ineq:	a definition of the inequality predicate, <, on time points, satisfying the usual axioms of transitivity, anti-symmetry, etc.

We assume that the facts in Input about the < relation between events are compatible with the < relation between time points, i.e. we assume that the following property is satisfied.

$$E1 < E2 \leftrightarrow \exists\, T1, T2\ [\ time(E1, T1) \wedge time(E2, T2) \wedge T1 < T2\ ]$$

We assume that the SEC vocabulary does not contain the function symbols *fin* and *init* and the events in both Input and Input' are all named by constant symbols.

We assume the completion semantics for both EC and SEC. Therefore, in the rest of this section, we let

EC = Comp(KER ∪ DER ∪ Input ∪ Domain ∪ Ineq)
SEC = Comp(S ∪ Input' ∪ Domain' ∪ Ineq)

By the standard definition of Comp (2), both EC and SEC contain CET.

Theorem 1: SEC implies at least the same *holdsAt* facts as EC, i.e.:

If EC $\models$ holdsAt(p, t)
then SEC $\models$ holdsAt(p, t).

Theorem 2 states that the converse of Theorem 1 holds when the input contains complete information about events, i.e. when the axioms, DER are redundant. The redundancy condition can be expressed by:

if EC $\models$ W
then Comp(KER $\cup$ Input $\cup$ Domain $\cup$ Ineq) $\models$ W $\quad\Psi$

for all W of the form start(period , e) or end(period, e). (Actually, for the proof of theorem 2 it is sufficient that Ψ holds for W of the form end(period, e).)

Theorem 2: If Ψ then if SEC $\models$ holdsAt(p, t)
then EC $\models$ holdsAt(p, t)
except for end points, i.e. except for the case
EC $\models$ end(after(e, p), e1) $\wedge$ time(e1, t).

Note that the if-form of S in SEC can be rewritten as

holdsAt(P, T) $\leftarrow$ happens(E1, T1) $\wedge$ initiates(E1, P) $\wedge$ T1< T $\wedge$
$\neg$ discontinued(E1, P, T)
discontinued(E1, P, T) $\leftarrow$ happens(E2, T2) $\wedge$ terminates(E2, P) $\wedge$
T1< T2 $\wedge$T2< T

Let SEC1 be SEC with S replaced by the two clauses above. Now the if-forms of both SEC1 and EC are acyclic. Therefore SLDNF is complete as well as sound for these theories [1]. Therefore, we have the following two corollaries. Let EC' be the if-form of EC and SEC1' be the if-form of SEC1.

Corollary 1: If EC' $\models_{SLDNF}$ holdsAt(p, t)
then SEC1' $\models_{SLDNF}$ holdsAt(p, t).

Corollary 2: If Ψ and SEC1' $\models_{SLDNF}$ holdsAt(p, t)
then EC' $\models_{SLDNF}$ holdsAt(p, t)
except for end points, i.e. except for the case
EC $\models$ end(after(e, p), e1) $\wedge$ time(e1, t).

6 Towards a new event calculus (NEC)

In this section we propose the core of a new calculus, NEC, based on SEC. First, we modify SEC by replacing the binary predicate *happens(E, T)* in S with a unary predicate *happens(E)*, and assume (it is easy to write) a definition of < that applies when either operand is an event name or a time point. As a result, we can input information about event occurrences without knowledge of their absolute times. As before, the actual definition of < is immaterial. What matters is that the definition satisfies the usual properties, such as transitivity and anti-symmetry. We can now define *holdsAt(P, TE)* so that properties hold either at time points or at events:

$$\text{holdsAt}(P, TE) \leftrightarrow \exists\ E1\ \{\ \text{happens}(E1) \wedge \text{initiates}(E1, P) \wedge\ E1 < TE \wedge \\ \neg \exists\ E2\ [\text{happens}(E2) \wedge \text{terminates}(E2, P) \wedge E1 < E2 \wedge E2 < TE]\ \} \qquad N$$

The definitions of all other predicates in NEC remain the same as in SEC and EC. The domain-specific predicates, *initiates*, *terminates* and < between time points, are defined in iff-form. However, the problem-specific predicates, *happens*, *act*, *time* and < where one of the arguments is an event, are defined in if-form. The use of if-form for the definition of problem-specific predicates is necessary, as we will see later, because of the derivation of implicit event occurrences not contained in the input.

Strictly speaking, we need four different inequality predicates. One for inequality between time points, one for inequality between events, one for inequality between a time point and an event, and < defined in terms of the first three. We should complete only the first and the last.

The remainder of NEC consists of domain-independent integrity constraints I2 and I2' and domain-dependent ones of the form of I1 and I3.

For any n incompatible properties p1, ... , pn:

$$\neg\ [\ \text{holdsAt}(p1, T) \wedge\ \text{holdsAt}(p2, T) \wedge\ \ldots\ \wedge\ \text{holdsAt}(pn, T)\ ] \qquad I1$$

To obtain the functionality of derived events (axioms O16-O18 in EC) we add the following integrity constraint:

$$\text{holdsAt}(P, E) \leftarrow \text{happens}(E) \wedge\ \text{terminates}(E, P) \qquad I2$$

We treat preconditions of events in a similar way:

$$\text{holdsAt}(P, E) \leftarrow \text{happens}(E) \wedge\ \text{precond}(E, P) \qquad I2'$$

The predicate *precond* is domain-specific, similar to the predicates *initiates* and *terminates*, for example:

$$\text{precond}(E, P) \leftrightarrow \exists\ X, Y, Z\ [\ (\text{act}(E, \text{move}(X, Y, Z)) \wedge P = \text{clear}(X)) \vee \ldots\ ]$$

which states that a precondition of moving block X from Y to Z is that X be clear. Although we include I2' in NEC, we do not need it for any of the results presented in the remainder of this paper.

The final type of integrity constraint, I3, imposes maximality of time periods for particular domain-specific properties. In EC maximality is assumed to hold for all properties and is built into the calculus. For example, if two events cause you to be happy, i.e.

```
e1   happy              e2   happy
o----------->           o------------>
```

then the two subsequent periods are not allowed to overlap. This is a consequence of O17, which implies that your first period of happiness must have ended some time in between the two events.

In NEC we do not build such a general maximality assumption into the axioms, but allow it to be imposed, when desired for a particular property p, by means of a domain-specific integrity constraint of the form:

$$\neg\ [\ \text{happens}(E) \wedge \text{initiates}(E, p) \wedge \text{holdsAt}(p, E)\ ] \qquad \text{I3}$$

N together with I2, I2', and domain-specific constraints of the form of I1 and I3 constitute the core of NEC. This core can be extended in many directions, for example to include the treatment of ramifications, the holding of properties for maximal or non-maximal time intervals and to deal with concurrency. These are topics of continuing research, and we will not pursue them further in this paper.

Example: Suppose event e happens and terminates property p, i. e.

happens(e) F
terminates(e, p) F

In SEC (if or iff form) we cannot derive further consequences from this. In EC we can conclude only *holds(before(e, p))*. In particular, we cannot conclude that p holds at any time point before e. In fact, from the iff-form of EC we can even derive the undesirable conclusion $\neg\exists\, T\ \mathit{holdsAt}(p, T)$.

In NEC, however, we can conclude that, for some event occurrence E, p holds at all time points between E and e, i.e.

$$\exists\, E\ [\ \text{happens}(E) \wedge \text{initiates}(E, p) \wedge E< e \wedge \forall\, T\ (E< T\leq e \rightarrow \text{holdsAt}(p, T))\ ]$$

This conclusion is derived by using the given facts F with I2 to derive holdsAt(p, e), then using the only-if direction of N and finally using transitivity of < and N in the if direction. A similar result is obtained by [16] using abduction with the if-form of EC.

7 NEC versus SEC and EC

In this section we show that NEC implies consequences analogous to axioms O16-O18 of EC. Recall that the intention of O16-O18 is twofold: first, to ensure that periods are maximal, and second, to ensure that periods of incompatible properties are disjoint (i.e. do not overlap). The analogous theorems are formulated in the vocabulary of NEC, and contain existential quantifiers instead of the Skolem functions *init* and *fin*. We use the following correspondence between the vocabularies of EC and NEC:

EC	NEC
initiates(E, P)	initiates(E, P) $\wedge$ happens(E)
terminates(E, P)	terminates(E, P) $\wedge$ happens(E)
incompatible(P1, P2)	$\neg\exists$ T (holdsAt(P1, T) $\wedge$ holdsAt(P2, T))

$E1 < E2$	$E1 < E2$
broken(E1, P, E2)	not needed
exclusive(P1, P2)	$\neg\exists T$ [holdsAt(P1, T) $\wedge$ holdsAt(P2, T)] $\vee$ [$P1=P2 \wedge \neg\exists E$(happens(E) $\wedge$ initiates(E, P1) $\wedge$ holdsAt(P1, E)] (i.e. either P1 and P2 are incompatible or they are the same and maximal)

Note that in the completion of EC, which is our semantics for EC,
holds(after(E, P)) $\leftrightarrow$ initiates(E, P)
holds(before(E, P)) $\leftrightarrow$ terminates(E, P)

Theorem 3 below, which shows that NEC implies the analogue of O16, has two parts. Part (a) deals with incompatible properties and (b) deals with a single property which is maximal. Let $\Phi1$ and $\Phi2$ be as follows:

$$\begin{aligned}&\exists E3\ [\ \text{happens}(E3) \wedge \text{initiates}(E3, P2) \wedge E3 < E2 \wedge \neg\, E3 < E1] \leftarrow \\ &\quad \text{happens}(E1) \wedge \text{terminates}(E1, P1) \wedge \text{happens}(E2) \wedge \\ &\quad \text{terminates}(E2, P2) \wedge E1 < E2 \wedge \neg\exists T\ (\text{holdsAt}(P1, T) \wedge \text{holdsAt}(P2, T)) \qquad \Phi1\end{aligned}$$

$$\begin{aligned}&\exists E3\ [\ \text{happens}(E3) \wedge \text{initiates}(E3, P) \wedge E3 < E2 \wedge \neg\, E3 < E1] \leftarrow \\ &\quad \text{happens}(E1) \wedge \text{terminates}(E1, P) \wedge \text{happens}(E2) \wedge \\ &\quad \text{terminates}(E2, P) \wedge E1 < E2 \wedge \\ &\quad \neg\exists E\ (\text{happens}(E) \wedge \text{initiates}(E, P) \wedge \text{holdsAt}(P, E)) \qquad \Phi2\end{aligned}$$

Theorem 3:
a) NEC $\models \Phi1$
b) NEC $\models \Phi2$

Proof:
a) We assume the conditions of $\Phi1$ and show the conclusion. The conditions of $\Phi1$ and I2 imply

holdsAt(P2, E2)

This and the only-if half of N imply

$$\begin{aligned}&\exists E3\ [\ \text{happens}(E3) \wedge \text{initiates}(E3, P2) \wedge E3 < E2 \wedge \\ &\quad \neg\,\exists E4\ (\text{happens}(E4) \wedge \text{terminates}(E4, P2) \wedge E3 < E4 \wedge E4 < E2)] \qquad (1)\end{aligned}$$

Moreover, the conditions of $\Phi1$ also imply
holdsAt(P1, E1) and so $\neg$ holdsAt(P2, E1).

From this latter conclusion and the contrapositive of the if-half of N we conclude

$$\begin{aligned}&\forall E3\ \{\ \text{happens}(E3) \wedge \text{initiates}(E3, P2) \rightarrow \\ &\quad \neg\ [\ E3 < E1 \wedge \neg\,\exists\ E^*\ (\text{happens}(E^*) \wedge \text{terminates}(E^*, P2) \wedge \\ &\qquad E3 < E^* \wedge E^* < E1)\ ]\ \} \qquad (2)\end{aligned}$$

(1) and (2) imply

$$\exists E3\ [\ happens(E3) \wedge initiates(E3, P2) \wedge E3 < E2 \wedge$$
$$\neg\ \exists E4\ (happens(E4) \wedge terminates(E4, P2) \wedge\ E3 < E4 \wedge E4 < E2) \wedge$$
$$\neg\ (E3 < E1 \wedge \neg\exists E^*\ (happens(E^*) \wedge terminates(E^*, P2) \wedge$$
$$E3 < E^* \wedge E^* < E1))].$$

Using transitivity of <, the equivalence
$\neg(A \wedge \neg B) \equiv \neg A \vee B$
and the elimination of a false disjunct we show the conclusion of Φ1.

If we assume a total ordering on events, we can replace the negative conclusion
$\neg\ E3 < E1$ in Φ1 by the positive conclusion
$E1 < E3 \vee E1 \parallel E3$
where $E1 \parallel E3$ means that E1 happens at the same time as E3. This assumption of total ordering is built into axioms O16-O18 of EC.

b) The proof of part (b) is similar to that of part (a).

Theorem 4 shows that NEC implies the analogue of O17. Part (a) deals with incompatible properties and part (b) deals with a single maximal property. Let Φ3 and Φ4 be:

$$\exists E3\ [\ happens(E3) \wedge terminates(E3, P1) \wedge\ E1 < E3 \wedge \neg\ E2 < E3] \leftarrow$$
$$happens(E1) \wedge initiates(E1, P1) \wedge$$
$$happens(E2) \wedge initiates(E2, P2) \wedge E1 < E2 \wedge$$
$$\neg\exists T\ (holdsAt(P1, T) \wedge\ holdsAt(P2, T)) \qquad \Phi3$$

$$\exists E3\ [\ happens(E3) \wedge terminates(E3, P) \wedge\ E1 < E3 \wedge \neg\ E2 < E3] \leftarrow$$
$$happens(E1) \wedge initiates(E1, P) \wedge$$
$$happens(E2) \wedge initiates(E2, P) \wedge E1 < E2 \wedge$$
$$\neg\ \exists E\ (happens(E) \wedge\ initiates(E, P) \wedge holdsAt(P, E)) \qquad \Phi4$$

Theorem 4: Assume that the input facts concerning *happens* and the definition of < satisfy the property that

$$(\exists T\ E < T) \leftarrow happens(E),$$

and that there are a finite number of event occurrences. Then

a) NEC |= Φ3
b) NEC |= Φ4

The proofs of theorems 4 and 5 are omitted for lack of space.

Theorem 5 shows that NEC implies the analogue of O18. Let Φ5 be:

$$\exists E3, E4\ [\ happens(E3) \wedge terminates(E3, P1) \wedge happens(E4) \wedge initiates(E4, P2) \wedge E1 < E3 \wedge E4 < E2 \wedge \neg\ E4 < E3] \leftarrow happens(E1) \wedge initiates(E1, P1) \wedge happens(E2) \wedge terminates(E2, P2) \wedge E1 < E2 \wedge \neg\exists T\ (holdsAt(P1, T) \wedge holdsAt(P2, T)) \qquad \Phi 5$$

Theorem 5: NEC $\models \Phi 5$.

Theorems 3-5 and the discussion in section 6 shows that NEC extends SEC and EC. To exploit this extension, however, requires the use of an appropriate proof procedure for logic programs in iff-form augmented with integrity constraints. One such proof procedure, with appropriate soundness and completeness results, has been developed by Fung [6].

As we remarked earlier in section 6, because NEC can be used to derive implied events, the predicates *happens*, *act*, *time*, and < where one of the arguments is an event occurrence cannot be completed. As a consequence, whereas in SEC and EC we might derive unconditional conclusions of the form

holdsAt(p, t)

in NEC, we derive conditional conclusions of the form

$$holdsAt(p, t) \leftarrow \neg\ \exists\ E2\ [\ happens(E2) \wedge terminates(E2, p) \wedge e < E2 \wedge E2 < t].$$

The proof procedure in [6] allows unconditional *holdsAt* conclusions to be derived by "retrospectively" completing the input. The proof procedure of Denecker and De Schreye [3] seems to give similar results.

8 Conclusions

We believe that the theorems presented in this paper help to explain why simplified forms of the event calculus have gradually replaced the original event calculus in practice. We believe that they also demonstrate the more general potential computational advantages of the iff-form of logic programs augmented with integrity constraints.

Acknowledgements
This research was supported by Fujitsu Laboratories.

References
[1] Cavedon, L., "Acyclic programs and the completeness of SLDNF-resolution", *Theoretical computer science* 86(1), 1991, 81-92.

[2] Clark K.L., "Negation as failure", in "Logic and data bases", Gallaire H. and Minker J. [eds.] , Plenum Press, 1978, 292-322.

[3] Denecker M. and De Schreye D., "SLDNFA: an abductive procedure for normal abductive programs", Proceedings of International Conference and Symposium on Logic Programming , 1992, 686-700.

[4] Denecker M., Missiaen, L., Bruynooghe M., "Temporal reasoning with abductive event calculus", Proceedings of ECAI, 1992.

[5] Eshghi K., "Abductive planning with event calculus", Proceedings of International Conference on Logic Programming,, MIT Press, 1988.

[6] Fung T.H., "An abductive proof procedure based on Clark completion", Proceedings of Logic Programming Workshop, GLP, 1994, Zurich.

[7] Kowalski R., "Database Updates in the Event Calculus", *Journal of Logic Programming,*, Vol. 12, No. 162, 1992, pp. 121-146.

[8] Kowalski R., "A dual form of logic programming", Lecture Notes, Workshop in Honour of Jack Minker, University of Maryland, 1992.

[9] Kowalski, R. and Sadri, F., "The Situation Calculus and Event Calculus Compared", Proceedings of International Logic Programming Symposium, MIT Press, 1994.

[10] Kowalski, R. and Sergot, M., "A Logic-based Calculus of Events", *New Generation Computing,* Vol. 4, No. 1, February 1986, pp. 67-95.

[11] Kowalski, R., Toni, F. and Wetzel, G., "Towards a declarative and efficient glass-box CLP," Proceedings of Logic Programming Workshop, GLP, 1994, Zurich.

[12] Missiaen L., "Localized abductive planning with the event calculus", PhD Thesis, Department of Computer Science, K. U. Leuven, 1991.

[13] Pinto J. and Reiter R., "Temporal reasoning in logic programming: a case for the situation calculus", Proceedings of International Conference on Logic Programming,, MIT Press, 1993, 203-221.

[14] Provetti A., "Hypothetical reasoning about actions: from situation calculus to event calculus", *Computational Intelligence*, Vol. 12, No. 2, 1995.

[15] Sadri F., "Three recent approaches to temporal reasoning", in "Temporal logics and their applications", Galton A. [ed.], Academic Press, 1987, 121-168.

[16] Shanahan M.P., "Prediction is deduction but explanation is abduction", Proceedings of International Joint Conference on Artificial Intelligence, 1989.

[17] Sripada, S., Rosser, B., Bedford, J., Kowalski, R., “Temporal Database Technology for Air Traffic Flow Management”, Proceedings of First International Conference on Applications of Databases, ADB-94, Lecture Notes in Computer Science, 819, Springer-Verlag, 1994.