Abstract
A semantics for tense, modality, and aspect in natural language must capture causal and contingent relations between events and states as well as merely temporal ones. The paper investigates a non-reified dynamic logic based formulation of the situation calculus as a formalism for a computational semantics for a number of temporal categories in English and suggests that some recent claims that dynamic logics are inherently unsuitable for this purpose have taken too narrow a view of the situation calculus.

1 Temporal Ontology
The most important thing to observe about the temporal ontology implicit in natural language tense and aspect is that it is not purely temporal. To take a simple example the English perfect when predicated of an event like losing a watch says that some contextually retrievable consequences of the event in question hold at the time under discussion (Such consequences have sometimes been described under the heading of "present relevance of the perfect.) As a result, conjoining such a perfect with a further clause denying those consequences is infelicitous.

(1) I have lost my watch (# but I have found it again)

In this respect the perfect stands in contrast to the more purely temporal tenses, such as the past, which make no comparable claim about the consequences of the core event.

(2) Yesterday I lost my watch (but I have found it again)

It is because categories like the perfect are not purely temporal that it is usual to distinguish them from the tenses proper as 'aspects.' Another aspect whose meaning is not purely temporal is the progressive or Imperfective. The predication that it makes concerning the core event is a subtle one. While the progressive clearly states that some event is ongoing at the time under discussion it is not necessarily the event that is actually mentioned Thus in a helow there seems to be a factive entailment about an event of writing. But in b, there are no such entailment concerning an event of writing a sonnet, for b is true even if the author was interrupted before he could complete the action.

Figure 1 The nucleus

(3) a. he was writing * he was wrote
   b. he was writing a sonnet * he was wrote a sonnet

Dowty [1979] named this rather surprising property of the progressive the imperfective paradox. The imperfective paradox is a sign that we must distinguish various types or sorts of core event in natural language temporal ontology. This system, which is described at greater length in [Steedman in press], is briefly summarised as follows.

There are two key insights into this system which most theories either build upon or are forced to reinvent. The first concerns the temporal ontology itself and is usually attributed to Vendler [1967] though there are predecessors in work by Jespersen Kenny and many earlier authors (including Aristotle Vendler's taxonomy) was importantly refined by 'erkul and Dowty, and has been further extended by many others. Such taxonomies typically distinguish 'states from events' and divide the latter into a number of sorts or types. Vendler distinguished 'activities (events which have duration but don't change state like heat* writing), achievements (events which have no duration but do change state like eats amyng), and accomplishments (which hav' duration and change state like heats writing a sonnet).

Many authors have proposed recursive sort hierarchies. Moens [1987, 1988] explained the aspectual sort hierarchy and possible coercions among Aktionarten in terms of a structure of the kind represented schematically in figure 1, representing an association in memory or the knowledge representation of all events with characteristic preparations and consequents, an idea that has since been adopted in DR Theory (Kamp and Reyle, [1993, p 557-570]. Moens claimed that the preparation is in Vendler's terms an activity, the consequent is a (perfect) state, and that the core event is an achievement. There is a great deal more to say about the status of these categories, but we will take it as read here, noting merely that we shall follow these authors in assuming that accomplishments like writing a sonnet are composites of an activity of writing and a culminating achievement.
Figure 2: Past vs. Perfect (from Reichenbach 1947)

finishing the sound

The second key insight concerns relations among temporal entities, and is due to Reichenbach (1947, Chapter VII, Sections 48 and 51) who sketched a semantics for tense in terms of three underlying abstract time-points: (Again there are precedents, notably in work by Jespersen.) Reichenbach’s real innovation was the notion of the reference point, which can be identified with the notion “the time (or situation, or context) that we are talking about.” It is easiest to convey the idea by example. Reichenbach offers the diagrams in Figure 2, in which the arrow indicates the flow of time to show the distinctions between the past perfect (or past perfect) and the present perfect (all of which he includes under the heading of “tenses of verbs”). The important insight here is that the simple past is used to make a statement about a past time, whereas the perfect is used to make a statement about the present as was noted earlier in connection with the present-relevance property of examples like (1). Besides the authors already cited this system has also been extended importantly by Hornstein (1990).

Reichenbach’s claim is consistent with the observation that past tense is unlike the perfect: demands that the past reference point be explicitly established, either by a modifier, such as a when clause, or by the preceding discourse. Thus a below is inappropriate as the first utterance of a discourse except to the extent that the reader accommodates a temporal referent in Lewis and Stalnaker’s sense of that term—that it refers to an appropriate individual in the database as one often must at the beginning of a modern novel. B is appropriate on the assumption that the hearer can identify the time in the when clause.

2 Temporal Relations and the Situation Calculus

One of the most useful and attractive features of the situation calculus of McCarthy and Hayes (1969) was the use of terms like result(arrive(person),s) as individuals denoting situations or states as functions of other situations. (Functions like result are known as situational fluents. In the present paper we will reserve the term fluent for this particular kind.) Such terms can be used in rules like the following to transparently capture the notion that a person is present in the situation that results from their arriving.

(5) w person present(person result(arrive(person), s))

The STRIPS representation of actions and the associated solution to McCarthy and Hayes’ frame problem was originally thought of in propositional terms. However, Kowalski (1979) (cited in 1974) showed it to be elegantly realizable in entirely declarative terms via the introduction of a closed world assumption and a more radical use of reflection to simulate modal quantification (see Nilsen 1980 p. 208-215 for discussion). Related techniques and their relation to ramification and quantification are further explored by by Schubert (1990, 1991) and Reiter (1991, 1992).

Kowalski proposed a predicate holds, which applies to a proposition represented as a term and a state. In this notation a minimal world can be defined as a starting state including three clear blocks is defined as follows:

(6) holds(clear(a), s) ∧ holds(clear(b), s) ∧ holds(clear(c), s)

The action of putting x on y can be represented as a STRIPS rule as follows. The preconditions are defined by the following rule which says that if you can get at x and you can get at y, the preconditions for putting x on y hold.

(7) holds(clear(x), s) ∧ holds(clear(y), s) ∧ (x ≠ y) ∧ preconditions(puts(x, y), s)

(8) a hold(on(x, y), s) 
   -- holds(clear(z), result(puts(x, y), s)) 
   -- hold(on(x, y), result(puts(x, y), s))

Since we are assuming negation as failure, we do not need to state explicitly that y is no longer clear. This fact is implicit in the following frame axiom which is the only frame axiom we need for the action of putting x on y.

(9) hold(p, s) ∧ (p ≠ clear(y)) ∧ (p ≠ on(x, z)) 
   -- hold(p, result(puts(x, y), s))

(10) Since we are assuming negation as failure, we do not need to state explicitly that y is no longer clear. This fact is implicit in the following frame axiom which is the only frame axiom we need for the action of putting x on y.

(11) hold(p, s) ∧ (p ≠ clear(y)) ∧ (p ≠ on(x, z)) 
   -- hold(p, result(puts(x, y), s))

(12) The use of inequalities rather than implication here embodied a Horn-like assumption restricting p to terms in these rules. Kowalski’s proposal was followed by much work on tense using similar calculi (Allen 1984, McDermott 1982, Kowalski and Sergot 1986). It was also closely related to the notion of “subsumption of qualifications” – see McCarthy 1977, esp. p. 1041, and much other subsequent work collected and reviewed in Ginsberg (1987). In particular Reiter (1991) shows how the restricted frame axioms or successor state axioms can be derived automatically.
We can now define a predicate possible closely related to the (deontic) modal operator $O$, over the set of possible states, via the following rules, which say that the start state $s_0$ is possible and the result of an action in a state is possible if its preconditions hold:

(10)  
\[ \text{possible}(s) \land \text{preconditions(action, s)} \rightarrow \text{possible(result(action, s)))} \]

The earlier goal of stacking a on b on c can now be realised as the goal of finding a constructive proof for the following conjunction:

(11)  
\[ \text{possible}(s) \land \text{holds(on(a b) s)} \land \text{holds(on(b c) s)} \]

These rules can be very straightforwardly realised in Prolog and can be made to yield a proof (although the search problem of finding such proofs automatically remains hard in general), in which:

(12)  
\[ \ast = \text{result(puton(a b) result(puton(b c), s))} \]

This technique restores declarativity to the logic embodying the STRIPS problem. There is a sense in which – despite the involvement of the closed world assumption – it also restores monotonicity, for so long as we do not add new facts (like some previously unobserved object being present, or a familiar one having fallen off its support) or some new rule or frame axiom (say defining a new action or stating a new precondition on an old one) then we can regard negation-as-failure as merely efficiently encoding classical negation.

Of course, in the real world we do learn new facts and rules, and we encounter exceptions to the closed world assumption of complete knowledge. These problems are known in AI as the ramification problem (that is, that actions may have indefinitely many unforeseen consequences that our default model does not and cannot predict) and the qualification problem (that actions may have indefinitely many preconditions that our default model does not and cannot anticipate). In many recent papers the frame problem is assumed to include these further problems. However, if we are in possession of an efficient default model which works reasonably well most of the time, it may well be wiser to regard the problem of coping with its failures as residing outside the logic itself, in the truth maintenance or housekeeping system. Rather than a nonmonotonic logic, we could think in terms of a system of truth-maintaining transitions between genuinely monotonic logics, a view that has been proposed by Kakuk.

However, there is another way of looking at all of these variants of the situation calculus. The extent to which the accessibility relation is defined in terms of a number of different events or causal primitives, possibly a large number it is possible to regard each of these as defining its own distinct accessibility relation, possibly differing from one in properties like transitivity. Such systems can then be viewed as instances of the “dynamic” logics that were developed in the first place for reasoning about computer programs – see Pratt [1976], Harel [1984], and Goldblatt [1992]. The application of various forms of dynamic logic in knowledge representation and natural language semantics has been advocated by Moore [1980], Rosenschein [1981], Webber [1983], Pednault [1989], and Scholl and Levendusky [1993].

Dynamic logics relativise the modal operators to individual actions, events or programs. For example if a (possibly non-deterministic) program or command $o$ computes a function $F$ over the integers, then we might write the following:

(13)  
\[ n \geq 0 \rightarrow [o(y = F(n))] \]

(14)  
\[ n \geq 0 \rightarrow (o(y = F(n)) \]

The intended meaning of the first of these is that for $n \geq 0$, after every execution of $o$ that terminates, $y = F(n)$.

The second is (dually) that there is an execution of $o$ which terminates with $y = F(n)$.

While all of the calculi that we have considered so far are ones in which the elementary programs $o$ are deterministic, dynamic logics offer a framework which deals gracefully with concurrent and probabilistic events, offering a notation in which all of the theories discussed in it can be compared.

The particular dynamic logic that we are dealing with here is one that includes the following dynamic axiom (the operator $r$ is an operation related to composition and to von Wright " $T$")

(15)  
\[ [o][p] \rightarrow [o \beta] p \]

In this we follow Moore [1980, ch 5] and Rosenschein [1981]. The situation calculus and its many variants can be seen as refined versions of this dynamic logic.

We achieve an immediate gain in perspicuity by replacing the nested notation in (15) by the equivalent dynamic expression $b$:

(16)  
\[ \text{hold(on(a b) o on(b c))} \land \text{result(puton(b c))} \land \text{accessibility(x y z) } \rightarrow \text{possible(puton(x y z))} \]

The consequences of puton($y z$) are now written as follows:

(17)  
\[ (\text{clear}(a) \land \text{clear}(b) \land \text{clear}(c)) \]

The axiom defining the preconditions of puton($y z$) is now directly definable in terms of the predicate possible, which can now be identified with deontic modal possibility.

(18)  
\[ \text{possible(puton(x y z))} \land x \neq y \rightarrow \text{possible(puton(x y z))} \]

The consequences of puton($y z$) are now written as follows:

(19)  
\[ \text{possible(on(x z) = \text{puton}(y z))} \land \text{clear}(z) \]

(20)  
\[ \text{on(y z) = \text{puton}(x z)) \land z \neq y \rightarrow \text{possible(puton(x z))} \]

The transitive part of the possibility relation is now reduced to the following:

(21)  
\[ (\text{possible}(a) \land \text{possible}(b)) \rightarrow \text{possible}(a b) \]

This fragment preserves the virtues of Kowalski’s treatment in a modal notation. That is, the following conjunctive goal can, given a search control be made to deliver a constructive proof that $o = \text{puton(b, c), puton(a, b)}$.
The suppression of state variables in dynamic logic affords some improvement in perspicuity over the otherwise equivalent previous proposals of howells, McCarthy, Schulbert, and Reiter that it is here used to capture, and makes it easier to extend the calculus.

The above examples only concern non-composite or non-durative events, like the original situation calculus. However, the following formal Horn clauses begin to capture the composite events discussed earlier, along the lines suggested by Steedman [1982], Moens [1987] and White [1994]. (The example is greatly simplified and omits many rules needed to capture even this small domain completely.) First we need axioms defining the consequent and preconditions for starting and stopping.

(23) a $\models [start(p) \land progress(p)]$

b $\models not(in\_progress(p)) \rightarrow possible\_start(p)]$

(24) $\models in\_progress(p) \rightarrow possible\_stop(p]$.

We also need a frame axiom for stopping (which could be derived as in Reiter's [1981])

(25) $\models p \land (p \equiv in\_progress(p)) \rightarrow \bot$(p)$p$

Finally, we need a definition of the progressive coercion achievements to accomplishments and accomplishments to preparatory activities. (Note that in b below, we assume in line with the discussion in section 2.1, that accomplishments are made up of an activity and a culminating achievement. These are here represented as terms in lieu of a proper system of sorts.)

(26) $\models in\_progress(activity) \rightarrow progress(activity(ach))$

$\models progress(activity(ach)) \rightarrow progress(ach)$

$\models progresses(ach)$

$\models achievement(ach)$

$\models preparation(ach) activity(ach)\rightarrow progress(ach)$

$\models progresses(ach)$

$\models achievement(ach)$

The following query asks (slightly artifically) for a plan $\alpha$ yielding a state where kraits is finishing writing the novel in Disgust of vulgar superstition.

(27) $\exists alpha \alpha (in\_progress(ach) \land progress(ach) \land finish(ach, write(kraits in\_disgust)) )$

(The function finish maps an accomplishment onto its culminating achievement and is distinct from stop the endpoint of an activity.) To find the plan we must assume that the knowledge base also makes explicit the relation between finishing an activity and its characteristic preparation the activity itself implicit in the nucleus of figure 1.

(28) $\models preparation(ach) \land finish(ach, write(kraits in\_disgust))$

The accessibility relation implicit in definition 21 now gives rise to a proof where

(29) $\models start(write(kraits in\_disgust))$.

The proof that generates this plan does not involve the subgoal of showing $\alpha (in\_progress(ach) \land progress(ach) \land finish(ach, write(kraits in\_disgust))$. Indeed the proof would be quite consistent with adding the denial of that fact, because the variable ach in rule 26 is not involved in the antecedent capturing the imperfective paradox.

Of course asking for a plan to bring about a situation in which kraits is finishing writing In Disgust of vulgar superstition is slightly artificial because such states are extreme, and there may be several such plans. For example, consider the effect of adding the following rule defining the consequent and preconditions of arriving

(30) a $\models [arrive(x) \land present(x)]$

b $\models not(present(x)) \rightarrow possible(arrive(x))$

The accessibility relation 21 now allows

(31) a $\models start(visit(kraits in\_disgust)) \land arrive(x) \land present(x)$

b $\models start(visit(kraits in\_disgust)) \land arrive(x)$

As plans these are rather foolish because of well-known inherent limitations in the simplest STIP? planners, although incorrect plans such as the following are still correctly excluded for the goal in question

(32) a $\models start(visit(kraits in\_disgust)) \land arrive(x)$

b $\models stop(visit(kraits in\_disgust))$

Part of the problem is that we are not yet distinguishing true consequences of fluents, including causal relations among fluents themselves, from facts that are merely coincidentally true in the state that results because of the inertial property of the frame axiom. Nor are we distinguishing causal relations between fluents from the temporal sequence.

We can remedy this shortcoming by distinguishing the temporal sequence operator from a causal or contingent sequential operator, which we will write as $\circ$ because of its relation to one of Levens' [1988] operators. Accordingly we need to add some further rules parallel to 21 reflecting dynamic modal causality including the following

(33) $\models possible(a) \land [a \equiv causal(b)] \rightarrow possible(c)$

We now add a rule saying that anyone else being present causes kraits to stop writing

(34) $\models present(\alpha$) $\land (x \equiv not(kraits) )$

$\models in\_progress(visit(kraits y))$,

$\models causal(stop(visit(kraits y))$}

We can now search for plans which make an event of kraits stopping writing necessarily occur like a below, as distinct from those that merely make it possible like b by constructively searching for a proof that possible($\alpha*stop(visit(kraits y))$)

(35) a $\models start(visit(kraits y)) \land present(\alpha$) $\land x$,

b $\models start(visit(kraits y))$

Again the examples are artificial; their usefulness for an account of tense and temporal anaphora will become apparent in the next section.

3 Temporal Anaphora

The event-based calculus over counterfactual partially specified states discussed in the previous section offers a promising candidate for a representation of Reichenbach's reference point $R$, in the form of determinative situational fluents [a]. This opens up the possibility of applying the general modal apparatus developed so far
not only for quantifying over states but to act as the temporal link between sentences and clauses, as in _when-clauses_ and multi-sentence discourse. Several logical and computational approaches have explored this possibility.

Temporal anaphora like all discourse anaphora and reference resolution is even more intimately dependent upon world knowledge than the other temporal categories that we have been considering. In order to control this influence, WP will follow the style of much work in AI drawing most of our examples from a restricted domain of discourse. We will follow Isard [1974] in taking a board game as the example domain. Imagine that each model in a modal structure is represented as a database, or collection of facts describing not only the position of the pieces in a game of chess, and the instantaneous moves at each frame, but the fact that at certain times durative or composite events like exchanging Rooks or White attacking the Black Queen are in progress across more than one state.

Consider the following examples from such a domain:

(36) a. When I took your pawn you took my queen
    b. I took your pawn. You took my queen

The Tense-clause in a, above establishes a reference point for the tense of the main clause, just as the definite NP _my queen_ establishes a referent for the pronoun. Indeed the Tense-clause itself behaves like a definite, in that it seems to presuppose that the event of my taking your pawn is identifiable to the hearer. (Of course, the reader will have effortlessly accommodated this presupposition in Lewis and Stalnaker’s sense of the term.) The first sentence in b, above, behaves exactly like the _when_ clause in setting the reference point for the second. The only difference is that the simple declarative _I took your pawn_ itself demands a previously established reference point to be anaphoric to, whereas the _when_ clause causes a new reference point to be constructed.

As has been frequently noticed, the state to which the tense in _you took my queen_ refers in a, above, is not strictly the state in which _I took your pawn_. It is the state that resulted from that action. However, it is not invariably the case that the temporal reference point moves on in this way. Most obviously a stative main clause is primarily predicated of the original reference point of the _when_-clause.

(37) When I took your pawn I did not know it was protected by your knight

(Presumably, the ignorance in question may have ended with that very move.) Events also may be predicated of the original reference point rather than moving the action on.

(38) When I took your pawn, I used a rook

In fact, as Ritchie [1979], Partee [1984], Moens and Steedman [1988], and Kamp and Reyle [1993] have pointed out, in strictly temporal terms, we can find main clauses that precede the reference point established by a _when_ clause.

(39) When I won my only game against Bobby Fischer, we played Australian Rules

These phenomena arise because the temporal referent is _not_ strictly temporal. Rather than being a time or an interval, it is (a pointer to) an event-nucleus of the kind discussed earlier.

In the terms of our modal frame, the preparation of an event is the activity or action that led to the state in which that achievement took place. The consequent is the consequent state, and includes the entire subset of states accessible from that state. The referent-setting effect of a _when_-clause can then be seen as identifying such a nucleus. The main clause is then temporally located with respect to the nucleus. This may be by lining it up with the core event itself, either as a proper part of the initial state, as in example 37, or as a property of the transition itself, as in 38. Alternatively, since accessibility is defined in terms of the subsequent actions, the actual subsequent actions is a possible main clause, as in 36. Or the main clause may be located with respect to the preparation, as in 39. Which of these alternatives a given example gives rise to is a matter determined by the knowledge representation, not by the grammar.

On the assumption that the consequent in the nuclear referent includes the entire subset of future states, the information needed by conditionals modals, and other referent-setting adverbials will be available.

(40) a. If you take my queen, you may win
    b. If you had taken my queen, you might have won
    c. Since you took my queen, you have been winning.

All of this suggests that states of partial possible worlds in a logic of action deriving ultimately from von Wright and McCarthy and Hayes, with a much enriched ontology involving a rather intimate connection to the knowledge-base are appropriate candidates for a Fincher-Habermaz anaphoric account of tense and temporality. But this does not tell us how the temporal referent is set up to act as a referent for anaphora.

In the dynamic situation calculus, the history of events is a sequence of fluents such as the following:

(41) start(write(keats in disgust)) arrive(chapman)
    stop(write(keats in disgust))

The referent of a _when_-clause such as _When Chapman arrived is simply the sequence up to and including arrive(chapman), namely_.

(42) start(write(keats, in disgust)) arrive(chapman)

To identify the referent we need the following definition of a relation we might call _evoke_. This is merely a logic-programming device which defines a search for a deterministic situational fluent of the form α, β or α and over a history in which the sequence operators are _left-associative_ (we only give the rules for the operator here).

(43) a. _evoke(α, β, (γ, δ))_
    b. _evoke(α, β, (γ, δ)) — evoke((α, β), (γ, δ))_

The referent-setting effect of _when_ can then be captured in terms of _two_ approximations in the following rules which first find the current history of events, then evoke a suitable reference point, then test for the appropriate relation when (Again this is a logic-programming hack which could be passed over, and again there are two further rules with _if_ , that are omitted here).
(44) \[ a \models \text{\text{S}}(\text{history}) \text{evoke}((\alpha \theta), \text{history}) \quad [\alpha, \theta]^p \]
\[ - \quad \text{when}(\beta, \text{state}(p)) \]
\[ b \models \text{\text{S}}(\text{history}) \text{evoke}((\alpha \beta), \text{history}) \]
\[ - \quad \text{when}(\beta, \text{event}(\alpha)) \]

The predicate \( S \) determines the Reichenbachian speech point, which is a fluent or sequence of fluents \( S \text{(history)} \) assumed to be available in the database, as a fact.

The first rule, \( a \), applies to when sentences with state-type main clause propositions, and says that \( \text{when}(\beta, \text{state}(p)) \) is true if you can evoke a fluent ending in \( \beta \) after which \( p \) holds. The second applies to when sentences with event-type main clauses, and says that \( \text{when}(\beta, \text{event}(\alpha)) \) is true if you can evoke a fluent whose last two events are \( \beta \) and then \( \alpha \).

The question \( b \), below concerning the ensuing state therefore translates into the query \( b \).

(45) \[ a \text{ When Chapman arrived, was heats finishing writing \text{In Disguist of Vulgar Superstition?} } \]
\[ b \quad \text{when}(\text{arrive}(\text{chapman})) \text{state}\text{progressive(achivement\text{-}\text{finish}(\text{write}(\text{heats in disguise}))))} \]

In our greatly simplified world this is true despite the fact that under the closed world assumption, heats had not finished the poem, because of the earlier elimination of this incoherent paradox.

A when-question with an event in the main clause as in \( b \), below translates as \( b \).

(46) \[ a \text{ When Chapman arrived, did heats stop writing \text{In Disguist of Vulgar Superstition?} } \]
\[ b \quad \text{when}(\text{arrive}(\text{chapman})) \text{event}\text{stop}(\text{write}(\text{heats in disguise})) \]

In the case at hand, this will yield a proof with the following constructive instantiation:

(47) \[ \text{when}(\text{stop}(\text{write}(\text{heats in disguise}))) \text{arrive}(\text{chapman}) \]
\[ \text{event}\text{stop}(\text{write}(\text{heats in disguise})) \]

In either case, the enduring availability of the Reichenbachian reference point for later simple tensed sentences can be captured on the assumption that the act of evoking a new relevant causes a seditious to the database, causing a new fact (say of the form \( R(\alpha) \)) to be asserted, after any existing fact of the same form has been removed, or retracted. (We pass over the formal details here, merely noting that for this purpose a blatantly non-declarative STRIPS-like formulation seems to be the natural one although we have seen how such non-declarativity can be eliminated from the system.)

The representation also captures the fact that heats stopped writing the poem because Chapman arrived whereas Chapman merely arrived after heats started writing, not because of it.

Of course, it will be clear from the earlier discussion that such a system remains over-simplified. Such examples also suggest that the fluents themselves should be considerably enriched on lines hinted at in earlier sections. They need a system of types or sorts of the kind discussed in section 1. They should also be structured into nested structures of causal or, more generally, contingent sequences.

Since we have also observed that main clause events may be simultaneous with, as well as consequent upon, the when clause event, fluents must also be permitted to be simultaneous, perhaps using the connective \( \text{\textbf{and}} \) introduced by Fejes [1987] to capture the relation between embedded events like starting to write \( \text{In Disguist of Vulgar Superstition} \) and starting to write, generalising the above rules accordingly. Partial ordering of fluents must also be allowed. The inferential possibilities implicit in the notion of the nucleus must be accommodated, in order to capture the fact that one event may cause the preparation of another event to start, thereby embodying a non-immediate causal effect.

Very little of this work has been done, and it may be unsafe to speculate in advance of concrete solutions to the many real problems that remain. However the limited fragment outlined above suggests that dynamic logic may be a promising framework in which to pursue this further work and bring together a number of earlier approaches. In this connection, it is perhaps worth remarking that of the seven putative limitations of the situation calculus and its relatives claimed in the critical review by Shoban and Goyal [1986, p. 422-424] five limitation to instantaneous events, difficulty of representing non-immediate causal effects, ditto of concurrent events, ditto of continuous processes, and the frame problem] either have been overcome or have been addressed to some extent in the published work within the situation calculus, while the remaining two (the qualification problem and the ramification problem) have not been overcome in any framework, possibly because they do not belong in the logic at all.

Acknowledgements

Thanks to Anos van Benthem, Stephen Isard, Mark Johnson, Marie Moens, Charlie Ortony, Jong Park, Matthew Stone, Rich Thompson, Bennie Webber, and Michael White for advice and criticism. Support was provided in part by NSF grant nos. TR91-17110 and CISE II, DAA A-88-42T19, DARPA grant no. N66001-84-C-0641 and ARO grant no. DAAL04-84-K-0326.

References

[Hare, 1984] David Harel Dynamic Logic, in Dan Glynn and Franz Guenthner, Handbook of Philosophical Logic vol 2 Dordrecht, Reidel
[Isard, 1974] Stephen Isard What would you have done if Theoretical Linguistics 1 233-255

STEEDMAN 1297

[Kowalski 1979] Robert Kowalaki Logic for Problem Solving Amsterdam, North Holland


[Lansky 1986] Amy Lansky A representation of Parallel activity based on events structure and causality Proceedings of the workshop on planning and reasoning about action Timberlane Lodge Mount Hood OR, 50-86


[McCarthy and Hayes 1960] John McCarthy and Pat Hayes Some philosophical problems from the standpoint of Artificial Intelligence in Bernard Meltzer and Donald Michie (eds ), Machine Intelligence 4 Edin


[Partee 1984] Barbara Partee Nominal and temporal anaphora Linguistics and Philosophy, 7 243-286


[Reiter, 1993] Ray Reiter Proving properties of states in the situation calculus Artificial Intelligence 64 337-351

[Ritchie, 1979] Graeme Ritchie Temporal clauses in English Theoretical Linguistics, 6 87-115


[Webber, 1983] Bonnie Webber Logic and Natural Language IEEE Computer, Special Issue on Knowledge Representation, October 1983, pp 43-46
