A Representation of Time for Planning

Technical Note 278

February 1983

By: Peter Cheeseman

Artifical Intelligence Center
Computer Science and Technology Division

SRI International

SRI International
333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Abstract

A new time representation is described that allows a continuously changing world to be represented, so that queries about the truth of a proposition at a instant or over an interval can be answered. The deduction mechanism used to answer the common queries necessary in planning is the same as that employed for deducing all other information, thereby avoiding the need for a specialized time expert. The representation allows any time information to be represented without forcing an over specification. The implementation of this representation requires mechanisms to detect the effects of world changes on previous deductions (truth maintenance).

§1 Introduction

Since plans operate over time, the importance of an appropriate representation of time is clear. Any such representation should include the following characteristics:

(1) The representation must be expressive, i.e., be able to represent any time relationship used in planning.

(2) It should not force the over specification of time relationships that are only partially known.

(3) It should allow both relative and absolute (i.e., chronological or calendar) time information.

(4) The representation should allow efficient (fast) computation of the most commonly needed time relationships.

Earlier planners represented time by instantaneous "snapshots" of the state of the world at particular times (e.g., STRIPS—Fikes et al. [1972]). This method is clearly insufficient for representing continuous changes, as well as conditions that are required to be true over different time intervals. More recent attempts to model time have concentrated on developing temporal-expert modules for use by other systems. These temporal experts use time information relating events to one another or to an absolute time scale.

A new method of time representation suited to planning is described in this paper. This representation allows integration of reasoning about time with all other reasoning performed by the planner. It associates time intervals with individual propositions and reduces considerably the computation required to deduce the truth value of any proposition in time. The propositional world representation does not explicitly include events—all events are captured through their effects on the world description.

The inference method (and notation) assumed for this representation is backward chaining, similar to that of PROLOG (Warren et al., [1977]). However, this method must be modified to maintain a consistent world representation, using methods such as "if-added demons" to monitor and respond to changes in the world representation. The deduction of time relationships between different propositions can involve searching through a partially ordered time network, which in the worst case may require searching through a large fraction of this network.

The following section reviews previous time representations used in planning, while the subsequent section introduces a new representation. The final section reviews other time representations discussed in the literature.

§2 Previous Methods of Time Representation in Planning

A representative early planner is the STRIPS system (Fikes *et al.* 1972), in which time is represented implicitly. In this system, the planner keeps a "snapshot" of the state of the world after the application of an operator (an event that changes the world). A sequence of snapshots provides an implicit representation of a particular time sequence investigated by the planner. For efficiency, these snapshots record only the changes caused by the preceding operator and inherit everything else that was true before it was applied. For example, in blocks world, after the application of the PUT-ON(a b) operator, the world is in a new state in which everything is the same except that ON(a b) is true, and CLEAR(b) is false. This change is dictated by the operator's add/delete list.

The inherent contradiction in the fact that the proposition ON(a b) is true and false at different times can be removed explicitly by using a "situational calculus" (McCarthy and Hayes, [1969]). In this representation, every proposition has an extra constant added to it to indicate the situation (state, snapshot) in which it is being asserted. For example, there is no contradiction in asserting both NOT(ON(a b s21)) and ON(a b s32), as the propositions are being asserted in different states. These situational markers can he explicitly time-ordered [e.g. (s32 $\succ$ s21)], thus ordering the corresponding states. If these time orderings are combined with the persistence assumption (i.e., that the truth value of a proposition remains unaltered unless explicitly negated at some latter time), then the truth value of a proposition at any time can be deduced. The inheritance implied by the persistence assumption is an efficient way of representing the world at any time provided that the operators change only a small portion of the world description in a single application.

The situational calculus approach allows any ordering information to be asserted between states, including partial time ordering as well as a strict sequential ordering. This freedom is utilized in parallel (non linear) planners in which a partial time order is imposed only to remove any inconsistencies in the corresponding world. This method has the advantage of producing plans that are as parallel as possible (allowing parallel execution) and do not make any premature commitment to a particular ordering (which may have to be undone as the plan unfolds).

This avoidance of premature orderings of operators (or corresponding states) can be taken a step further. For example, if it is discovered that situation s1 is incompatible with s2, then either (s1 $\succ$ s2) or (s2 $\succ$ s1) may resolve the incompatibility. However, there may be insufficient information at this stage to choose between the possible orderings. Rather than make an arbitrary choice, Eder (1976) suggested imposing the constraint ORDERED(s1 s2), which means that s1 comes before s2 or vice versa. The presence of such constraints makes the deduction of what is true in any given state more difficult, as all possible orderings must be considered. Even without such constraints, the deduction of the truth value of a proposition in a partially ordered time network can be computationally expensive.

The limitations of the situational-calculus approach come from the built-in assumptions of the representation. Because situations are just instants in time, it is impossible to represent quantities that change continuously, such as location or density. Moreover, because this is a state-based representation it is impossible to represent what is happening during an operator application (when the world is changing), since only the initial and final states are given.

§3 A New Time Representation

In this representation, a world (which extends through time), is described by a set of propositions stating the intervals for which other propositions are true. For example, the propositions:

DURING(ON(a b) t1 t2) and DURING(NOT(ON(a b)) t3 t4)

assert that a is on b for the entire time between t1 and t2, and a is not on b between t3 and t4. Additional assertions, such as (t2 $\succ$ t1) and (t3 $\succ$ t2), etc., give the relationship (if known) between intervals. When there is more than one interval assertion for the same proposition (or its negation), overlapping intervals of the same truth value should be merged, while those of opposite truth value must be disjoint (otherwise the proposition would be true and false at the same time). A consistent sequence of such intervals is referred to as the time line of the proposition. A time line may have gaps in it, corresponding to intervals in which the truth value of the proposition is unknown.

This representation follows the example of other authors (e.g. Allen 1981) in making intervals the principal time relation, as this is the one most commonly used in planning. An alternative axiomatization based on adding a time moment or time variable as an extra argument to every proposition has been investigated by the author, but this was found less convenient than the interval representation presented here.

The interval representation makes it more difficult to reason about instants in time. In the interval representation, a moment is represented as a degenerate interval. For example, to assert that the robot is holding a screw at moment t5, the form is

DURING(HOLDING(robot screw) t5 t5)

Quantities that change continuously in time can be represented as explicit functions of universally quantified time variables, possibly constrained to a specific interval. For example, the position of a robot over the interval t1 to t2 can be represented by the rule:

DURING(AT(robot1 location(T1)) T1 T1) $\leftarrow$ (T1 $\succeq$ t1), (t2 $\succeq$ T1).

In other words, the location of robot1 (e.g. as an x-y coordinate pair) is given as a function of time (T1) over the interval t1 to t2. If such functions of time are not given explicitly, they may be constrained by asserting such properties as continuous, monotonically increasing, etc. Although, in general, such constraints on functions do not allow precise prediction of the robot's position at any given time, they can constrain the possible values to a useful range. Since such constraints are applied only to specific functions, this reasoning about functions is still within first-order predicate calculus.

The representation proposed above describes the world over all time for any planning stage. Operators map the current world representation to a new one according to the add/delete lists of the operator. Thus, operators (events) do not appear explicitly in the world description, and so cannot be time-ordered within it. The evolving plan (a partial time ordering of operator applications) is stored separately by the planner and is finished when the corresponding world description satisfies the given goal criteria.

Since a world representation consists of a set of interval assertions (DURING relations), the following rule can be used to decide if a proposition (P) is true over interval T1 to T2:

$$\text{DURING(P T1 T2)} \leftarrow \text{DURING(P T3 T4)}, \quad (\text{T1} \succeq \text{T3}), \quad (\text{T4} \succeq \text{T2}).$$

Other possible relationships (e.g., BEFORE, OVERLAPPING etc.) between two intervals can be deduced by using similar rules.

The use of propositions as terms in DURING propositions causes some difficulties, since they are not allowed as terms in other propositions in first-order predicate calculus. Similar difficulties occur in the use of such predicates as "KNOW" and "BELIEVES" (see Moore 1980 for example).

The main reasons for this ban are as follows. Firstly there is a need to translate expressions involving the DURING predicate applied to a logical compound expression into a compound of DURING propositions (or vice versa). Consider the following examples:

(a) DURING((P and Q) t1 t2) becomes
DURING(P t1 t2) and DURING(Q t1 t2)

(b) DURING((P or Q) t1 t2) becomes
DURING(P t1 t2) or DURING(Q t1 t2) etc.

This difficulty can be largely avoided by insisting that the above transformation rules always be used to create a normal form consisting of a logical combination of DURING relations that do not contain logical compounds (except negation).

Secondly if the proposition within a DURING relation is itself a DURING relation—i.e., DURING relations nested to any level—then there is considerable ambiguity in their interpretation. This difficulty can be avoided by insisting that the predicates within a DURING proposition not include other DURING predicates. This is not an important restriction in planning, but could cause problems if used in a wider context.

One possible method of avoiding such complications is to eliminate the outer DURING relation by adding the interval's beginning and end times to every world proposition as arguments. For example:

DURING(ON(a b) t1 t2) becomes ON(a b t1 t2)

The main difficulty with this method is that, instead of a single rule for deducing the relationship between different intervals (e.g., the last rule mentioned above), a separate rule must be written for each predicate. This option seems impractical for planning.

§4 An Example

Since the above representation makes assertions about the truth value of propositions over intervals, operators can dictate how the world will evolve after they are applied, what conditions must be true before they are applied and any conditions that must be preserved during their application. For example, consider the following well-known blocks world problem, shown in Figure 1.

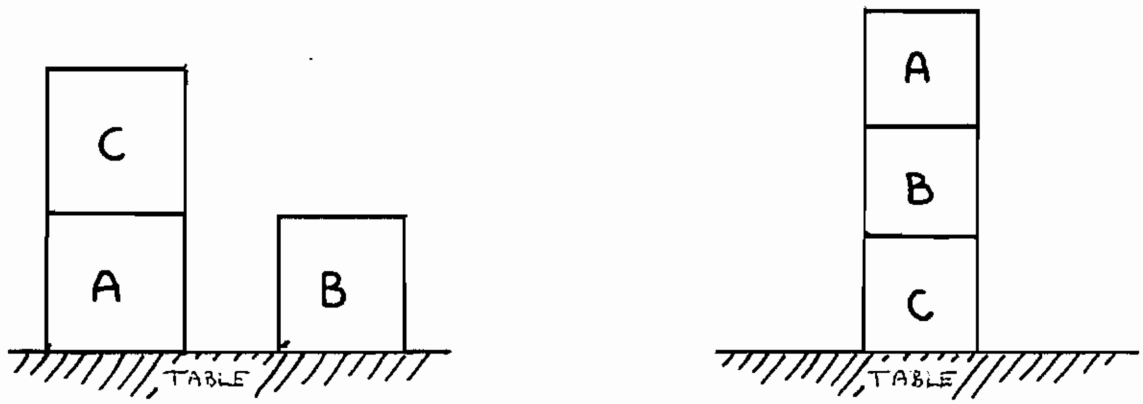

Figure 1

In this problem, the goal can be represented by the description
DURING(ON(a b) t-end t-end) and DURING(ON(b c) t-end t-end),

where t-end is a moment in the future. Similarly, the initial world can be represented by the description

DURING(ON(c a) t-past t-future), DURING(ON(a table) t-past t-future) and
DURING(ON(b table) t-past t-future),

where the first assertion, for example, means that c is on a for all times between t-past and t-future (i.e., two special instants that come before and after any other moment, respectively). A suitable axiomatization of operators for this domain is the following:

PUT-ON(X,Y,T1,T2)—i.e., put X on Y between times T1 and T2.
CONDITIONS:
DURING(NOT(ON(X Y)) T3 T4), T1 $\succeq$ T3, T4 $\succeq$ T2,
DURING(CLEAR(Y) T1 T1)
DURING(CLEAR(X) T1 T2)
DURING(HOLDING(X) T1 T2)
EFFECT:
ADD: DURING(ON(X Y) T2 T5), T5 $\succeq$ T2, T2 $\succeq$ T1.
DURING(NOT(ON(X Y)) T3 T1).
DELETE: DURING(NOT(ON(X Y)) T3 T4).

LET-GO(X T1 T2)—i.e., let go of X between times T1 and T2,
CONDITIONS:
DURING(HOLDING(X) T3 T4), T1 $\succeq$ T3, T4 $\succeq$ T2.
EFFECT:
ADD: DURING(NOT(HOLDING(X)) T2 T5), T5 $\succeq$ T2, T2 $\succeq$ T1.
DURING(HOLDING(X) T3 T1)
DELETE:
DURING(HOLDING(X) T3 T4).

PICK-UP(X,Y,T1,T2)—i.e., pick up X off Y between the times T1 and T2.
CONDITIONS:
DURING(ON(X Y) T3 T4)), T1 $\succeq$ T3, T4 $\succeq$ T2,
DURING(CLEAR(X) T1 T2),
DURING(HOLDING(X) T1 T2).

EFFECT:
ADD: DURING(NOT(ON(X Y) T2 T5)), $T5 \succeq T2$, $T2 \succeq T1$.
DURING(ON(X Y) T3 T2)
DELETE: DURING(ON(X Y) T3 T4).

GRASP(X T1 T2)
CONDITIONS:
DURING(NOT(HOLDING(X)) T3 T4), $T1 \succeq T3$, $T4 \succeq T2$.
EFFECT:
ADD: DURING(HOLDING(X) T2 T5), $T5 \succeq T2$, $T2 \succeq T1$,
DURING(NOT(HOLDING(X)) T3 T1).
DELETE: DURING(NOT(HOLDING(X)) T3 T4).

Some important differences distinguishing the above operators from previous state based operators. In the PUT-ON operator, for example, the last condition requires that the robot be holding X throughout the interval of the operator (T1 to T2), while the second condition requires only that Y be clear at the beginning of the action. Such a requirement cannot be expressed by a single state-based operator. In addition, the effects of the operator are asserted to be true in the world from the end of the operator's application (T2) to some indefinite time in the future (T5). The first condition is a context condition that is usually satisfied and, in the process, finds the interval into which the new effects will be inserted. This interval is assumed to be unique and exist directly in the data base (i.e. not deduced); otherwise the delete list would not update the world description correctly. This assumption can be guaranteed, for example, by ensuring that the time line for every primitive predicate will be minimally described, by joining adjacent intervals of the same truth value.

Operators are rules for changing from one (implicit) world description to another (which is hopefully nearer the goal state). In addition, rules are usually necessary for deducing information within a world description. Some deductive rules for this problem are the following;

DURING(CLEAR(table) T1 T2)—i.e. The table is always clear.
DURING(CLEAR(X) T1 T2) ← DURING(NOT(ON(Y X) T1 T2)—i.e. X is clear at time T if there is nothing on X at this time.
$(T1 \succeq T2) \leftarrow (T1 \succ T2)$
$(T1 \succeq T2) \leftarrow (T1 = T2)$
$(T1 \succeq T3) \leftarrow (T1 \succeq T2)(T2 \succeq T3)$
DURING(P T1 T2) ← DURING(P T3 T4), $T1 \succeq T3$, $T4 \succeq T2$.—where P is any proposition.

The first two rules allow the deduction of the truth value of a CLEAR predicate over any time interval. It is because CLEAR is a deduced predicate that it is not included in the ADD list of the PICK-UP operator, since its truth value over any time interval can be deduced when needed.

As an example of this representation, consider the case in which the operator PICK-UP(c a t1 t2) is applied to modify the world representation from

DURING(ON(c a) t-past t-future)
to $(t2 \succ t1)$, DURING(ON(c a) t-past t1), DURING(NOT(ON(c a)) t2 T5), $(T5 \succeq t2)$.

The proposed world representation makes it possible to answer any queries about the state of the world by means of normal PROLOG-style backward chaining. If, for example, it has been asserted that (t1 ≻ t21) in the above, then DURING(ON(c a) t21 t21) will evaluate to true. However, if no ordering of t1 and t21 is given, the proof will fail. If the closed-world assumption is made for the ON predicate, this proof failure means that this predicate is false. Note that the truth value of DURING(ON(c a) T T) is undefined if T is between t1 and t2, since the truth value changes during this interval.

The freedom to constrain time instants such as t1, t2, etc., in a world representation can be used to express commonly used time information. For example, if the PICK-UP operator is known to take 10 time units, t2 can be replaced by (t1 + 10); thus, the end time of this operator will always be 10 time units after it begins. Uncertainty about such time intervals can also be represented by constraining t2 with such constraints as (t2 ⪰ tmin) and (tmax ⪰ t2). The time instants can also be assigned to particular clock times by assertions such as occurs-at(t1 10:15:43), occurs-at(t2 10:19:21) and the like. This absolute time information allows deduction of such relationships as (t2 ≻ t1), provided that built-in predicates to decide the order of chronological or calendar times are available.

Although the time line of a particular proposition must be totally ordered (otherwise the world is incompletely specified and possibly inconsistent) and reduced (e.g., adjacent time intervals merged), the relationship between instants from different propositions do not have to be specified. The result of this freedom is that inter propositional time relationships can range from completely unspecified to totally ordered, with partial ordering being the most common.

A variant of the above representation is to add an extra constant to every DURING proposition to indicate in which world the proposition is being asserted. This method, similar to the use of situational markers in state-based representations, makes all previously examined worlds explicit. In the implicit representation above, only the current world is stored. All previous worlds have been over written, and can be reconstituted only by undoing the sequence of operators that has been applied.

§5 Deducing Time Relationships in the New Representation

A planner usually works by repeating the following cycle. The simplified version given here, however, ignores such problems as the bookkeeping necessary for backup, what to do if a goal evaluates to unknown, and methods of assessing progress on different OR branches.

Step 1—Evaluate all new goals in the context of the current world description, adding any that are not true to the list of unsolved goals.

Step 2—From the list of unsolved goals, select one as the next target goal. If this list is empty, the problem has been solved.

Step 3—Select an operator to achieve this goal (with other possible side effects). Alternative choices are stored as possible backup points. Delete the selected goal from the list of unsolved goals. If there are no remaining operators for this goal, and it is one of the given goals, then the problem is unsolvable (since further backup is impossible). Otherwise, undo the operator that generated the current goal and go to Step 2.

Step 4—Insert the effects of the chosen operator into the previous world description and check as to whether this generates any inconsistencies. Attempt to remove any such inconsistencies by imposing a minimal time ordering. If any inconsistencies cannot be removed, undo this operator application and go to Step 2.

Step 5—Add the preconditions of the operator to the new goal list, and go to Step 1.

The new representation proposed in the preceding section creates some problems when used in the above planning cycle. One problem is that a goal (which has been deduced to be false in the current world), is sometimes best replaced by its logically equivalent form. For example, an initial goal in the blocks-world problem is DURING(ON(a b) t-end t-end). This goal is achievable by the PUT-ON(a b T1 t-end) operator (i.e., the goal becomes true immediately the operator is applied), but greater planning flexibility is available if the goal is first replaced by its logical equivalent:
DURING(ON(a b) T3 T4), t-end $\succeq$ T3, T4 $\succeq$ t-end.
The result of this substitution is that when the PUT-ON operator is unified with a new goal, the start and end times of PUT-ON are free variables, which can be bound to any particular moment. This freedom to pin down floating instants is very useful in resolving time conflicts.

A more serious problem arises when we seek the reasons a particular goal cannot be established in a particular context, so that the abortive proof attempt can suggest additional time orderings that would enable the proof to succeed. For example, the world representation given in the previous section can be depicted by the time lines in Figure 2. If the goal DURING(CLEAR(a) t3 t-future) (shown in Figure 2) is evaluated in this context, the proof attempt will fail because no relationship between t3 and the previous time line has been given. However, it is obvious from the figure that the additional assertion (t3 $\succeq$ t2) will let the proof succeed. Methods of proof failure analysis (enabling proofs to succeed) have yet to be fully investigated.

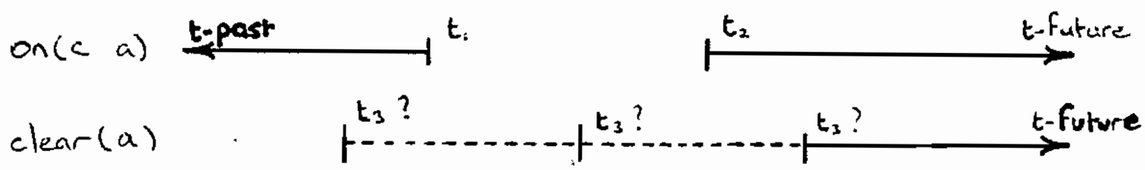

Figure 2

A fundamental problem in planning is that of truth maintenance (see Doyle [1979]), which is part of the difficulty in using a non-monotonic logic—i.e., a logic in which the addition of new axioms affects previous proofs. For example (from the preceding section), the goal DURING(CLEAR(b) t1 t1)—which is generated by a precondition of the operator PUT-ON(a b t1 t-end)—is deduced to be true, because at this stage PUT-ON(b c t2 t3) has not been applied. However, when this operator is applied, the previous proof is rendered incorrect; consequently, methods must be provided for detecting when previously true propositions are affected by new information inserted into the world representation. One solution is to provide "If-added demons"—i.e., rules that instruct the planner to re-evaluate specific propositions if new information matching the pattern of the rule is added to the world representation. Such rules should be generated whenever vulnerable facts are used in the proof of any proposition, so that such facts can be monitored for future violations.

§6 Comparison with Other Time Representations

Many authors have recognized the deficiencies in the STRIPS-style time representation and have proposed new representations (and associated inference procedures) to provide a more complete time representation. In particular, most representations are concerned with the relative time orderings of events, including their relationship to the special moment "now". However, the representation proposed above, is designed to maintain a consistent world description (model) in which events (roughly equivalent to operator applications) appear only when a world model is being changed, and so are not represented explicitly.

An early attempt to design a powerful time representation is given by Kahn and Gorry (1977), who argue for a time specialist—a program to store and answer questions on time relationships among events given to the specialist program. This division of labor is questionable, as the preceding section shows the advantages of integrating the reasoning about time relationships with the basic reasoning system used by the planner. For example, the deduction as to whether an object is clear over a given interval can also reveal what additional time relations are necessary to make it true if the deduction fails, thus integrating the deduction of the truth value of the proposition and its relation in time. A further limitation is Kahn and Gorry's representation of events as instants of time, which precludes the represention of continuous events.

The approach pioneered by Kahn and Gorry has been extended by Allen (1981). Allen's time specialist has changed from assuming the representation of events by time instants to the assumption that they are represented by intervals. There are nine possible ways two intervals can relate to each other (compared with three for instants of time); thus the rules (axioms) needed to deduce relationships between any two arbitrary events are more complex. A difficulty with Allen's computational method is that it requires that all time relations be recomputed whenever a new time relationship is entered. This is inefficient, since many of these relationships are never used. The computation of particular time relationships on demand, as presented in this paper, eliminates this difficulty.

Allen's approach has been further developed by Vilain (1982), who showed that computing arbitrary time relationships entails computation of the transitive closure over the set of all given time relations between event intervals. This computation can be performed in n^3 time and space, where n is the number of intervals about which assertions have been made. In large planning tasks, this computational cost can be excessive. Vilain extended the interval-based logic to include both time instants and absolute times.

The representation presented in this paper and its associated inference procedures exhibit a linear worst-case behavior in deciding if a particular proposition (either at an instant or over a given interval) is true or false, where the dependence is on the number of sub intervals on the time line of the particular proposition. Depending on the completeness of the (partial) time orderings, it is a more complex deduction to decide if a relationship between two time intervals (for different propositions) is say "DURING".

Fortunately, the basic planning cycle above, rarely requires the deduction of inter-interval relations between propositions, since the most common time deduction is deciding the relation between a new proposition (and associated time interval) and the existing time line for the same proposition. An example of inter propositional time deduction is shown in Figure 2. Such deductions involve proving the existence (or otherwise) of a path in a partially ordered graph of binary time relations between the relevant propositions. In the worst case, the complexity of this proof is on the order of the number of inter propositional time relations in the graph.

McDermott (1981) has developed a complex logic for representing time. This logic can be regarded as a state-based representation carried to a continuous limit. That is, a particular world history (and future) is regarded as an infinite totally ordered sequence of states. In this representation, a fact is the set of states in which that fact is true. Furthermore, the sequence of states representing a particular world history is allowed to split into multiple futures (and pasts).

A major motivation for McDermott's representation is that he does not believe it possible to capture an event (something that is happening) by representing the changes of fact that take place. For example, the event "John ran around the track three times" is not represented by the change of fact "John is more tired" and "John has a memory of running around the track three times." Because the representation presented in this paper is fact-based, (since events [operators] are not explicitly represented in the world model, but rather at a metalevel at which the plan is being built), it is necessary to respond to this critique.

The simplest reply is that a fact-based representation should represent the world during the event in a way that reflects what is actually happening, not, say, the level of fatigue and memory content after the event is over, as suggested by McDermott. A possible representation of McDemott's example is as follows:

DURING(running(John) t23 t26)—i.e., John is in a running state between times t23 and t26.
DURING(at(John location1(T2)) T2 T2) $\leftarrow$ (T2 $\succeq$ t23)(t24 $\succeq$ T2)
DURING(at(John location2(T3)) T3 T3) $\leftarrow$ (T3 $\succeq$ t24)(t25 $\succeq$ T3)
DURING(at(John location3(T4)) T4 T4) $\leftarrow$ (T4 $\succeq$ t25)(t26 $\succeq$T4)— i.e., John's location
is given as a function of time, for each lap.
DURING(tiredness(John fatigue-level[T5])T5 T5) $\leftarrow$ (T5 $\succeq$ t23) (t26 $\succeq$ T5)
i.e., John's tiredness is an explicit function of time.

This representation could be expanded to include constraints and continuity requirements upon the various functions given, as well as to indicate that the change in location is caused by running, etc., but the main point in this representation is that a fact-based representation has captured the essence of the "John ran around the track three times" event without mentioning the latter at all.

Another major concern of McDermott is the representation of causality. The representation described in this paper is capable of representing causality by introducing the notion of involuntary operators. These are operators that automatically apply whenever their conditions are true. For example, a gravity operator applies to a object whenever its support is removed. When such operators are present, it is possible to use them in plans by deliberately creating conditions in which they apply if their effects are desired. However, deciding whether any such involuntary operators become applicable in any world representation is a computationally expensive process.

Many of the difficulties discussed by McDermott are avoided in the representation given here. In particular, by keeping one (active) world description at any time, we ensure that other, previously inconsistent worlds will not cause conflicts. Trying to represent the current and all previously considered worlds in a single representation, can lead to a more complex logic.

A planner implemented by Vere (1981), has the most general time representation of any planner known to the author. This planner is similar to NOAH (Sacerdoti, 1977), and NONLIN (Tate, 1977), but it allows operators to have durations (which may be a function of the operator arguments), and "windows"—i.e., an interval in absolute time. This is an event (operator)-based time representation, in which the preconditions are assumed to remain true until the end of the operator application. Consequently, in this representation the only way to specify preconditions that must be true at some time during the operator application is to split the operator into separate successive suboperators for which the persistence assumption holds. In the examples investigated by Vere, this representation was sufficient.

Another author concerned with the representation of time, and whose ideas strongly influenced this paper is Schubert (1976). He argues the case for making time a privileged concept over, say, location by building time information into every predicate. Schubert also shows that, when the truth value of a predicate is dependent on a particular context, the context information can be represented by corresponding time information. For example "Mary is livelier with her lovers than with her parents" in the representation of this paper becomes

> DURING(liveliness(Mary level1) T1 T2) ← DURING(with(Mary lover(Mary)) T1 T2)
> DURING(liveliness(Mary level2) T3 T4) ← DURING(with(Mary parents(Mary)) T3 T4)
> greater(level1 level2)
> i.e., the first rule says that Mary has liveliness of level1 if she is with one of her lovers in the interval T1 to T2, while the last assertion states that the first level of liveliness is greater than the second.

§7 Summary

The representation of time presented in this paper is designed to overcome the inability of the standard planning representation to represent continuous changes and conditions over intervals. This is achieved by asserting every world proposition to be true during a given interval (represented by its endpoints). In this representation, an operator transforms the previous world description into a new one as specified by the add/delete list of the operator. A world description is over all time, not just at a moment as in former state-based planning systems. This interval representation allows instants to be represented by degenerate intervals of zero duration. The relationship among the intervals of different propositions can be specified to the degree to which they are known, thus allowing partial time orderings.

This representation has the advantage of integrating the reasoning about time with all other reasoning about what is true in the world. This occurs because the same representation and inference procedure— (PROLOG-like backward chaining)—are used for all inference, thus saving the need for a separate time expert as advocated by other authors.

In the representation proposed in this paper, the time information is associated with each proposition and not linked directly with events (or operator applications). Any propositions logically dependent on others will have interpropositional constraints between their time intervals (a partially ordered time network). Most of the cost in deciding the truth value of a particular proposition at a specific moment (or over a given interval) comes when the partially ordered time network has to be searched. This search has a worst-case cost proportional to the number of time orderings in the network, compared with an n^3 cost for event-based representations.

This new representation meets the criterion of expressiveness (i.e., any time relationship [or value] can be expressed) without overconstraining the given time information. By simply attaching chronological or calendar times to instants, absolute as well as relative time information can be expressed and reasoned about. The proposed representation can also represent intervals of uncertain length as well as those of known duration.

References

Allen, J. F., "An Interval-Based Representation of Temporal Knowledge", Proc. International Joint Conf. on Art. Intell., Vancouver pp221-226 1981.

Doyle, J., "A Truth Maintenance System", Artificial Intelligence 12, 3, Nov. 1979.

Eder, G., "A System for Cautious Planning", Research Report No. 27, Dept. Art. Intell., Univ. of Edinburgh. 1976.

Fikes, R. E., Hart, P. E., and Nilsson, N. J., "Learning and Executing Generalized Robot Plans", Artificial Intelligence, 3, 4, pp.251-288 Winter 1972

Kahn, K. and Gorry, G. A., "Mechanizing Temporal Knowledge", Artificial Intelligence, 9 pp.87-108, 1977.

McCarthy, J. and Hayes, P., "Some Philosophical Problems from the Standpoint of Artificial Intelligence", Machine Intelligence, Vol. 4, B. Meltzer and D. Michie, eds., American Elsevier, N.Y. pp. 463-502, 1969.

McDermott, D., "A Temporal Logic for Reasoning About Processes and Plans", Research Report 196, Dept. of Computer Sci., Yale Univ. 1981

Moore, R. C., "Reasoning About Knowledge and Action", SRI Artificial Intelligence Tech. note No. 191, Oct. 1980.

Sacerdoti, E. D., "A Structure for Plans and Behavior", Elsevier Computer Science Library, 1977.

Schubert, L. K., "Extending the Expressive Power of Semantic Networks". Artificial Intelligence, 7, 2, pp163-198 Summer 1976.

Tate, A., "Generating Project Networks", Proc. Fifth International Joint Conference on Artificial Intelligence, MIT, Aug. 1977.

Vere, S. A., "Planning in Time: Windows and Durations for Activities and Goals", Research Report Jet Propulsion Lab. Nov. 1981.

Vilain, M. B., "A System for Reasoning About Time", proc. American Assoc. for Artificial Intelligence, Pittsburgh, Aug. 1982.

Warren, D. H. D., Pereira, L. M. and Pereira, F. "Prolog—The Language and its Implementation Compared with Lisp." In SIGPLAN/SIGART Newsletter. ACM Symposium on A.I. and Programming Languages, August 1977.