Part II
Temporal Reasoning

Introduction to Part II

1 INTRODUCTION

Reasoning about time is an essential competence that all humans possess and a signature of intelligent behavior in any cognitive system. Our ability to represent temporal knowledge of actions and events in the world is essential for modeling causation, constructing complex plans, hypothesizing possible outcomes of actions, and almost any higher-order, cognitive task. It is not surprising, therefore, that temporal reasoning has been a central area of research in artificial intelligence since the 1960s. This overview of Part II will situate some of the work on temporal reasoning, particularly certain aspects that are especially relevant to natural language processing. As we have seen in Part I, natural language expresses temporal information through tense, aspect, temporal adverbials, and other devices. Motivated by linguistic considerations, Part I introduced constraint- and logic-based formalisms for analyzing tense, and event ontologies for representing event classes and structure. These mechanisms were applied in some cases to identification of the temporal location of events mentioned in text. We now turn from linguistic considerations to considerations motivated by artificial intelligence in general.

The problems of temporal reasoning involve in part, as in natural language, locating events in time. Thus, given the narrative,

(1) a. Yesterday, John fell while running.
    b. He broke his leg.

a natural language system may seek to anchor the *falling, running*, and *breaking* events to the particular time (yesterday), as well as order the events relative to each other, e.g. the running precedes the falling, which precedes the breaking. Temporal reasoning is therefore concerned with representing and reasoning about such anchoring and ordering relationships. Temporal reasoning is also concerned with creating the most appropriate formalisms for representing events, states, and their temporal properties. However, the particular form of temporal representation depends on the type of reasoning problem under consideration. In common-sense inference, for example, knowing that the falling occurred before breaking, and that the falling occurred yesterday (facts obtained here from linguistic data), along with common-sense knowledge of the behavior of breakings and fallings, may allow a system to infer that the falling precedes and causes the breaking, and that these events occurred yesterday. In planning, on the other hand, a given outcome is desired (e.g. a robot arriving at a crater), and common-sense knowledge of the behavior of events and states (such as landing, avoiding obstacles, etc.) may allow one to infer what needs to happen when.

Historically, temporal reasoning developed largely out of the planning community in AI, but the ability to perform temporal reasoning in knowledge-intensive environments is critical for many diverse applications. Some of these are listed below:

1. Maintaining temporal consistency in a knowledge base;
2. Temporal question-answering (cf. Androutsopoulos et al. 1998);
3. Scheduling tasks and events; and
4. Causal diagnosis.

There are four features that typically distinguish the structure of time within a model of temporal reasoning:

1. Primitive time unit: the choice of instants (i.e. points) or intervals (i.e. periods) as temporal markers of the flow of time.
2. Branching: whether different time-lines are possible or not.
3. Discreteness: the choice of whether to represent time as a collection of discrete elements, or else as an element between two points.
4. Boundedness: whether time is infinite or finite in each direction.

Each of these issues is addressed in distinct ways in the frameworks represented in this part. Most early approaches to AI, beginning with the frameworks (to be discussed below) of the Situation Calculus (McCarthy and Hayes 1969), the Event Calculus (Kowalski and Sergot, Chapter 10), and also early work in planning (McDermott, Chapter 9), assumes instants to be basic. Other studies, including the papers in this part by Allen and by Hobbs and Pustejovsky, take intervals to be basic.

A flow of time that is a strict partial ordering is said to be linear if any two distinct points are related. A branching model of time can be said to be branching to the future if there is some point that has two unrelated points in its future. Similar remarks hold for the flow of time in the past. When a structure does not branch to the future or the past, it is called a nonbranching (or linear) structure of time. Typically, model-theoretic treatments of tense and aspect in language assume a nonbranching past and possible branching futures. Dowty (1979), for example, invokes the nonlinearity of intensional states inherent in a Montague-style treatment of the progressive, e.g. 'John is crossing the street.' The future branches to many world-time pairs, some of which are inertially closer to achieving the goal state than others. Irrealis contexts also typically require some form of branching future (cf. van Bentham 1983).

Regarding discreteness of time: a dense flow of time exists when, between any two distinct points, there is a third point. A discrete flow supposes that time proceeds in discrete steps, where there is a well-defined next point or period. Finally, the feature of boundedness addresses the issue of the beginning- and end-points in the structure modeling the flow of time. Typical models assume a bounded past and an unbounded future.

The goals of an approach and the specifics of a particular reasoning task will often dictate the choice of one's temporal primitives. For example, from a logical standpoint, instants are attractive since we understand the idea of truth at an instant, but the notion of truth at an interval requires further explication. Dean and McDermott (1987) argue that instants are more efficient for reasoning within temporal database management systems. For reasoning about continuous change, as Galton shows in this part (Chapter 13), we need a notion of durationless events, thus arguing for instants. Galton's paper shows how it is possible to represent intervals in terms of points or have a point-based theory that represents intervals. The differences between point-based and interval-based models have been discussed in considerable detail, for example, in van Bentham (1983) and Shoham (1989).

Each of the temporal reasoning formalisms represented here differs in its expressiveness and efficiency for reasoning about natural language problems. The requirements of a particular reasoning task will largely dictate the topology of time needed, and determine the appropriateness of the data structures and inference rules used. Although early work on natural language was formulated in terms of the basic framework of the Situation Calculus (as with McDermott's paper), most work on the temporal interpretation of text and discourse has assumed some sort of Allen-style interval calculus. The expressiveness inherent in the interval calculus has meshed well with much of the recent work on tense and aspect interpretation in linguistic semantics (e.g. Leith and Cunningham, 2001).

2 TEMPORAL LOGIC

A temporal logic allows one to use the representation and inference mechanisms of logic to reason about time. For this to happen, temporal information needs to be added to the logic. From a logical standpoint, there are two ways to provide for a temporal interpretation of a proposition:

1. Add a modal operator over the propositional expression, so that temporal order is interpreted from the syntactic combination of operators over expressions;
2. Add an additional argument to the predicative expression, one representing time directly as a point or period, or as a time-dependent individual, such as an event.

The latter approach will be discussed below under the Situation Calculus (Section 3). In Part I, we introduced an instance of the first approach, that taken by Prior in the construction of **Minimal Tense Logic**, known as K_t. In such systems, operators play the combined role of verbal tense, temporal adverbials, as well as temporal prepositions and connectives. For K_t, four axioms form the core knowledge about temporal relations (as already stated in the Introduction to Part I):

a. $\phi \rightarrow H\ F\phi$: What is, has always been going to be;
b. $\phi \rightarrow G\ P\ \phi$: What is, will always have been;
c. $H(\phi \rightarrow \psi) \rightarrow (H\phi \rightarrow H\psi)$: Whatever always follows from what always has been, always has been;
d. $G(\phi \rightarrow \psi) \rightarrow (G\phi \rightarrow G\psi)$: Whatever always follows from what always will be, always will be.

F and P are usually referred to as **weak operators**. They can be defined in terms of the other two as follows: $F\phi = \neg G \neg \phi$; $P\phi = \neg H \neg \phi$. K_t becomes a complete inference system when the two rules of temporal inference below are added to the rules of propositional logic:

(3) a. From a proof of ϕ, derive a proof of $H\phi$.
b. From a proof of ϕ, derive a proof of $G\phi$.

To reason about the truth of expressions in a propositional tense logic, we construct a model, where our interpretation functions (or valuations) make reference to moments of time. Let us define a **temporal frame** as consisting of T, a set of moments in time, and an ordering relation R, the *earlier than* relation. The truth value of an atomic formula ϕ can be determined relative to a frame, according to the following rules:

(4) a. $V_{M,t}(\mathbf{G}\phi) = 1$ iff for every $t_i \in T$ such that tRt_i: $V_{M,t_i}(\phi) = 1$
b. $V_{M,t}(\mathbf{F}\phi) = 1$ iff for some $t_i \in T$ such that tRt_i: $V_{M,t_i}(\phi) = 1$
c. $V_{M,t}(\mathbf{H}\phi) = 1$ iff for every $t_i \in T$ such that t_iRt: $V_{M,t_i}(\phi) = 1$
d. $V_{M,t}(\mathbf{P}\phi) = 1$ iff for some $ti \in T$ such that t_iRt: $V_{M,t_i}(\phi) = 1$

Examples of the tense logic as applied to natural language sentences are given below:

(5) a. John will have left Boston.
$\mathbf{F}(\mathbf{P}(leave(j, b)))$
b. John was going to leave to Boston.
$\mathbf{P}(\mathbf{F}(leave(j, b)))$

Using the model described, we can determine the truth-conditions of the propositional expression for each sentence above. For example, for (5a), we have the following valuation:

(6) a. $V_{M,t}(\mathbf{P}(leave(j, b))) = 1$ iff for some $t_i \in T$ s.t. t_iRt: $V_{M,t_i}((leave(j, b))) = 1$
b. $V_{M,t}(\mathbf{F}(\mathbf{P}(leave(j, b)))) = 1$ iff for some $t_i \in T$ s.t. tRt_i: $V_{M,t_i}((\mathbf{P}(leave(j, b)))) = 1$

Intuitively, this states that the expression is true if, at some moment of time, t_1, after now (t_0), there is a moment of time, t_2, before t_1, such that 'John leaves Boston' is true. A problem with K_t, as already pointed out in Part I above, is that there is no explicit notion of the present. Notice that in the model above, we just assumed that the base from which we are performing the valuation is conveniently assumed to be the present. This will not be sufficient when we need to model a more dynamic and expansive notion of the present for reasoning tasks.

2.1 Extensions to tense logic

There have been many extensions and modifications to the basic form of the Propositional Tense Logic of System K_t. One major addition was introduced by Kamp (1968), namely the binary temporal operators **S** (since) and **U** (until).

(7) a. $\mathbf{S}\phi\psi$: ψ has been true since a time when ϕ was true.
b. $\mathbf{U}\phi\psi$: ψ will be true until a time when ϕ is true.

These operators have become standard within computer science in the area of temporal database reasoning systems (as discussed in Manna and Pnueli 1992, and Baudinet et al. 1993), where persistence of database updating functions over relations can be modeled with **S** and **U**, e.g. 'Smith has been manager of Dept. A since Smith was promoted to manager.'

In the 1970s, temporal logic was adopted by computer scientists working in program verification and specification as a standard methodology for program analysis. Pnueli (1977) is one of the first major works in this area, and the growing importance of temporal logic, interval temporal logic, and other extensions to tense logic, is seen in their role in modeling reactive and hybrid systems in computer science (cf. Gabbay et al. 1995 for extensive discussion).

An example of this is **Interval Temporal Logic** (ITL), which is a notation for both propositional and first-order reasoning about periods of time as used in descriptions of hardware and software systems. ITL is able to handle both sequential and parallel composition and offers proof techniques for reasoning about program properties involving safety and liveness. Safety states that something bad will not happen, while liveness properties assert that something good will eventually happen.

3 SITUATION CALCULUS

Perhaps the most widely adopted attempt to model action and change in the early days of AI was the situation calculus (McCarthy 1963, 1968; McCarthy and Hayes 1969). This model represents actions and their effects on the world. The world is represented as a set of situations, which model the possible configurations of the world at a particular time. In this sense, there is a strong similarity between possible worlds (Carnap 1947) and situations, although no semantics for the latter was spelled out in the early work. Fluents are time-varying properties of individuals. Actions are functions that map states to states, and hence act as state transformers.

The situation calculus was used for many different tasks, but was particularly popular in planning paradigms. The major problems with the classic situation calculus are two fold: (1) concurrent actions cannot be represented; (2) there is no representation for the duration of actions or delayed effects of actions. These problems make the pure SC inadequate for many reasoning tasks, but it has been extended and enriched by numerous researchers and is still a very active area of research (e.g. Reiter 2001).

There are typically two strategies employed for representing the situation calculus within a first-order logical representation (FOL): the use of temporal arguments and the use of metalanguage predicates. The first approach is similar in many respects to Davidson's proposal (1967) for event individuation of predicates; in the case of the Situation Calculus, a state variable is added to every predicate in the language.

The state-based (temporal argument) representation of the Situation Calculus (where temporally-sensitive variables are employed) interprets events as state transform functions. Beginning and end states are characterized as predicates with state variables added. As mentioned above, actions cause state transitions. For example, the state-based representation of sentence (8),

(8) John gave *Lord of the Rings* to Mary.

is as illustrated in (9).

(9) a. *Have*(*s*1, *J, LOTR*)
b. *Have*(*s*2, M, LOTR)

The initial state is changed by the application of a state transformer, *give*, modeled as an initiation rule.

(10) *Have*(*z, y, Result*(*give*(*x, y, z*), *s*)))

The second approach to interpreting the Situation Calculus involves the use of metalanguage predicates, which relate the truth value of an expression to a situation. Taking the example in (8) again, this would entail the following representation:

(11) a. *HOLDS*(*Have*(*J, LOTR*), *s*1)
b. *HOLDS*(*Have*(*M, LOTR*), *s*2)

The *HOLDS* predicate states that a property is true in a specific situation. States are changed by *Initiate* and *Terminate* actions. The way to express the *Initiate* action in this interpretation of SC would be as follows:

(12) *HOLDS*(*Have*(z, y), *Result*(*give*(x, y, z), s))

This interpretation of the Situation Calculus is more expressive than the temporal argument approach, since it effectively creates a second-order representation where predicates can be quantified over. This allows for the treatment of some of the classic problems in aspectual semantics (see Introduction to Part I).

There are some problems with the Situation Calculus that do not disappear under either interpretation outlined above. Chief among these is the fact that situations must be totally ordered, making planning difficult. Frame axioms, rules characterizing the persistence and effects of actions, as we will discuss in Section 6 below, are trivial to express in the Situation Calculus; the number of such rules, however, becomes prohibitively large, as they are proportional to the products of the number of fluents and actions performable over them. In addition, events are not explicitly represented in the model.

3.1 McDermott's Use of the Situation Calculus

McDermott's paper, 'A Temporal Logic for Reasoning about Processes and Plans', (Chapter 9) starts with the general assumptions of the Situation Calculus, but with some important modifications. McDermott frames his contribution as a temporal model with a first-order language, in the spirit of Moore (1980) and Hayes (1979). McDermott assumes the world is defined as discrete situations (states) associated with a date. **Chronicles** are states that are coherently structured into a possible history. McDermott then goes on to define a **fact** as a set of states wherein a particular proposition is true.

He abandons the classical Situation Calculus (cf. McCarthy (1968)) notion of events as fact changers, and defines an **event** as a set of intervals over which a proposition is minimally true (i.e. it happens once). This will be essentially the same intuition that Allen adopts for his definition of event. McDermott's model assumes points as primitive, where intervals can be constructed from a totally ordered convex set of states. This model is continuous (dense time) and has a branching future.

4 EVENT CALCULUS

A somewhat different approach to representing events is taken by Kowalski and Sergot in their paper, 'A Logic-based Calculus of Events' (Chapter 10). This work was originally intended as a model for database update and narrative understanding, but has developed into a richer framework for general issues in temporal reasoning and planning. The main innovation in this model comes in the way that events are represented. Whereas events are viewed as transformers from state to state in the Situation Calculus, they are primitives in the Event Calculus, acting as *updates* on the state of the world. In this sense they are *additive* information operations. More specifically, they are seen as actions that initiate or terminate the properties of individuals (known as fluents, adopted from the Situation Calculus). As in the Situation Calculus, Kowalski and Sergot introduce a *Holds* predicate, which expresses that a property or relationship associated with a specific period of time is true during that period. Hence, for some property, *P*:

(13) $Holds(P)$

From this, they define the *HoldsAt* relation, which expresses that a relation, *u*, holds at a specific time instant, *t:*

(14) $HoldsAt(u, t)$

To illustrate, let us return to the example from the previous section, of John giving *Lord of the Rings* to Mary. Intuitively, the initial and goal conditions can be expressed as below.

(15) a. $HoldsAt(Have(J, LOTR), t_1)$
b. $HoldsAt(Have(M, LOTR), t_2)$

The *Hold* predicate states that a property is assumed to persist until the occurrence of some event interrupts this property. This is referred to as default persistence. For example, the event of *giving*, e_0, terminates the relation holding at t_1 and then initiates that relation holding at t_2.

(16) a. $terminates(e_0, Have(J, LOTR))$
b. $initiates(e_0, Have(M, LOTR))$

Time in this model is represented as a partially ordered set of points, and the occurrence of an event is represented by associating it with the time-point at which it occurs. This is accomplished by a metalanguage predicate *happens*, where *e* is an event instance and *t* is a time-point:

(17) $happens(e, t)$

Reasoning in the Event Calculus entails deriving the maximal validity intervals (MVIs) over which properties hold, as the result of the actions of events. An MVI is maximal if it cannot be properly contained in any other valid interval.

The paper by Chittaro and Combi (Chapter 11) extends the framework of the Event Calculus to allow for the representation of events with indeterminant temporal anchoring and granularity. They introduce a framework they call the **Temporal Granularity and Indeterminacy Event Calculus** (TGIC) to model these properties. This entails modifying the algorithm for computing maximal validity intervals to allow satisfaction under varying levels of granularity of temporal scale (years, months, days, hours, etc.). This flexibility extends the expressiveness of the Event Calculus to accommodate a more general concept of event. This is illustrated with examples from a clinical domain, showing how varying granularities of events using this procedure can facilitate reasoning.

It should be pointed out that the Event Calculus shares with the Situation Calculus the basic concepts of property initiation and termination, and these similarities are discussed in Kowalski (1994). There are some major differences, however, including the following.

1. The Situation Calculus makes use of branching time while the Event Calculus used linear time.
2. The SC has a notion of previous state that is absent from the EC, by virtue of the explicit use of situations.
3. State transitions in the SC are functions but not in the EC.

These are explored in more detail in van Belleghem et al. (1995).

5 ALLEN'S TEMPORAL INTERVAL ALGEBRA

One of the most important works in the area of temporal representation and reasoning is James Allen's article, entitled 'Towards a General Theory of Action and Time' (1984); see Chapter 12. In this system, temporal intervals are considered primitives and constraints (on actions, etc.) are expressed as relations between intervals. There is no branching into the future or the past. In Allen's interval algebra, there are thirteen basic (binary) interval relations, where six are inverses of the other six, excluding equality.

(18) a. before (b), after (bi);
b. overlap (o), overlappedBy (oi);
c. start (s), startedBy (si);
d. finish (f), finishedBy (fi);
e. during (d), contains (di);
f. meet (m), metBy (mi);
g. equality (eq).

These are shown schematically in Figure 1 below.
The reasoning system is supported by a transitivity table, which defines the conjunction of any two relations. All thirteen relations can be expressed using *Meet*. For example, for two periods *i* and *j* we can define the relation *Before* as follows:

(19) $Before(i, j) =_{df} \exists m[Meets(i, m) \wedge Meets(m, j)]$

5.1 Interpreting intervals

Allen makes a basic distinction between properties, which we have already encountered, and *occurrences*, which are inspired by Davidson's theory of actions and events (Davidson 1967).

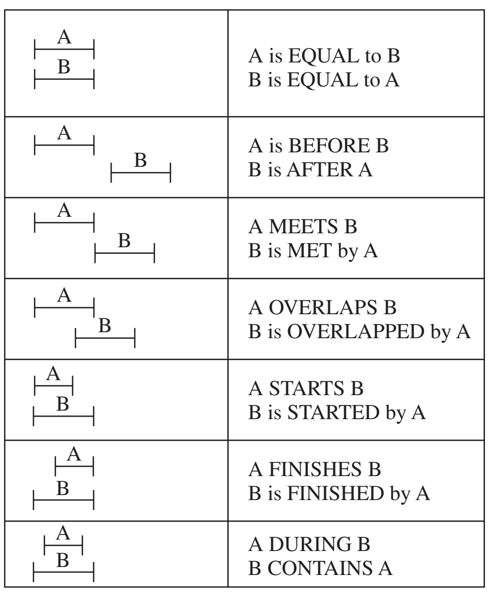

FIG. 1

Occurrences are a sort of generalized eventuality category, which divide into two classes: **processes** and **events**.

He then defines metalanguage predicates for each of these three classes.

CLASS	PREDICATE
property	*HOLDS*
event	*OCCUR*
process	*OCCURRING*

HOLDS is used to assert that a property is true during a specified temporal interval.

(20) $HOLDS(p, t)$

is said to be true if and only if the property p holds during the temporal interval t. This predicate is defined in terms of a downward monotonic subinterval property, shown below:

(21) $HOLDS(p, T) \leftrightarrow \forall t\ [IN(t, T) \rightarrow HOLDS(p, t))]$

This cannot be said of either kind of occurrence, however. For notice that events do not have homogeneous behavior for the properties which live on the interval defining that event. For example, if John dies, then there is a change of state which, as in the situation and event calculi, has to be represented as an opposition of properties. Allen introduces the *OCCUR* relation for events, and defines it as follows (where Allen uses the expression P_t to refer to the necessary and sufficient set of conditions for an event's occurrence in t).

(22) a. $OCCUR(e, t) \wedge IN(t', t) \rightarrow \neg OCCUR(e, t')$
b. $OCCUR(e, t) \leftrightarrow P_t \wedge \forall t'[IN(t', t) \rightarrow \neg P_{t'}]$

Similarly, because of issues of granularity, even processes are not completely homogeneous in the way that properties are. Imagine the process of playing a piano. Those subintervals when the keys are not being struck are not actual piano playings, hence we need to have a weaker relation interpreting such classes. This is called *OCCURRING* and is defined as follows.

(23) $OCCURING(p, t) \rightarrow \exists t'[IN(t', t) \wedge OCCURRING(p, t'))]$

5.2 Discussion of Allen's Theory

The interval algebra as formulated by Allen has enjoyed great favor in natural language processing and AI. In terms of expressiveness, however, Galton (Chapter 13) argues that it is inadequate for representing continuous change. In continuous change, objects occupy locations instantaneously. Galton shows that 'x is at rest at L' and 'x is at L' are indistinguishable in Allen's system, arguing that what one needs is a concept of a property being true at an instant, without requiring that it be true at any interval containing or bounding that instant. Galton attempts to revise Allen's theory based on a combined instant–interval scheme, where instants fall within or limit intervals. A second criticism can be made in terms of computational tractability of interval algebras.

Vilain and Kautz (1986) show that consistency and closure computations in the interval algebra are NP-hard. However, they also show that part of the interval algebra can be converted to a point algebra where the computation is tractable (for an application of such an approach, see Pustejovsky et al.'s paper on TimeML in Part IV).

6 TEMPORAL REASONING AND THE FRAME PROBLEM

There are a number of classic problems that relate to temporal reasoning in environments where attributes and individuals change as a result of actions. Chief among these are the following:

1. Frame problem: accounting for those properties of a state that are not changed by performing a particular action.
2. Ramification problem: the explicit effects (direct or indirect) of performing an action.
3. Qualification problem: the conditions under which a particular action is applicable in the first place.

These problems are often collectively grouped under the first term, the **frame problem**. To illustrate what is at play here, consider the *give*-example from above. The action of giving makes explicit only who possesses the book at a certain time. Nothing else about the state of the world is mentioned by this specific action. The frame problem is that of determining what properties (fluents) are not impacted by an action. For example, all things being equal, giving a book to someone will not change the color or weight of the book. All that has basically changed is who has possession of the book. For all other fluent properties in the situation that we are modeling given the occurrence of an action, there must be a logical device to allow them to persist. McCarthy and Hayes (1969) introduced the notion of **inertia** into the Situation Calculus to account for these cases. A frame axiom is a rule associated with a particular action or class of actions that does just this. It keeps the color of the book the same, and so on (Shanahan 1997). Such persistence axioms reflect our knowledge of the way that the world is changed by actions, and how it is updated accordingly. Because the number of frame axioms grows proportionally to the product of possible actions over fluents in a domain, theories of nonmonotonic reasoning have been developed to capture the appropriateness of which properties should change, e.g. McCarthy's theory of circumscription, (1986) Reiter's default logic (1980), and Asher and Morreau's nonmonotonic reasoning (1991).

The ramification problem examines the related problem of computing those fluents that are impacted by the actions being performed. Rules defining these changes are called **effect axioms**; the computational issues with ramifications involve the appropriate pruning of inferences one can draw from an action. Schubert (1999) has approached the problem as one of finding an adequate explanation of change.

7 CONCLUSION

Given this account of temporal reasoning, we can see that the approaches we have discussed (with the exception of the situation calculus) offer expressive frameworks that can be applied to temporal reasoning in natural language. Both the McDermott and

Allen contributions explicitly address problems of temporal inference in natural language. The approaches explore reasoning with states, events, and point and interval representations of time.

Clearly, mapping from natural language representations to such frameworks requires an appropriate representation and annotation of events and the relations between them. As will be seen in Part III, most of the events in discourse have no explicit temporal anchoring and no explicit orderings relative to each other. Therefore, rules must be developed that capture how the text or discourse structures event-orderings. Hobbs and Pustejovsky (Chapter 14) show how an annotation scheme can be linked to a formal theory of time intended for use in representing the temporal content of websites and the properties of web services (the DAML Time Ontology). This linking allows interpretations of documents in terms of the annotation scheme to be mapped to a temporal representation where formal queries can be posed to a temporal reasoning system. The more we can tie in annotation to temporal reasoning, the closer we will come to solving some of the basic understanding problems for reasoning in language. This would be a very important capability in automatic reasoning.

REFERENCES

Abadi, M. and Manna, Z. (1989). 'Temporal Logic', *Programming Journal of Symbolic Computation* 8: 277–95.

Androutsopoulos, I., Ritchie, G. D., and Thanisch, P. (1998). 'Time, tense and aspect in natural language database interfaces', *Natural Language Engineering* 4(3): 229–76.

Baudinet, M., Chomicki, J., and Wolper, P. (1993). 'Temporal Deductive Database', in A. Tansel et al. (eds.), Temporal Databases: Theory, Design and Implementation. Redwood City, CA: Benjamin-Cummings, 294–320.

Belleghem, K. van, Denecker, M., and De Schreye, D. (1995). 'Combining Situation Calculus and Event Calculus', in Leon Stirling (ed.), *Proceedings of the 1995 International Conference on Logic Programming*, Cambridge, MA: MIT Press, 83–97.

Bentham, J. van (1983). *The Logic of Time*. Dordrecht: Reidel.

Carnap, R. (1947). *Meaning and Necessity: A Study in Semantics and Modal Logic*. Chicago, IL: University of Chicago Press.

Dean, T. L. and McDermott, D. (1987). 'Temporal data base management', *Artificial Intelligence* 32: 1–55.

Dowty, D. (1979). *Word Meaning and Montague Grammar*. Bouton: D. Reidel.

Gabbay, D. M., Hogger, C.J., and Robinson, J. A. (eds.) (1995). *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 4: Epistemic and Temporal Reasoning*. Oxford: Oxford Science Publications.

Galton, A. P. (1984). *The Logic of Aspect: An Axiomatic Approach*. Oxford: Clarendon Press.

Hanks, H. and McDermott, D. (1986). 'Default reasoning, nonmonotonic logics, and the frame problem', in *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia, PA. Menlo Park, CA: AAAI Press, 328–33.

Hayes, P. (1985). 'Naive Physics 1: Ontology for Liquids', in J. Hobbs and R. Moore (eds.), *Formal Theories of the Commonsense World*. Norwood, N.J: Ablex Publications.

Kamp, H. W. (1968). 'Tense Logic and the Theory of Linear Order'. Ph.D. dissertation, UCLA, Los Angeles, California.

Kroger, F. (1987). *Temporal Logic of Programs*. Berlin: Springer-Verlag.

Lamport, L. (1980). ' "Sometime" is sometimes "not never": on the temporal logic of programs;' *Proceedings of the 7th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (POPL). New York: ACM Press, 174–85.

Lamport, L. (1982).'TIMESETS: A New Method for Temporal Reasoning About Programs', in D. Kozen (ed.), *Proceedings of the workshop on Logics of Programs*, Lecture notes in Computer Science, 131. Berlin: Springer-Verlag, 177–96.

Kowalski, R. A. and Sadri, F. (1994). 'The Situation Calculus and Event Calculus compared', in M. Bruynooghe, (ed.), *Logic Programming: Proceedings of the 1994 International Symposium*. Cambridge, MA: MIT Press, 539–53.

Leith, M. and Cunningham, J. (2001). 'Aspect and Interval Tense Logic', *Linguistics and Philosophy* 24: 331–81.

Manna, Z. and Pnueli, A. (1983). 'Verification of Concurrent Programs: A Temporal Proof System', Department of Computer Science, Stanford University, Report No. STAN-CS-83-967.

—— (1992). 'Temporal Logic', in Z. Manna and A. Puneli, *The Temporal Logic of Reactive and Concurrent Systems*. Berlin: Springer-Verlag.

McCarthy, J. (1986). 'Applications of Circumscription to Formalizing Common Sense Knowledge', *Artificial Intelligence* 26(3): 89–116.

—— and Hayes, P. (1969). 'Some philosophical problems from the standpoint of artificial intelligence', in B. Meltzer and D. Michie, (eds.), *Machine Intelligence*, vol. 4. Edinburgh: Edinburgh University Press.

Moszkowski, B. (1986). *Executing Temporal Logic Programs*. Cambridge: Cambridge University Press.

Pnueli, A. (1977). 'The Temporal Logic of Programs', *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 46–57.

—— (1979). 'The Temporal Semantics of Concurrent Programs', in *Proceedings of the International Symposium on Semantics of Concurrent Computation*. Berlin: Springer-Verlag.

Reiter, R. (1980). 'A logic for default reasoning', *Artificial Intelligence*, 13 (1,2): 81–132.

—— (2001). *Knowledge in Action : Logical Foundations for Specifying and Implementing Dynamical Systems*. Cambridge, MA: MIT Press.

Richards, B., Bethke, I., Does, J. van der and Oberlander, J. (1989). *Temporal Representation and Inference*. London: Academic Press.

Schubert, L. (1999). 'Explanation Closure, Action Closure, and the Sandewall Test Suite for Reasoning about Change', in H. Levesque and F. Pirri (eds.), *Logical Foundations for Cognitive Agents*. Berlin: Springer-Verlag.

Shanahan, M. P. (1997). *Solving the Frame Problem*. Cambridge, MA: MIT Press.

Shanahan, M. P. 1999a. 'The Event Calculus Explained', in M. J. Wooldridge and M. Veloso (eds.), *Artificial Intelligence Today: Recent Trends and Developments*. Springer Lecture Notes in Artificial Intelligence no. 1600. Berlin: Springer-Verlag, 409–30.

Shanahan, M. P. (1999b). 'The Ramification Problem in the Event Calculus', C. S. Mellish (ed.), *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden. Morgan Ranfmann 140–6.

Shoham, Y. (1989). 'Time for action: On the relation between time, knowledge and action', in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, Detroit, MI. Morgan Ranfmann: 954–9.

Vilain, M. and Kautz, H. (1986). 'Constraint Propagation Algorithms for Temporal Reasoning', in *Proceedings of the Fifth National Conference on Artificial Intelligence* (AAAI-86), Menlo Park, CA: AAAI Press, 377–82.