ISSN 0105-8517

Time Representation & Use in Expert Systems

Nils Klarlund

DAIMI PB - 211
April 1986

DATALOGISK AFDELING
Bygning 540 - Ny Munkegade - 8000 Aarhus C
tlf. (06) 12 83 55, telex 64767 aausci dk
Matematisk Institut   Aarhus Universitet

PB - 211

N. Klarlund: Time Rep. & Use in Expert Systems

TRYK: RECAU (06) 12 83 55

Table of Contents

0 Introduction

The object of this report is to survey some of the uses and representations of time as it has been studied within the A.I. (artificial intelligence) community, especially within the expert systems subarea. In most applications the very concept of time is difficult to grasp. Therefore, we will mention some of the philosophical ideas that have had importance in the A.I. literature.

One of our main theses is that there exists many levels of uses of time. In this report a rather broad range of papers are mentioned. In the conclusion we will define degrees of "understanding" to distinguish different knowledge representations with respect to the "depth" of their use of time.

Another viewpoint is that representations of knowledge should primarily be discussed using a fixed formalism, preferably based on predicate logic.

We assume throughout this report that the reader has an elementary knowledge of predicate logic.

1 Expert Systems & Knowledge Engineering

We will shortly discuss what expert systems and knowledge engineering are in this chapter. As a result of the studies behind this report the author will here formulate some critical opinions. They will more or less be justified in chapter 3, where some articles and ideas are reviewed.

Expert systems

The concept "expert systems" has become yet another buzz word widely used by commercial and military companies and many researchers. An expert system is often conceived as a program embodying knowledge of a particular application area, called the knowledge domain, and having some inference machinery which makes it possible to use the knowledge for problem solving.

In some sense, "expert system" is a much more decent concept to market than "artifical intelligence", as it promises less. However, it should be emphasized that it is not possible to build systems acting as true experts. This is simply due to the seemingly insurmountable hindrance called "common knowledge".

We can only for some limited universes of discourse automatize the use and storage of knowledge. Hence, the correct concept to employ should perhaps be "knowledge databases". Nevertheless, we will stick to the term "expert system".

Expert systems are complex programs which often have required tens of man years to develop. They appear as results of efforts to identify and formalize knowledge of narrow domains normally only mastered by human experts. As opposed to usual complex programs as found for business or scientific applications, expert systems primarily manipulate symbolic data. Although inputs to expert systems often will be numerical (e.g. representing physical measurements) their processing wil result in their transformation to symbolic data corresponding to qualitative aspects as perceived by humans. In contrast to relative simple algorithms of numerical analysis, operations research, signal theory etc., algorithms used in expert systems are complicated reflecting an expert's way of reasoning about domain knowledge.

The field of expert systems faces the same problem as artificial intelligence in general, namely the total lack of a general understanding of human reasoning. Many such theories have been proposed (e.g. [Sowa 1984]), but no one is generally accepted.

Heuristics

Some attempts have been made to introduce human-like paradigms into A.I. Domain specific "rules of thumb" for problem solving are called *heuristics*. The use of heuristics is often thought of as a distinctive mark of expert systems making them less "algorithmic" or "machine-like". This must be considered to be just a feeling, as programs resulting from the use of heuristics are just as algorithmic as any other programs. This stems from results of the mathematical theory of computability: all powerful programming formalisms are equally powerful.

Unfortunately, there does not seem to exist a firm basis for the uses of heuristics. The very nature of heuristics forbids it. They are applied in an *ad hoc* fashion which makes the overall semantics of expert systems obscure. Nevertheless, these techniques e.g. as applied in the DENDRAL (mass spectrometry interpretation) and HEARSAY-II (speech recognition) projects have been quite succesful.

Knowledge engineering

A discipline called knowledge engineering has recently emerged. Companies like *Teknowledge*, *Machine Intelligence Corporation* (both in Palo Alto, CA) and *Thinking Machine Corporation* , Cambridge, MA sell expert system know-how to clients who want to partially or fully automatize functions normally carried out by experts.

Such systems can be partioned in at least three groups according to their function (inspired by [Stefik 82]):

- Interpretation (analysis of data, diagnosis, monitoring)
- Prediction (forecasting the course of the future from information about the past)
- Planning (how to achieve goals, design)

Their construction involves at least a person who is an expert within the knowledge domain, and a person who is an expert on expert systems. The latter is sometimes given the prestigious title "knowledge engineer". The tools of such engineers are primarily their knowledge about AI and system building. Maybe they will use ready-made software components and tools as databases, parsers, inference machines etc.

Time and expert systems

The ubiquity of time is easily seen from the above partition. Interpretation is often done of time-varying information as when monitoring patients at hospitals. Time is inherent in prediction; and planning can be to achieve a goal by initiating physical events. For some applications area, however, there is no notion of time at all. It is possible to associate time to every program as being a computing resource. In this report we will, however, only deal with temporal issues of the knowledge domains.

A system like DENDRAL (mass spectrometry interpretation) does not need to know anything about time as its domain is static.

Medical systems

One should think that we cannot avoid representing time in medical systems. As an example, it must have importance for how long a period a patient has had pains, if they were periodical etc. It seems that most such systems do not have a well-defined time concept. An exception is the ALVEN system [Tsotsos 85] and the Ventilator Manager which will be shortly described in chapter 3.

Prediction

Interesting questions of time arise for expert systems which are to predict the course of the future. There may exist several possible futures, some of them being dependent on future events. An adequate time representation should encompass such non-determinism.

Planning

Planning involves time as a plan is a pattern of actions to be performed. A plan is constructed with a particular goal in mind. Actions to be performed to obtain the goal may require subgoals to be carried out. Proper attention must be taken to temporal relationships between actions (as "let the water boil before putting the eggs into the water"). Several proposals for formalisms that deal with event description and planning have been proposed. We will mention some of these proposals in chapter 3.

2 Time

This chapter will discuss the difficult concept of time. We will mention some of the philosophical work and outline how representations adequate for expert systems can be derived. Most of the questions raised here are discussed very clearly in [Prior 67].

The nature of time

What the physical time really is will be of no concern to us as long as the knowledge domain is not the theory of relativity or something similar. We are more interested in modeling our (naïve) perception of time. Such a modeling is best performed with mathematical means, at least if we want to use the models computationally. There are at least two important issues, both of epistemological nature: 1) we look for a model of how we perceive time (which is not the same as what time really is), and 2) we must relate such a model to our use of time in natural language (English). One of the achievements of modern philosophers has been to formulate temporal logics which are used both for the modeling of our time conception, and for describing the underlying temporal structure of our language. The problems raised in 1) and 2) have also been treated by researchers within cognitive psychology and linguistics.

Models of human perception of time

The english philosopher J.E. McTaggart is quoted in [Prior 67] for having stated: "There could be no time if nothing changed". A philosopher of logic describes the world at a given instant by propositions. Time is necessary only to explain how the values of such propositions change.

The same McTaggart introduced the notion of A-series and B-series. A-series is a conception of time built around the operators *past*, *present*, *future* corresponding to "*in the past...*", "*now...*", "*in the future...*". These operators are relative to "now" and related to tenses in natural language. In a B-series model dyadic *b* operators such as *before* are used to compare points or intervals of time. Propositions are absolute, not dependent on any "now". We will later meet this dichotomy between A and B conceptions when looking at actual expert systems. Other questions about our perception of time are:

- do we construe time as consisting of intervals (segments, durations) or as consisting of points (or both) ?
- is time infinite or finite ?

- is time "dense" (i.e. is there between any two points in time always another point) ?

These questions can all be formulated as constraints on the kind on mathematical structures that temporal logics are interpreted over.

Language and time

Temporal references in our language exist very often in the form of *tenses*. According to Webster's New Collegiate Dictionary, a tense is a "distinctive form in a verb for the expression of distinctions as to time". For a verb like "*be*" some tenses are "*is*", "*it will be*", "*it has been*", "*it was*", "*it has been that it will be*", "*it was going to be*", etc. Such tenses can translated to temporal operators in an A-type temporal logic.

<u>Formalization of tenses</u>

As an example, if we wish to express the tense "*it has been*" with an operator **P**, we can give the semantics of $\mathbf{P}p$ "*it has been that* p" for a proposition p as the following. At an instant t, $\mathbf{P}p$ is true if and only if there is an instant t' so that t' is before t and p is true at t'. Note the use of the word "before" in the definition. This suggests that B-series are more fundamental than A-series as A-expressions can be reduced to B-expressions. However, many philosophers believe that B-logic can be reduced to A-logic, i.e. A-logic is more fundamental.

Similar to the definition of **P**, we define the operator **F** to denote future. Hence, $\mathbf{F}p$ is true if p will be true some time in the future.

<u>Uses of tenses</u>

Many philosophical articles about time treat substantial philosophical problems using only a few operators à la **P** and **F**. For example, we can express that future becomes past with the help of:

$$\forall p : \mathbf{F}p \supset \mathbf{F}(\mathbf{P}p)$$

This formula is read that for each propositions p that will become true, it will also (in the future) be the case that p has been true.

Axioms of tenses can also be used to describe constraints on the structures over which the logic is interpreted. As an example,

$$\forall p : \mathbf{F}p \supset \mathbf{FF}p$$

expresses that time is dense: if it will be the case that *p*, then it will also be the case that it will be the case that *p*. For details, see [Prior 67].

FF is an example of an iterated tense. Other iterated tenses are **PP** (*"it has been that it has been"*), **PF** (*"it has been that it will be"*), **FPP** (*"it will be that it has been that it has been"*) etc.

An important feature of tenses is that the semantics is context dependent. By knowing that *"the sun is shining today"* is false, we cannot deduce whether **P**"*the sun is shining today*" is true or false, e.g. whether it is true or false that at some day in the past the sun was shining. Hence, the semantics of **P***p* is not merely a function of the simple boolean semantics of *p* . However, so-called "intensional logic" permits to achieve compositionality, see e.g. [Hobbs & Rosenschein 78].

<u>Reference points</u>

A tense may be seen as an operator which when applied to a proposition at a certain time or reference point, gives a new reference point for the interpretation of the proposition. Hence, iterated tenses may define several reference points ordered according to the meaning of the tenses. The question of how many tenses are necessary for the modeling of natural language has been treated numerous times in the philosophical literature.

<u>"While...", "...until..."</u>

Besides the tenses that move reference points, there are other important constructs in natural language which sometimes are regarded as tenses. Adverbial phrases as *"while..."*, *"...until..."*, *"during..."* have all been formalized and also been implemented in computer systems as we will see later.

<u>"Beginning of..."</u>

It is tempting to suggest that the underlying concept of the constructs above is an *interval*. Such a conception would make it possible to easily handle other temporal constructs as *"the beginning of ..."* , *"the end of..."* etc. Note that whereas we until now have not explicitly mentioned time or time points in the syntax of our formalization attempts (**P**,**F** etc.), it will be difficult to translate the sentence containing *"the beginning of..."* to a formal language without introducing an syntactic notion of time.

The question of what "beginning" means was already studied by Aristotle.

Fuzzy Concepts

Another type of temporal constructs poses great difficulties and is nearly impossible to formalize as the meaning often requires an in-depth understanding of the context. Consider the sentences "*Some time ago I ate to dinner*" and "*The war ended some time ago*". In the first sentence "*some time ago*" probably just means a few hours. In the second sentence "*some time ago*" denotes a time span of years.

Another kind of uncertainty can be found in statements like "*He came at about two o'clock*". Is it 2:00 ± 1 second or 2:00 ± 1 hour ? (Notwithstanding the problems of determining whether it is A.M. or P.M., what "*he*" denotes etc.) We will see a system which tries to cope with these issues. In contrast to the other issues mentioned in this chapter, the problem of fuzziness of time has not been thouroughly treated by philosophers.

Ampliation

The problem of ampliation arises when things are not the same at the reference time as at the time of description. The sentence "*the president was a studious pupil*" cannot be translated to "in the past there exists a x such that *president*(*x*) and *studious-pupil*(*x*)". In the past the president was not president. The *y* such that *president*(*y*) was true in the past, is different from the *x* such that *president* (*x*) is true now. In order to translate "*the president was a studious pupil*" to logic, the value satisfying the predicate *president* must be transfered or "ampliated" to the past. The term "ampliation" was introduced by scholastic philosophers.

Multiple futures

We usually aggree that there is only one past which is a kind of line or succesion of events or time points or intervals. When speaking about the future, there is normally not *one* future but several possible futures. This has been formalized in philosophical logic giving rise to operators for denoting "*it is possible*" and "*it is neccessary*".

As an example, "*it will neccessary be that the sun shines*" can be interpreted in a branching time logic as: whatever branch the world evolves along from now, there is some instant where "*the sun shines*" is true. Such logics are called branching time logics. In [Øhrstrøm 84] some of these logics are compared and related to philosophical conceptions of time.

3 Uses of Time concepts

In this chapter we will illustrate the variety of time concepts of relevance to expert systems and A.I. Examples spanning from databases to philosophy will indicate the substantiality of the problems of time representation and use.

The word representation can be construed in two ways. Either it designates the concrete memory organization in a computer, or it refers to the abstract structuring of entities denoting facts. We will concentrate on the second aspect which we believe must be investigated prior to computer storage techniques.

Before presenting the examples, we will argue the case for the use of logic within A.I. Our choice of examples reflects our propensity to rigourous approaches to "knowledge engineering".

Logic as a knowledge representation language

Historically, the language of predicate logic was developed in the search for an abstract formalism of mathematical or stringent philosophical reasoning (Boole, Frege, etc.). Today the predicate logic and its refinements in the form of exotic logics (modal, intentional, temporal, epistemic, etc.) are still the most important conceptual tools for philosophers who try to formalize ideas about language and understanding. A good overview of applications of logic within A.I. can be found in the little book [Turner 84].

A number of other kinds of formalisms has been introduced within the A.I. field as e.g. frames [Minsky 75] and conceptual dependencies [Schank 75]. It seems that these formalisms have been introduced primarily for computational reasons, and that their best use is for memory organization. (In the book "Introduction to Artificial Intelligence" p. 393-415 [Charniak & McDermott 85], it is discussed how such memory organization may speed up inference procedures.)

Hayes [Hayes 1985] has argued that most representation schemes are equivalent with first-order logic when dealing with representational power (not computational efficiency). He notes that many systems have invisible assumptions about their domain which are not documented. Such assumptions show up when the formalism is translated to first-order logic. Further, he remarks that thinking about control issues (how to infer consequences of stored information) is premature, until the sorts of desired inferences become clear (admitting that the study of meta-information about the inferential process brings up its own representational problems). As we will see, control issues are sometimes embedded as a part of the representation.

First-order logic is normally thought of as stating properties of simple domains where propositions, predicates and functions get a fixed interpretation in each structure (model). General reasoning about predicates and functions is not possible as quantification over those constructions is not permitted. However, as Hayes points out, a logic is made a higher-order logic only when quantifiers are allowed to range over *all* predicates.

On the other hand, it has been shown that higher-order logic can be collapsed into first-order logic if self-reference is a little constrained [Perlis 85].

In the following, a number of examples taken from the A.I. literature will be discussed some of them based on logic. We will discuss issues such a the notion of time invoked, abstract and concrete representation of the problem domain, and inference mechanisms.

Registration and interpretation of events

We present two "early" papers from the seventies dealing with problems of storing and retrieving information about time and events. They provide the first steps towards more general representations. As opposed to some of the later papers we will present, the ideas in these two early papers have been carried out practically in the form of running computer programs.

Bruce' model and the "Chronos" program

In the paper "A model for Temporal References and Its Application in a Question Answering Program" [Bruce 72] a formalism for the description of aspects of the temporal structure of natural language is presented, and it is shown how it can be implemented in a computer program. It is claimed that the formalism can "...understand most of the temporal meaning in a sentence". This has to taken with a grain of salt as we have already discussed problems as Chronos does not at all manage such as ampliation and multiple futures; not even predicates are implemented. Also there is no general way of treating imprecise time. It is interesting that Chronos was build on the basis of philosophical or linguistic observations (Prior, Reichenbach, Jespersen are cited). The logic developed employs temporal segments or intervals, and permits tenses to be expressed.

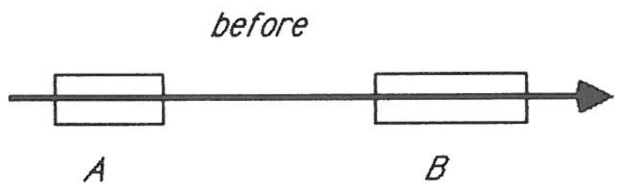
In Bruce's model time consists of *time-points* which are partially ordered. Hence, time is not assumed to be linear, but the branching of time does not seem to have implied special concepts to be implemented as e.g. modalities. Segments are linear sets of time-points containing all points between the first and last point. Hence, segments could also be called intervals. Bruce also assumes a set of *events*. Further, he requires that there is precisely one time segment corresponding to each event. This time segment is given by a function *duration*. Dyadic segment operators as *before*, *during*, *same-time*, *overlaps*, *after*, *contains* etc. are easily defined with reference to time-points. Some examples:

(BE) $before(A, B)$ iff for all time-points $a, b: a \in A \wedge b \in B \supset a < b$

(ST) $same\text{-}time(A, B)$ iff $A = B$

(DU) $during(A, B)$ iff $A \subseteq B \wedge A \neq B$

(BE) reads: the segment A is before the segment B if and only if every time point in A is before any time point in B:



It seems that the author has not forbidden empty segments. Hence, the relation *before* is not transitive (!). A relation *after* is defined symmetrically to *before*. Bruce defines a *tense* as an n-ary relation of time segments defined by a relational statement of the form

$$\bigwedge_{i=1}^{n-1} R_i(S_i, S_{i+1})$$

where each S_i is a time segment and each R_i is a binary ordering relation (such as *before*) on the time segments S_i and S_{i+1}. It is assumed that all the intermediate sets $S_2, \ldots S_{n-1}$ be singleton sets. A tuple $(S_1, \ldots, S_n)$ satisfying a tense is called an element of tense. For such an element S_1 is called the time of reference and S_n the time of event.

Using tenses

The phrase "*he had gone*" is according to Bruce represented with the help of the tense

$$AA(S_1, S_2, S_3) \equiv after(S_1, S_2) \wedge after(S_2, S_3)$$

Here S_1 is the time of speech, S_2 is a reference point, and S_3 is the time of "his going". It is seen that *after* corresponds to the **P** operator we have already met in chapter 1. As another example he gives "*I watched TV from 2 to 4*". He proposes the tense

$$AS(S_1, S_2, S_3) \equiv after(S_1, S_2) \wedge same\text{-}time(S_2, S_3)$$

and says that S_2 is the duration bounded by 2 and 4, and S_3 is the event of watching TV. That S_2 lasts from 2 to 4 seems to be in contradiction with the earlier stated requirement that S_2 be a singleton set. It seems unclear why he does not simply use the tense $after(S_1, S_2)$ where S_2 is the event of watching TV.

Bruce is vague about how to model concepts as beginning, ending, while etc. It is, however, clear that if segments are adequate for representation of the times where events occur, then such concepts can be easily defined. As an example, "while e_1 occurred, e_2 occurred" is translated to

$$during(S_1, S_2)$$

where S_i, $i = 1,2$ is the time segment of the event e_i.

As an illustrating example of the occasional (or persistent?) naïveness of A.I. research, we will mention that Bruce "shows" that modalities generally can be represented by conjoining a tense form with a form for the additional meaning. As an example, he translates "*I should go*" to

$$B(s, e) \wedge \text{obligatory(going)}$$

Unfortunately, this is no further explained. Such a discussion about modalities is futile as long as the meaning of proposed constructs is not thoroughly discussed. The pitfall is to convert sentences in English to some beautifully looking formal phrases or tangled graphs believing that meaning has been captured formally.

Chronos

There is no description of what requirements the program "Chronos" fulfill. Functionally it is said to contain a parser/generator, a database of facts about events, a theorem prover and a control component. No details are given about the inference procedures. We will here reproduce one of the dialogues that can be performed with the program:

User:	(BILL ARRIVES)
Chronos:	(INFORMATION ACCEPTED)
User:	(WHEN DID BILL ARRIVE *)
Chronos:	EIGHT
User:	(MARY HAS ARRIVED)
Chronos:	(INFORMATION ACCEPTED)
User:	(WHEN DID MARY ARRIVE *)
Chronos:	(IN THE PAST)
User:	(JOE HAD ARRIVED AT THREE)
Chronos:	(INFORMATION ACCEPTED)
User:	(WHEN DID JOE ARRIVE *)
Chronos:	(IN THE PAST)
User:	(DID JOE ARRIVE BEFORE THREE *)
Chronos:	YES
User:	(ALICE ARRIVED AT SIX)
Chronos:	(INFORMATION ACCEPTED)
User:	(WHEN DID ALICE ARRIVE *)
Chronos:	(IN THE PAST FROM SIX TO SIX)

Conclusion

This early paper (1972) is interesting for its discussion of some linguistic ideas, although neither the adequacy of the chosen representation, nor the requirements of the proposed systems are sufficiently elucidated.

The time specialist of Kahn and Gorry

In [Kahn & Gorry 77] a program called a time specialist is presented. It is claimed to be a program knowledgable about time in general. We quote from the introduction of the paper:

> "To deal effectively with everyday life, a person must understand how events are related to one another in time and how plausible deductions can be drawn from the temporal characteristics of these occurrences. The paramount importance that such an understanding plays in human problem solving argues forcefully that no computer program can be

> "intelligent", if it lacks a sophisticated temporal sense. In this paper, we will consider the problems of equipping a computer with such an understanding."

As with the prior paper the contents only very partially satisfies the promised goals. The "temporal characteristics" mentioned above are answers of questions like "How many years after event *A* did event *B* take place?" (where *A* and *B* are uniquely identified events). As this is the level of reasoning, the word understanding is used in the same sense as a database containing tax data (income and fortune of citizens etc.) "understands" the fiscal system.

The authors present what they call a rudimentary time specialist program which understands temporal specifications. A temporal specification is conceived as a statement that partially specifies the time of one or more events. Some examples fetched from the paper are:

1) Three weeks ago John had a cold.
2) John was born June 6, 1966.
3) A few months from now, John will finish his thesis.
4) Two or three years after graduating, John went back to school.

Each temporal specification consists of a relation between two events, one of which is called the "event" corresponding to the main event in the specification. The other is called the "reference event". In the specification 1), the reference event is "now", and in 2) the reference event is the "zero" point of the calendar.

The authors have not dealt with any of the linguistic issues of time. Input to the system is given in form of sentences like "(time-of(beginnning-of life)(date (1952 1 25) (fuzz nil)))" meaning "I was born January 25, 1952". The authors present the following dialogue which they have translated into English from the machine representation (!):

User:	I was born January 25, 1952. When I was a few weeks old I had an operation. I began grade school when I was 5. I graduated grade school in June 1965. I went to England two years later. I stayed in England for 8 or 9 weeks.
User:	How old was I when I left England?
System:	15.5 years give or take 1.6 months.
User:	Explain your answer.
System:	The time of the end of the trip to England is August 17, 1967 give or take 1.6 months

and the time of birth is exactly, Friday, January 25, 1952.

User: How long ago did I go to England?

System: Between 8 and 8.3 years ago.

User: What happened before I was 10?

System: The beginning of grade school, the operation, and the birth.

User: I was born May 5, 1955.

System: I thought the time of birth is January 25, 1952 not May 5, 1955, are you referring to the same birth?

User: Yes.

System: Is May 5, 1955 correct?

User: Yes.

System: Earlier I was asked how old the person was at end of the trip to England, and I replied 15.5 years give or take 1.6 months. However, my reasoning was based on the fact that the birth was January 25, 1952, which you now say is incorrect. We could either forget it or I could answer the question again. Shall I re-run it?

User: Yes.

System: The answer is now "12.3 years give or take 1.6 months".

Although the above dialogue seems convincing, this must be attributed more to the presentation than the inferences obtained. The calculations performed are really rather trivial.

The most interesting concept introduced is that of *fuzzy time*. The inexactness of temporal specification can be represented with the help of *fuzzy amounts* like ABOUT, NEARLY, A-BIT-MORE-THAN,... combined with A-HALF, ONE,... and DAY(S), WEEK(S), MONTH(S), YEAR(S) (e.g. NEARLY A-HALF YEAR). How such fuzzy amounts are added are not explained. An event such as 4) "A few months from now, John will finish his thesis" is represented as

```
(TIME-OF (END-OF (JOHN'S WORKING-ON THESIS))
         (AFTER   (ALL-OF TODAY)
                  (FUZZY-AMOUNT (NIL A-FEW MONTHS))))
```

All events are contained in a database. Events in the database that happened during a particular time interval are represented with the help of a *date line*, a list of events sorted in chronological order. Before and after relationships between events are kept in separate chains called *before/after chains*. For reasons of efficiency, facts that are often referred to may be organized by *special reference events*.

When new information about events is given to the system, deduced consequences have to be inserted (in date lines, before/after chains or as special reference events). All consequences do not need to be calculated upon data entry. The program records the methods which have been applied to a fact, but it is possible later to apply another method.

The program performs consistency checks, and as it was seen at the sample dialogue, it is possible to change already given information. The program have recorded what prior use of that information has found place, so that earlier answers using the information can be recalculated.

Conclusion

The representation of events and the deduction processes are not exposed clearly and only on an informal level. Therefore, we will leave the discussion of this article here.

Medical systems

Several expert systems have been developed within the area of medicine. Unless the area is a very narrow one of static data interpretation (e.g. chromosone analysis by caryo diagrams), time and change will be essential concepts for consultation systems. Decision making processes are in their nature temporal. Both the time of occurrences of events and the time of taking events into consideration are important for the modeling of interpretation, diagnosis and recommendations.

As knowledge in medical expert systems should be easy to understand and modify, time and change must be represented in an uniform manner preferable in a formal language resembling natural language. Hence, temporal logics could be a good starting point for the construction of medical systems.

We will rather briefly discuss three expert systems within medicine and see how they have coped with time and change.

The MYCIN system

(This paragraph is based on [Shortliffe & Fagan 82].) In the mid-1970s the MYCIN program was developed at Stanford University. It was designed to give assistance to physicians in the narrow domain of selection of antibiotics for patients with severe infections. It is claimed to have reached performances comparable to those of physicians. The handling of time and change is simple. MYCIN ignores information which changes with time. It is only possible to analyze a snapshot of the patient's pathological picture.

The ALVEN system

In the paper [Tsotsos 84] a more recent medical expert system with a very narrow domain of knowledge is presented. The purpose of ALVEN is that of evaluation of the dynamics of left ventricular tantalum marker implants from X-ray image sequences. It is, however, a bit unclear how the system is thought to function as a physician's tool. Tsotsos does not give any account of clinical tests of the system, and the only example he discusses required 30 minutes of VAX 750 CPU time. So it cannot be regarded as more than a prototype system.

As some medical knowledge is necessary to read the paper some elementary comments are appropriate here. The left ventricle (heart chamber) is the most important of the two chambers and the study of its dynamics (how the walls move) is of significance for the diagnosis of heart diseases such as sclerosis. It is possible in connection with a coronary bypass surgery to place tantalum markers on the outer surface of the left ventricle. Such markers show up on X-ray films of the heart indicating the contur of the ventricle.

We quote [Tsotsos 84]: "The goal is to analyse both pre-operative (without markers, using contrast media) and post-operative marker films ..., to evaluate the efficacy of surgery, locally and globally, quantitatively and qualitatively, over the recovery period (several months), and to evaluate the effects of drug interventions."

Although modern X-ray diagnostic systems using digital image processing equipment are available (two such systems exist in Denmark; one is a "Phillips DVI-V DVI-CV APU integrated system for cardiovascular digital subtraction angiography" in Gentofte which has been inspected by the author), there seems to be no commercially available computerized techniques for the evaluation of data, except for simple measurements such as ventricle volume. According to Tsotsos prior attempts to model left ventricular dynamics have been mostly mathematical. Mathematical models are not flexible so the goal of ALVEN is a system containing concepts which correspond as directly as possible to the understanding and terminology of the physician. Tsotsos says that there are two major aspects of the work he has undertaken:

- an understanding of visual motion
- reasoning about spatio-temporal relationships

By the first point he seems only to mean a general representation of simple objects moving in a series of pictures. The second item seems to cover a method for the description and identification of composite motion. The paper discusses both points even though the distinction is not kept clear. Only a short presentation of the ideas in the paper will follow as the paper is written in an esoteric language which is not very well explained. The kind of time coming into play is different from other works. Time is not studied as something underlying natural language, but rather as a parameter in a physical model. However, as the strategy is to introduce a formalism close to humans, some simple concepts in our language like "before" and "after" will appear in the model.

The most important feature of ALVEN is that five types of inference processes may work in parallel. These inferences are closely linked to the chosen knowledge representation. Tsostos uses a kind of frames called *classes* for representation with three kinds of links:

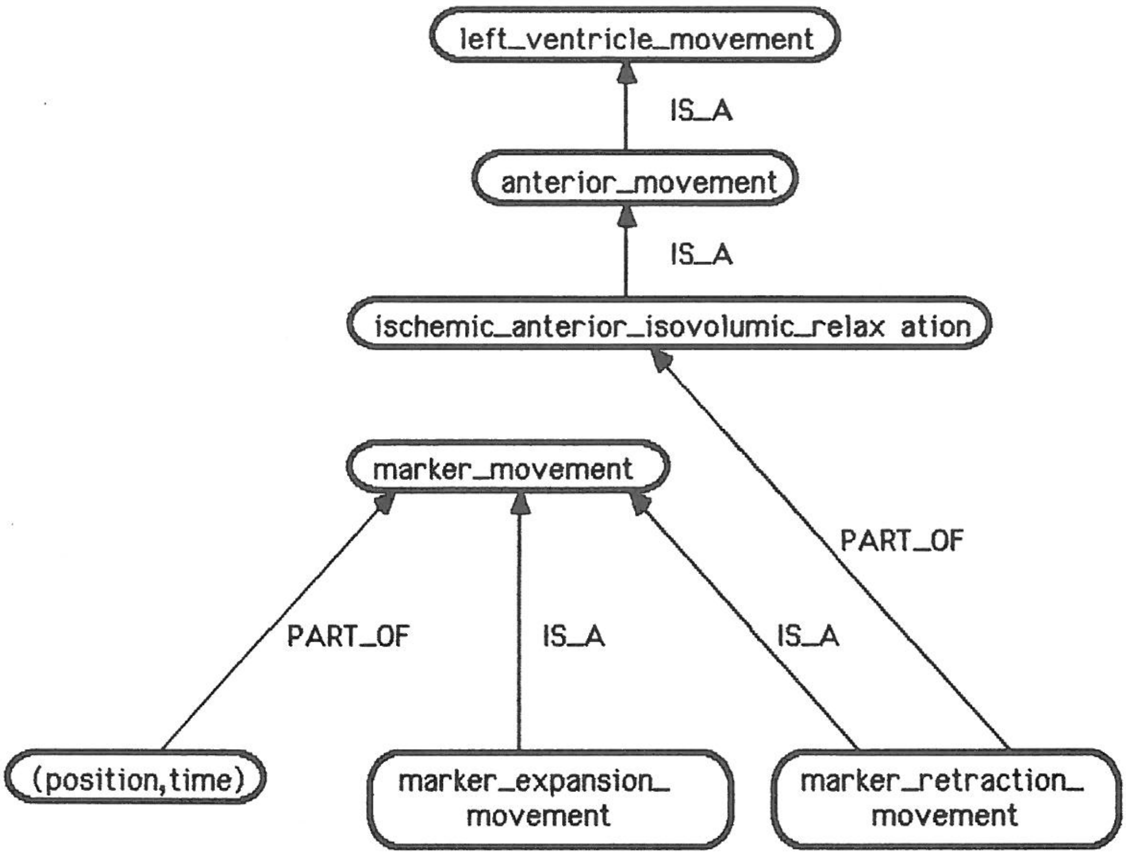
- PART_OF: A concept *A* is a PART_OF concept *B* if *B* consists of parts one of which is *A*. As an example we can say that the different phases of the movement of the heart are all PART_OF the heart movement cyclus. These links correspond to levels of resolution.

- IS_A: The concept *A* is a concept *B* if every instance of *A* also is an instance of *B*. As an example, an "inward wall movement" IS_A "wall movement". These links correspond to levels of specificity.

- SIMILARITY: This concept seems to constitute a kind of declarative control for comparison of similar classes which are not necessarily connected by other links.

<u>A constructed example</u>

We have tried to figure out an example of knowledge representation. (It may contain misunderstandings as it is built on a not very complete understanding obtained by reading the paper.) The following figure is a skecth of how knowledge about change could be represented:



The (position,time) concept is directly extractable from the actual image. A marker_movement consists of several such (position,time) tokens. We also see that a marker_expansion_movement and a marker_retraction_ movemement both are marker_movements with some additional constraints. A marker_retraction_movement is a part of a ischemic_ anterior_isovolumic_relaxation which is a special kind of an anterior_ movement which, in turn, is a special kind of a left_ventricle_movement. The use of such a knowledge representation scheme is roughly described in the following.

The information from the input is used to instantiate concepts of the leaves of the PART_OF hierarchy. Going up this hierachy we can, driven by the data, generate hypotheses. In the example, we may generate the hypothesis marker_movement. These hypotheses may be specialized or refined by downward traversal of the IS-A hierarchy. Such a new set of hypotheses have to be checked by other instances. Referring to the example, we could try to specialize the concept of marker_movement to marker_expansion_movement. This latter hypohesis has to be checked using the underlying hierarchy (not shown on the figure). If a new set of hypotheses is shown to be consistent, the further refinement will be attempted. Failure, on the other hand, will necessitate the selection of alternate hypotheses.

Tsostos uses a ranking of hypotheses explained by a full investigation of the PART_OF hierarchy. Such a ranking is used to delete some hypotheses while other are used for the prediction of new events (i.e. new positions of the markers).

Relaxation algorithm

The interpretation over time is, in addition, performed with a relaxation algorithm. A *relaxation algorithm* denotes a set of processes which iteratively changes the interpretation of each element to be in greater harmony with its neighbors. This harmony is expressed by hypothesis satisfication.

A hypothesis ranging over time depends on the spatial information (marker positions) as a function of time. With the help of relaxation processes local hypotheses are evaluated with respect to each other.

Tsostos have used dynamic weights of the links between local hypotheses. This means that the values of the links change as a function of time.

Conclusion

The work of Tsostos seems to be of an *ad hoc* nature. The most important issue of relevance to our subject is the ideas of how change can be described as a hierarchy of concepts. The notions of PART_OF is not directly found within temporal logic even though it seems intuitively very sound. The nearest we can come this notion is the "$\wedge$" operator, $p \equiv p_1 \wedge \ldots \wedge p_n$, meaning that the properties $p_1, \ldots, p_n$ are the parts of the property p. Similarly, we can approximate the IS_A notion with a "$\supset$" operator.

Tsostos uses five simultaneous and interacting inference processes, one of which is a relaxation algorithm, traversing the trees up and down, left and right. This makes the overall semantics of the interpretation difficult to understand.

The Ventilator Manager system

According to [Shortliffe & Fagan 82] VM is an extension of a physiologic monitoring system designed to perform the following specialized tasks in an intensive care unit:

- to detect possible errors in measurement
- to recognize untoward events in the patient/machine system and suggest corrective action

- to summarize the patient's physiologic status
- to suggest adjustments to therapy based on the patient's status
- to maintain a set of case-specific expectations and goals for future evalutation by the program

As opposed to the ALVEN system, VM interprets several different kinds of time-varying data such as heart rate, blood pressures, CO_2 in expired air etc.

Knowledge representation

As in MYCIN and many other expert systems knowledge is represented in VM by production rules. The general form of a rule is:

IF: Relations about one or more parameters hold
THEN: 1) Make a conclusion based on these facts
2) Make appropriate suggestions to clinicians
3) Create new expectations about the future values of parameters

The conclusion in 1) may trigger other inferences to be drawn or indicate actions to be performed. A rule definition also includes information about all of the therapeutic states in which it makes sense. A typical rule is shown below:

STATUS RULE: STABLE-HEMODYNAMICS
DEFINITION: Defines stable hemodynamics based on blood pressures and heart rates.
APPLIES: to patients on VOLUME, CMV, ASSIST, T-PIECE
COMMENT: look at mean arterial pressure for changes in blood pressure and systolic blood pressure for maximum pressures.

IF: HEART RATE is ACCEPTABLE
PULSE RATE does NOT CHANGE by 20 beats/minute in 15 minutes
MEAN ARTERIAL PRESSURE is ACCEPTABLE
MEAN ARTERIAL PRESSURE does NOT CHANGE by 15 torr in 15 minutes
SYSTOLIC BLOOD PRESSURE is ACCEPTABLE

THEN: The HEMODYNAMICS are STABLE

The VOLUME, CMV, ASSIST and T-PIECE are different kinds of ventilatory therapies. The attributes ACCEPTABLE depend on the clinical situation e.g. which stage in the intensive care therapy the patient has reached. The attributes CHANGE are presumably also dependent on the clinical situation. This indicates that whereas time may be considered isolated, the notion of change is intimately connected to the overall context.

Conclusion

The VM system appears to have been made in a more systematic way than the ALVEN system. The VM approach may be very useful to study when constructing expert systems that reason about change.

Planning

Only one paper will be presented in this section due to lack of space and time. Albeit short, we think this paper presents some very interesting ideas. Another good reference of planning with logic is [McDermott 85].

Process control using temporal logic

In the paper [Fusaoka et al. 83] a new way of using temporal logic is outlined. The approach is a rare example of a systematic way of handling reasoning *about* temporal relationships as being temporal relationships. This is strongly opposed to reasoning about instances of temporal relationsships as it is done in the event representation systems that we have discussed before. Although their paper is short, and contains no epistemological considerations, we believe it is an important example for illustrating the distinctions of levels of temporal understanding.

The domain of knowledge

The field of automatic control has been subject for a mathematical theory based on Laplace and Fourier transformations etc. Instead of a mathematical modeling the authors propose a logical modeling of control systems. If there are only a few states of the system under study, it is quite easy to represent them using temporal logic. Further, using temporal logic the dynamics of the system as well as the desired behavior can be expressed in terms of temporal characterizations of state sequences.

The purpose

The paper shows how such a formalization may lead to the automatic synthesis of control rules for the system. A control rule determines when and under which circumstances switches and processes have to be changed in order to obtain the desired behavior. The desired behavior may be a state to obtain, but it may also be requirements of stability, i.e. concepts which hing on temporality.

The logic employed

The authors use a standard version of temporal logic as found within computer science. That is, the logic is interpreted over linear, countable, right-infinite sequences of the form $\sigma = s_1, s_2, \ldots$. The temporal operators are:

$\Diamond P$	meaning P is true now or at some future instant
$\Box P$	meaning P is true now and at all future instants
$\circ P$	meaning P is true at the next instant
$P\mathbf{U}Q$	meaning P is true at all instants until Q becomes true

(Depending on the exact definition of $\mathbf{U}$, Q be may required to become true or not.)

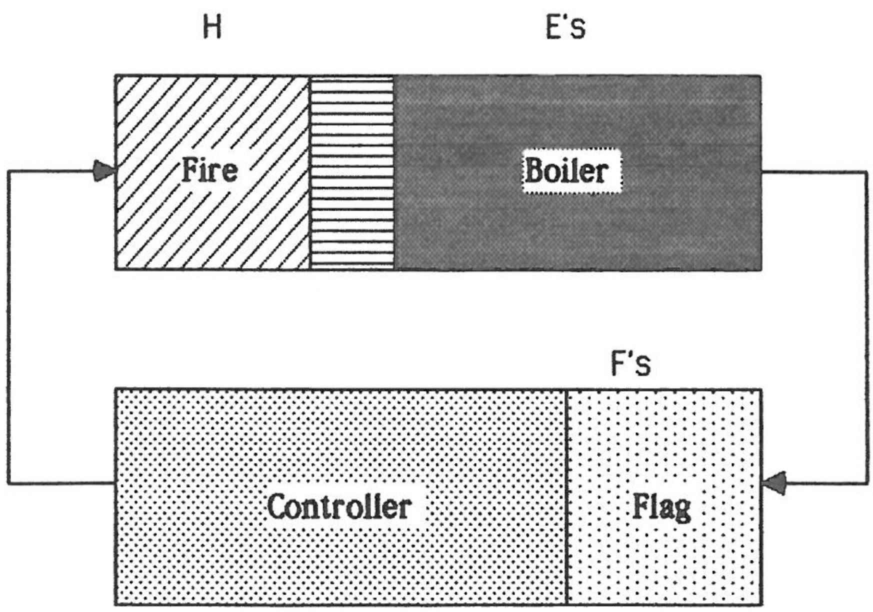
System description

A system consists of objects which are connected with each other to allow message passing. An object may have internal states called "flags". Each possible value of the state corresponds to a simple (atomic) proposition.

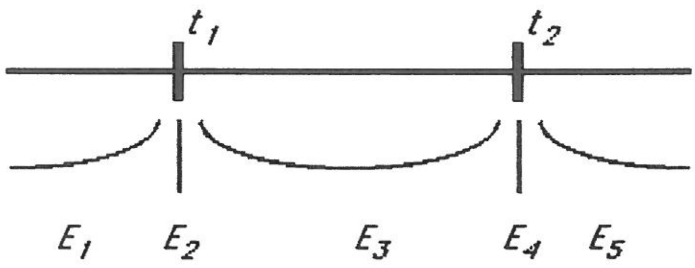
The sending of a message by an object is stated as temporal conditions relating flags in the sending object and the receiving object.

Example

The paper discusses the following very simple system consisting of a boiler and a temperature controller:



The fire may either be on or off represented with the simple proposition H. The temperature is a state of the boiler represented by the flags E_1, ... E_5. If the temperature is less than t_1 then E_1 is true etc. as illustrated by :



Then we have as axioms:

$$\Box(E_1 \vee \dots \vee E_5)$$
$$\Box(\neg(E_i \wedge E_j)) \quad \text{for each } i, j \ (i \neq j)$$

stating that it is always the case that exactly one of the variables E_1, ... , E_5 is true. The state F of the controller is represented with the flags F_0, ..., F_5. F_0 is true when there is no change of E, while F_1,..., F_5 is a function of E as defined by the axioms:

$$\Box(F_j \equiv E_i \wedge \circ E_j)) \quad \text{for each } i, j \ (j = i \pm 1, 1 \leq i \leq 5, 1 \leq j \leq 5)$$

Hence, F_j ($j \geq 1$) is true if and only if E is about to change to E_j. In the paper the dynamics of the system is described by 10 formulas (which are not explained). As an example, the formula

$$\Box(E_1 \wedge \neg H \supset F_0 \wedge \circ E_1)) \quad \text{for each } i, j \ (j = i \pm 1, 1 \leq i \leq 5, 1 \leq j \leq 5)$$

denotes that is always the case that if the temperature is below t_1 (E_1) and there is not heat on ($\neg H$) then the controller registers no change (F_0) and at the next instant the temperature will be below t_1 ($\circ E_1$).

<u>Specifications of desired behavior</u>
Goals of behavior like eventuality and stability can be formulated in the logic. Eventuality is a requirement that the system reaches a desired state:

$$\textit{Initial-conditions} \supset \Diamond(\textit{Final-state})$$

This is read that if the inital conditions are true then it will sometime in the future be true that the system is in the final state.

Stability requires that the system stay in the same state forever after that state has been reached:

$$State \supset \Box(State)$$

Eventuality and stability properties can be merged into:

$$Initial\text{-}conditions \supset \Diamond\Box(Final\text{-}state)$$

which is read that if the initial conditions are true then it will sometime in the future be true that the system will always be in the final state. In our example, the goal that the temperature from some point in the future is always greater than or equal to t_1 and less than or equal to t_2 is written:

$$Initial\text{-}conditions \supset \Diamond\Box(E_2 \vee E_3 \vee E_4)$$

What the initial conditions are, is the subject of the next section.

Synthesis of control

The control of the system can also be formalized as temporal logic formulas. Control is done in order to obtain a goal under the constraints posed by the dynamics. This general statement of system description is translated into logic as:

(SD) $Dynamics \wedge Control\text{-}rules \supset Goals$

When we are able to describe the dynamics and the goals of the system, it is also possible to describe the control rules formally. It is trivial to satisfy (SD) if $Control\text{-}rules \equiv Goals$. However, the formula *Goals* will not tell on a step by step account what to do. We need to synthesize a formula *Control-rules* which can guide the system at each instant.

A formalized reasoning method

The novelty of this paper is the proposal for an automatic synthesis of control rules. The method relies on the construction of structures (models) for the dynamics and goal formulas. As the method is only explained with the help of an example, it is difficult to judge if it is generally applicable. In the example, the *Control-rules* formula is not derived quite from scratch. The suggested form of the rules is:

$$\Box((E_1 \supset H_1) \wedge (E_2 \supset H_2) \wedge \ldots \wedge (E_5 \supset H_5))$$

where each H_i is either H or $\neg H$. This means that whether heat is on or not is a function of the temperature.

For their example, Fusaoka et al. are able to determine the H_i 's by an algorithmic process.

Conclusion

The paper indicates that the are formal reasoning methods which can cope with more than simple logical inferences from a database of information. Although the paper does not mention the rôle of meta-logic, the ideas earn attention when discussing levels of reasoning. This has to be compared with the number of ideas for formal knowledge representations without formal methods for their uses.

Systematic approaches to time representation

In the following sections we will study some approaches to general representation of time and events. The kinds of time presentation we have seen in the preceding sections were all rather specialized. In order to develop general techniques for building expert systems, we must posses some standard tools for the representation of time. Such tools must be the result of the use of exact methods, as opposed to specialized results obtained in an *ad hoc* fashion.

As an simple example of the need of a temporal representation, consider the sentence:

"*If X owns Obj and X sells Obj to Y then Y owns Obj*"

where X and Y are persons, an Obj is an object. This could easily be translated to first-order logic as:

(SELL) $\forall X, Y, Obj: owns(X, Obj) \wedge sells(X, Obj, Y) \supset owns(Y, Obj)$

However, this translation has an unfortunate property. If $owns(X, Obj)$ is true for specific X and Obj , then we can infer using (SELL) that $owns(Y, Obj)$ is true. But then we have that $owns(X, Obj) \wedge owns(Y, Obj)$ is true, which should not be the case as an object normally only has one owner.

What we miss in the above example is the aspect of change. We will return to this example below.

The situation calculus

In a classic paper from 1969 [McCarthy & Hayes 69] AI techniques are used for some philosophical problems as causality, ability and knowledge. It also introduces the *situation calculus* which has caused many discussions and disputes.

A situation s represents the state of the universe at an instant of time. The object of the situation calculus is to describe properties of situations, not to describe them fully as this is impossible. A state is a partial characterization of a situation. An action or event is a function from one situation to another. It is described by a set of conditions of the initial situation, and a set of conditions which hold in the situation obtained by performing the action.

In situation calculus, the new situation that results if the event e occurs in the situation s is denoted RESULT (s, e). A predicate T is introduced to relate a state (partial description) with a situation. T(s, *state*) means that *state* is true in s. With this notation, we can formalize the action of selling from the example just mentioned as:

$$\text{(SELL')} \quad \forall X, Y, Obj, s: \ \mathrm{T}(s, owns(X, Obj)) \supset \mathrm{T}(\mathrm{RESULT}(s, sells(X, Obj, Y)), owns(Y, Obj))$$

This means that for all X, Y, Obj and all situations s, if it is true that *owns*(X, Obj) in s, and if the event *sells*(X, Obj, Y) takes place in s, then in the new situation (RESULT(s, *sells*(X, Obj, Y)) the state *owns*(Y, Obj) will be true.

The well-known *frame problem* is an unfortunate drawback of the situation calculus. Every property which rests unchanged during a shift from one situation to another has to be explicitly stated as being not changing. In fact, every logic of time faces this problem. In interval temporal logic, where a set of intervals is assigned to each property, assumptions of default extensions of these intervals must be added as to avoid a myriad of explicit axioms.

Simultaneous actions cannot be described as an action, because an action is just a function which applied on one situation gives a new situation. As an example (from [Allen 84]): "I walked to the store while juggling three balls".

Another limitation is illustrated with the following example, fetched from [Hayes 85]. Assume a simple world consisting of blocks and a robot. That the block b is held by the robot in situation s is stated $T(s, \mathit{held}(b))$. If there is no other block on a block b, then b is said to be clear. Formally, $T(s, \mathit{clear}(b))$. If block b is placed on block c, then we write $T(s, \mathit{on}(b, c))$. As an example of axiom, "a block is not clear, if there is another block placed on it" is expressed as:

$$\forall\, b, c, s: T(s, \mathit{on}(b, c)) \supset \neg T(s, \mathit{clear}(b))$$

Let us now try to find axioms for the event that a block b being held by the robot is put down on some other block. The function *next* is introduced to denote the next situation, so the situation following s will be $\mathit{next}(s)$. The axiom

$$\forall\, b, c, s: T(s, \mathit{clear}(c)) \wedge T(s, \mathit{held}(b)) \supset T(\mathit{next}(s), \mathit{on}(b, c))$$

does not work, as it states that for any place c (a block or the table), if the place c is clear and if the block b is held by the robot then in the next situation it is true that block b is in the place c. The problem is that we have said that the block held will be put down in every clear space. If we instead use

$$\begin{aligned}&\forall\, b, s: \exists\, c:\\ &\qquad T(s, \mathit{clear}(c)) \wedge T(s, \mathit{held}(b)) \supset T(\mathit{next}(s), \mathit{on}(b, c))\end{aligned}$$

then we only say that the block is being put down in one place. However, we have not stated that the block can in fact be placed in every clear site. This axiom will not exclude "pathological" worlds as worlds where blocks are always placed on a certain block. What we need to state is that for all places c, if it is not inconsistent with other axioms, then it is *possible* that b is placed in c. So it seems that we cannot avoid possible futures. This example can also serve as a critique of the next logic to be mentioned which also does not employ multiple futures.

Allen's logic

In [Allen 84] a general and thorough treatment of representation of actions and time is given. Although the paper does not discuss computational issues, it is clear that the representation is not a "glamour representation" as it is founded on logic. Allen's logic is a many-sorted logic with variables ranging over time, events, properties, actions, and processes. Using this ontology he gives examples of how to represent different courses of events mainly in terms of actions. In another paper [Allen & Kautz 85], it is discussed how reasoning about actions for planning can be carried out algorithmically.

<u>The basic ontology</u>

The fundamental concept of time in Allen's logic is an *interval*. As opposed to a point in time, an interval in time permits closer and closer inspection if we regard time intervals to be analogoues of intervals on the real line. Allen assumes that time is dense. That is, there is always an interval between two non-overlapping intervals.

If we define intervals in terms of time points, we run into a problem when defining intervals. Consider the interval where the proposition "the house is red" was true. After this interval comes an interval where the opposite proposition is true, i.e. the house is not read. For reasons of symmetry the ends of these interval which meet should either both be open or both be closed. In the first case, there would be a point where both the house was read, and the opposite proposition were true. In the second case, there would be a point where neither proposition was true.

This problem could only be overcome by chosing (arbitrarly) intervals to be open at one end and closed at the other end.

Instead, by adopting intervals as primary items, we do not need to decide such questions. The relationships between intervals are instead defined by axioms. Intervals are represented with binary predicates on intervals as (from [Allen 84]):

DURING(*t1*, *t2*)	*t1* is fully contained within *t2*
STARTS(*t1*, *t2*)	*t1* shares the same beginning as *t2*, but ends before *t2* ends
FINISHES(*t1*, *t2*)	*t1* shares the same end as *t2*, but begins after *t2* begins
BEFORE(*t1*, *t2*)	*t1* is before *t2* and they do not overlap
MEETS(*t1*, *t2*)	*t1* is before *t2* but there is no interval between them
EQUAL(*t1*, *t2*)	*t1* is the same interval as *t2*

Some axioms of intervals are:

$$\mathrm{BEFORE}(t1, t2) \wedge \mathrm{BEFORE}(t2, t3) \supset \mathrm{BEFORE}(t1, t3)$$

$$\mathrm{MEETS}(t1, t2) \wedge \mathrm{DURING}(t2, t3) \supset (\mathrm{OVERLAPS}(t1, t3) \vee \mathrm{DURING}(t1, t3) \vee \mathrm{MEETS}(t1, t3))$$

Using the relationships above, the property that an interval is fully contained within another interval can be expressed:

$$\mathrm{IN}(t1, t2) \equiv (\mathrm{DURING}(t1, t2) \vee \mathrm{STARTS}(t1, t2) \vee \mathrm{FINISHES}(t1, t2))$$

Relationships between intervals are not interesting in themselves. We need yet to connect intervals to propositions. Allen makes a distinction between static aspects called *properties* (e.g. a red house) and dynamic properties called *occurrences* (e.g. the house was painted red). The notion of occurrence is divided into two distinct concepts: *events* and *processes*. In the sequel, we will shortly discuss properties, events and processes and show some axioms.

Properties

A property holding over an interval does "fully" hold over this interval. As an example, if the house is red in the interval [1900,1986] then it is red in every subinterval of [1900,1986]. The predicate HOLDS(p, i) is used to assert that the property p holds at the time interval i. The axiom

(PROP) $\mathrm{HOLDS}(p, T) \equiv \forall t.\, \mathrm{IN}(t, T) \supset \mathrm{HOLDS}(p, t)$

says that a property p holds over an interval T if and only if it holds over all subintervals of T. Said in another way, the set of intervals over which a property holds is closed under the IN relation. (Actually, Allen uses a somewhat stronger axiom which will not be mentioned here.)

To allow properties to act as complex logical expressions, functions *and*, *or*, *not*, *all*, and *exists* are introduced. They correspond to the logical operators $\wedge$, $\vee$, $\neg$, $\forall$, $\exists$. Some axioms governing these functions are:

$$\mathrm{HOLDS}(\mathrm{and}(p, q), t) \equiv \mathrm{HOLDS}(p, t) \wedge \mathrm{HOLDS}(q, t)$$
$$\mathrm{HOLDS}(\mathrm{not}(p), T) \equiv (\forall t.\mathrm{IN}(t, T) \supset \neg\mathrm{HOLDS}(p, t))$$

Events

The predicate OCCUR(e , t) denotes that event e takes place over the interval t. As opposed to properties, the set of intervals over which an event occurs contains no intervals such that one is contained within another:

(OCC) $\mathrm{OCCURS}(e, t)\ \&\ \mathrm{IN}(T, t) \supset \neg\,\mathrm{OCCURS}(e, T)$

The following is an illustration of how events can be formulated in terms of properties. Suppose CHANGE-POS($Ball$, x, y) is a function of type event. Given a ball $Ball$, a start position x and a destination y, it gives the event consisting of moving the ball $Ball$ from x to y. For example, that the event "BALL1 moved from POS1 to POS2" occurred over time T100 is written:

OCCUR(CHANGE-POS(BALL1, POS1, POS2), T100)

A necessary condition for a CHANGE-POS event to occur could be defined relative to a property "at":

(CPOS) $\mathrm{OCCUR}(\mathrm{CHANGE\text{-}POS}(Ball, x, y), t) \supset$

$$\exists\, t1, t2.\ \mathrm{MEETS}(t1, t) \wedge \mathrm{MEETS}(t, t2) \wedge \mathrm{HOLDS}(\mathrm{at}(Ball, x), t1) \wedge \mathrm{HOLDS}(\mathrm{at}(Ball, y), t2)$$

This reads that in order for $Ball$ to change position from x to y over time t then there must exist time intervals $t1$ and $t2$ such that $t1$ meets t and t meets $t2$, and $Ball$ is at position x during $t1$ and at position y during $t2$.

Axiom (OCC) states that an event occurs over the minimal interval possible. Thus, the above necessary condition (CPOS) enlarged with

$$\ldots \wedge \neg\,(\exists\, t'.\ \mathrm{IN}(t', t) \wedge \exists\, t1, t2.\ \mathrm{MEETS}(t1, t') \wedge \mathrm{MEETS}(t', t2) \wedge \mathrm{HOLDS}(\mathrm{at}(Ball, x), t1) \wedge \mathrm{HOLDS}(\mathrm{at}(Ball, y), t2))$$

on the right side, would gives us a necessary and sufficient condition for CHANGE-POS to occur.

Processes

Often we talk about something happening over an interval of time without insisting that it happens during every subinterval. "Peter was walking during t" does not imply that he was walking during all subintervals of t. He could have stopped for a period looking at a plane in the sky. For such phenomena, Allen has made a distinct sort in the logic, namely processes. That a process p happens during t, is written OCCURRING(p, t). There is only a rather weak axiom for OCCURRING:

$$\text{OCCURRING}(p, t) \supset \exists t'\text{: IN}(t', t) \wedge \text{OCCURRING}(p, t')$$

which means that in order for p to occur over t, there must exist a subinterval t' of t where p also is occurring.

Actions

In short, the function ACAUSE(*agent*, *occurrence*) gives the action of the *agent* causing the *occurrence*. If the occurrence is an event then the corresponding ACAUSE is called a *performance* (It seems not to be clear if performances are just events). If the occurrence is a process then the ACAUSE is called an *activity*. As an example,

```
OCCUR(
        ACAUSE( Peter,
            CHANGE-POS(BALL1, POS1, POS2)),
        T100)
```

means that the performance ACAUSE(Peter, CHANGE-POS(BALL1, POS1, POS2)) occurs over T100.

Other Concepts

In addition to the above constructs, Allen introduces a variety of other functions to deal with causality, belief, intention and plans. We have picked some of these out to give a flavor of the theory's generality:

COMPOSITE($e1$, $e2$)
: the event consisting of both the event $e1$ and $e2$

ECAUSE($e1$, $t1$, $e2$, $t2$)
: event $e1$ over $t1$ causes event $e2$ over $t2$

GENERATES($a1$, $a2$, t)

action *a1* generates action *a2* over *t*

BELIEVES(A, p, T_p, T_p)

agent *A* believes during T_p that *p* holds during T_p

TO-DO(a, t, p)

the plan *p* include that the action *a* is performed at time *t*

NOT-TO-DO((a, t, p)

the plan *p* include that the action *a* is <u>not</u> performed at time *t*

IS-GOAL-OF(A, g, gt, t)

the agent *A*'s desired world at *t* contains the goal *g* holding during *gt*

COMMITTED(A, p, t)

the agent *A* is committed to act according the the plan *p* at time *t*

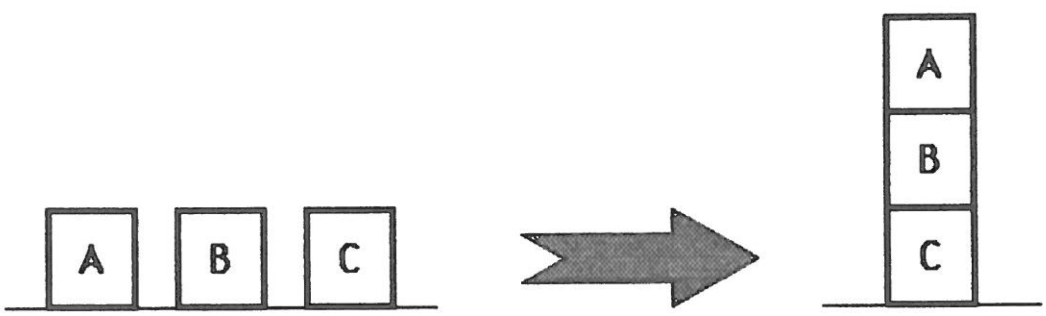
CHANGE-MIND(A, p, t)

the agent *A* believes that *p* holds during *t*, but later believes that *p* does not hold during *t*

Allen has defined several axioms for these functions, but they will not be discussed here for reasons of space. In his most advanced example, Allen shows how to formalize the situation "Sam hid his coat by standing in front of it". This formalization hinges on the use of the above mentioned "exotic" concepts of having beliefs and intentions.

<u>Planning</u>

In [Allen & Kautz 85] some computational questions about Allen's logic are discussed. The paper deals with how descriptions of properties (in the sense just defined) and events can be used for simple temporal reasoning. The main example of the paper is how we can automatize the planning of how to build a tower of three blocks. The blocks are initially on the table; then they are put together:



Initial situation *Final situation*

If *F* denotes the interval where the goal is to be fulfilled, then the goal can be formalized:

$$\mathrm{HOLDS}(\mathrm{ON}(A,\ B),\ F) \wedge \mathrm{HOLDS}(\mathrm{ON}(B,\ C),\ F)$$

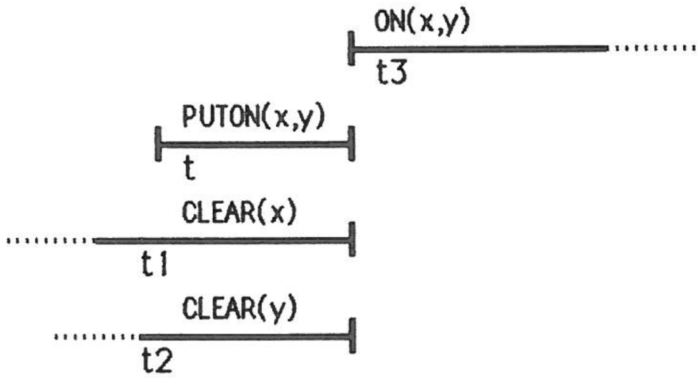
and if I denotes the interval where the initial conditions are true, these can be formalized:

$$\mathrm{HOLDS}(\mathrm{CLEAR}(\mathrm{A}),\ \mathrm{I}) \wedge \mathrm{HOLDS}(\mathrm{CLEAR}(\mathrm{B}),\ \mathrm{I}) \wedge \mathrm{HOLDS}(\mathrm{CLEAR}(\mathrm{B}),\ \mathrm{I})$$

The reasoning includes finding out that B must be put on C before A can be put on B. This condition is calculated using only relationships between intervals, and no axioms about properties or events are used, apart from:

$$\begin{aligned}
&\mathrm{OCCURS}(\mathrm{PUTON}(x,\ y),\ t) \supset \\
&\quad \exists\, t1,\ t2,\ t3: \\
&\qquad \mathrm{HOLDS}(\mathrm{CLEAR}(x),\ t1) \wedge \\
&\qquad \mathrm{HOLDS}(\mathrm{CLEAR}(y),\ t2) \wedge \\
&\qquad \mathrm{HOLDS}(\mathrm{ON}(x,\ y),\ t3) \wedge \\
&\qquad \mathrm{FINISHES}(t,\ t1) \wedge \\
&\qquad \mathrm{FINISHES}(t,\ t2) \wedge \\
&\qquad \mathrm{MEETS}(t,\ t3)
\end{aligned}$$

which states that in order for the event PUTON(x, y) to occur over time t the two objects x and y have to be CLEAR up to and including t, and that t is followed by an interval where ON(x, y) holds. This is depicted below:

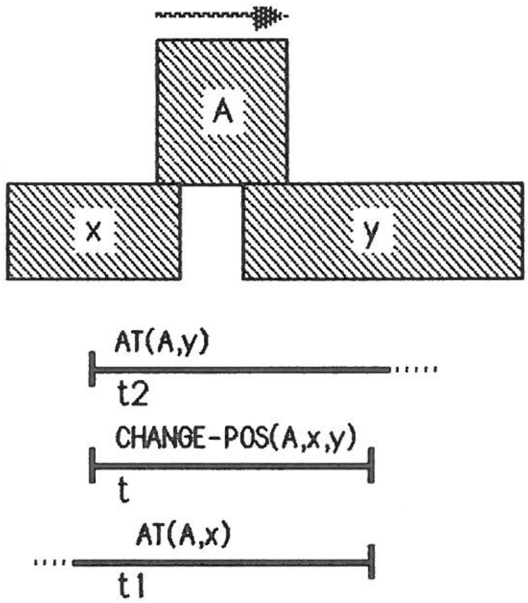


It is assumed that two events, namely PUTON(A,B) and PUTON(B,C) will do the job of bringing us from the inital situation to the final situation. Allen & Kautz do not propose any hints on the cardinal issue of finding such events which have to occur. To both events are associated four intervals so that we altogether get ten intervals. By deriving constraints among the intervals, it is concluded that the PUTON(A,B) event must occur before the PUTON(B,C) event.

<u>A critique of the event concept</u>

It seems that we run into problems with the example in the previous section when trying to apply the rigourous definition of event in [Allen 84]. An event should only occur in the minimal interval possible. But there is *nc* such minimal interval t for PUTON. If the necessary conditions are fulfilled for t, then for any t' such that IN(t', t), the necessary conditions will also be satisfied for t'. The problem is, of course, that the event is instantaneous. Allen has provided no means of expressing that an event takes place without duration. In [Allen & Hayes 85] instants and moments are introduced in a enlarged ontology, but no axioms are given that relate the concepts of time to events, processes etc.

We may also question whether it is reasonable to assume that events always occur over the minimal interval possible. Assume that we are moving a block A from x to y where the block A is so big, and the distance between x and y is so small, that block A does not cease to be at x before it begins to be at y:



Then the definition of CHANGE-POS(A,x, y) would make the interval over which it occurs infinitely small! There seems to be no natural definitition of this interval.

Above, the interval is indicated to begin when both AT(A,x) and AT(A,y) begins to be true, and the interval ends when AT(A,x) ends. This suggests that properties perhaps should be defined over the *maximal* possible interval. This issue will be taken up again in chapter 4.

Conclusion

The ideas of Allen deserve attention as they represent a rigourous and general approach to knowledge representation. As opposed to the situation calculus, simultaneous events and properties can easily be described. We have argued in chapter 3, that intervals are a natural way to talk about time. As opposed to tense-logic, Allen employes a notation requiring a lot of explicit references to time. Therefore, his formulas contain a great number of quantifiers which tend to make them rather hard to read.

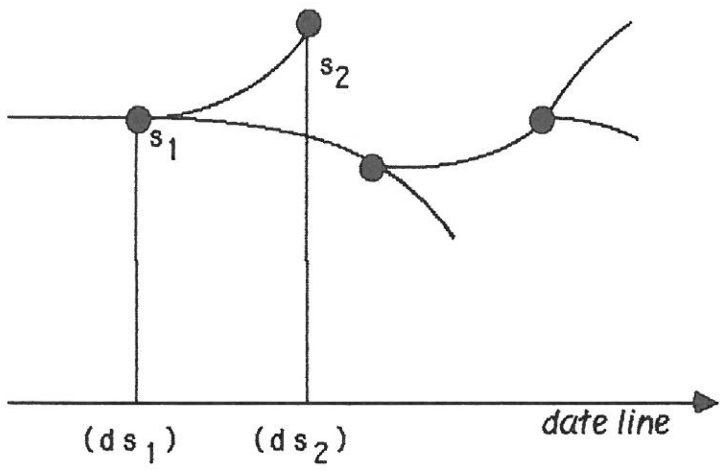
As a consequence, the logic does not mirror normal uses of natural language. By insisting on relationships between time intervals without implicit reference points as "now", it becomes a B-type temporal logic.

The logic is linear, i.e. it has no embedded concept of possible worlds, or possible courses of time. Instead, some "believe" concepts are introduced to permit reasoning about "private" possible worlds. The properties of such private worlds are not studied. As an example, if agent A believes p and the formula "$p \supset q$" is a tautology, is it then true that A also believes q?

It is indicated in [Allen & Kautz 85] that there are complete decision procedures for relationships between intervals. But apart from this, the computational aspects of the theory have not been the subject of thourough analyses.

McDermott's logic

In [McDermott 82] a comprehensive enterprise is undertaken to solve representational problems about time, events, facts, plans, tasks, actions and continuous change in quantities. The underlying ontology is a universe consisting of a tree of states. A state corresponds to an instant. The tree is only allowed to branch in forward direction meaning that a state may have many succesor states or futures, but only one predecessor or past. Every state s has a date (d s). (McDermott uses LISP or "Cambridge Polish" syntax, so $\lambda s.$(d s) is the function which when applied on a state gives a date.) The branching of time is depicted below:



A *chronicle* is a set of states which forms a complete history of the universe, or said in another way, it is the complete branch of the tree. A chronicle is a totally ordered set of states. An *interval* is a subset of a chronicle. An interval is determined by the first state s1 and last state s2 so that the interval can be denoted by [s1,s2]. A *fact* is something that for each state is either true or false. An *event* is something that for each interval either occurs or does not occur over the interval. To an event we may associate the set of intervals over which it occurs. If this set contain no overlapping intervals, the event is called discrete. The definition of overlap is according to McDermott:

```
(iff  (overlap [?s1, ?s2] [?s3, ?s4])
      (or  (=< ?s1 ?s3 ?s4 ?s2)
           (=< ?s3 ?s1 ?s2 ?s4)
           (< ?s3 ?s1 ?s4 ?s2)
           (< ?s1 ?s3 ?s2 ?s4)))
```

This formula is hard to read although its meaning is simple. It is indicative of the rather cumbersome notation found in McDermott's papers.

Conclusion

As opposed to Allen, McDermott employs a branching time logic. This permits a greater expresiveness as mentioned in chapter 2. McDermott gives dozens of axioms. Hence, it may be difficult to see whether they are complete and sound. Due to lack of time, we will not go further into McDermotts ideas, but encourage the ambitious reader to consult them ([McDermott 82], [MdDermott 85]).

Other work

McDermott's group at Yale University, CT has recently come out with three papers. [Dean 85] outlines a mechanism for non-monotonic temporal reasoning involving counterfactuals and disjunctions. [Miller et al. 85] gives an overview of a project called FORBIN which deals with techniques for spatial and temporal planning. The work presented herein seems more to rely on heuristics than logic.

In a short, but comprehensive paper another of McDermott's students, Y. Shoham presents ten requirements for theories of time and change [Shoam 85].

Montague's intensional logic, which incorporates that values of propositions are context- and time dependent, has been implemented on computers [Hobbs & Rosenschein 78]. To the authors knowledge, no papers have yet shown how Montague's logic can be used for planning.

The question of event recognition have also been discussed by [Borchardt 85]. An "event calculus" is introduced. No axiomatic characterization of its properties is given, only some examples of LISP functions and lists.

In [Øhrstrøm & Klarlund 86], a temporal logic employing both A- and B-concepts is presented together with a proof system. A system based on this temporal logic has been implemented in PROLOG. A (relative) completeness result is given: the inferences that can be drawn by the program, are the same as those that can be inferred using the proof rules.

4 Ideas for Future Research

Here, we will shortly present some ideas for future research. In the first section we will try to sketch a proposal for a temporal logic. The two following sections present two unsolved problems of knowledge representation.

An outline of another theory of time and events

In the last chapter we have seen several examples of formal languages for the representation of time and change. Allen's notation, and especially McDermott's notation are not very easy to read. We will here try to formulate some requirements for a representation which is easier to read, or said in another way, more like natural language. A new kind of temporal logic is tentatively developed along with this discussion. Further, it is sketched how properties of such a representation may be formulated as axioms resembling simple algebraic laws.

Requirements

For the sake of simplicity, it will be assumed that time is discrete consisting of the set of integers. An interval is a (perhaps empty) set of consecutive points of the form $\{i, i+1, \ldots j\}$.

<u>Hierarchical representation</u>

In order to easily decompose complicated temporal statements the representation should be hierarchical. Both Allen's and McDermott's logics permit hierarchical composition of temporal statements, but they tend to be obscured by the introduction of explicit intervals and quantifiers over such intervals. It would be better to construct operators that could bind descriptions together more directly, instead of descending to the level of relationships among time intervals.

<u>Temporal descriptions</u>

Natural language seems to refer to intervals by qualitative descriptions as in e.g. "*while I was waiting at the bar it began to rain*". These intervals can then be related using words as "*until*", "*before*", "*while*" etc. Hence, we require that the formalism is build upon a notion of interval. Qualitive descriptions of intervals will be called temporal descriptions. *A temporal description defines a set of possible courses of events.*

A course of events can be described by a sequence of propositions characterizing succesive instants. More formally, a temporal description *d* specifies a set of sequences of instant propositions. Hence, d can be viewed as an element of $\mathbb{P}(\mathbb{P}^*)$ where $\mathbb{P}$ is the domain of propositions characterizing an instant. $\mathbb{P}(\mathbb{P}^*)$ is the domain of sets of sequences of instant propositions.

As an example, the sentence describing complex courses of events as "*The alarm clock buzzed twize before I hit the button*" gives rise to a set of sequences, each describing that the alarm clock buzzed twize before I hit the button.

Occurrence of temporal descriptions

Being equipped with a qualitative aspect, we may for a temporal description d at a given instant in time i talk about whether d occurs at i or not : d occurs if and only if there is a sequence $ps \in d$, $ps = \langle p_1, p_2, \ldots \rangle$ of instant propositions such that the first proposition, p_1, is satisfied at i, the next proposition, p_2, at $i + 1$ etc. We say that the sequence ps is fulfilled at i. If d occurs at i, we may also say that d is true at i.

Past, present, future

Logics founded on the notions of past, present, future seem to be preferably, as they automatically incorporate the concept of "now". It has already been indicated that a temporal description d should denote an interval. So if d occurs at an instant i, it is natural to let the interval associated with d begin at i and last according to the length of a sequence of d which is fulfilled. As there may be several such sequences, we will assume that the length of the largest one is used to denote the length of the interval. More precisely, if $ps \in d$ is a largest sequence that is fulfilled at i, then the interval of the occurrence of d at i is

$$\{ i,\ i+1, \ldots ,\ i + \text{length}(ps) - 1$$

If d does not occur at i, we will not associate any interval with d at i

As an example, "*the block is clear*" is a description consisting of the sequences $\langle c \rangle$, $\langle c, c \rangle$, $\langle c, c, c \rangle$ where c is an instant proposition which is true at the instant i if and only if the block is clear. If this description occurs, it says that the block is clear now, and it will stay clear in a non-empty interval, namely until (but not including the point) where it is not clear anymore (this interval may consist of only one point, namely "now").

<u>The dual nature of temporal descriptions</u>

We have now seen how temporal descriptions should possess a dual nature in a temporal logic. Both aspects are coupled to an implicit "now". Their relationship can be summarized by the following: *The quantity is the maximal amount of time from now over which the quality occurs.*

<u>Names of instances of qualities</u>

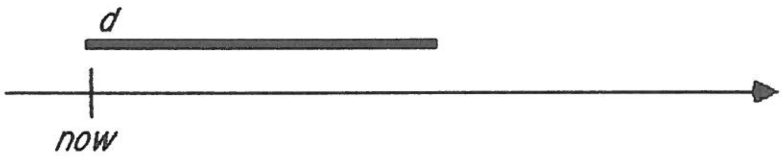
In natural language, temporal descriptions are often refered to using an "it" as "*I was a sailor during the war. It was a dangerous time*". Therefore, our logic should provide means of naming specific instances of temporal descriptions.

A tentative proposal

In this section, we will first define a notion for the occurrence of a temporal description and a notion for the interval associated with such an occurrence. Then a number of operators to combine temporal descriptions will be introduced. The meaning of these operators are explained giving necessarry and suffient conditions for the composed descriptions to occur. For reasons of space, the set of possible courses of events belonging to the composed descriptions will not be given.

<u>Occurrences and intervals of temporal descriptions</u>

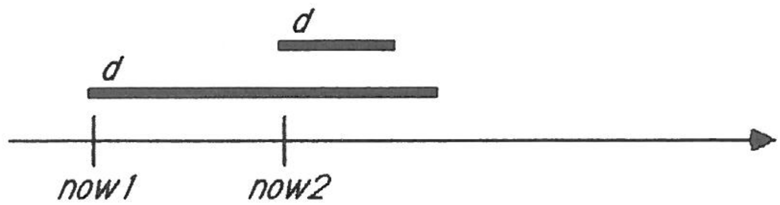
The proposition that the temporal description *d* occurs is denoted Occ(*d*). The truth-value of this proposition changes as time change. The interval associated to an occurrence of *d* is denoted Int(*d*). Below is shown the interval of an occurrence of *d* at an instant *now* :



Note, that if Int(*d*) = [*now*, ... ,] then *d* occurs at *now*, but *d* need *not* occur at any of the other instants in Int(*d*).

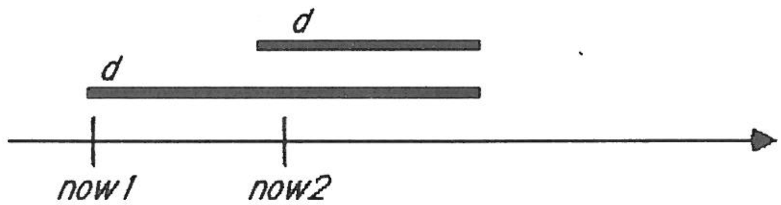
In the sequel, we will let Int(*d*) denote the interval of an occurrence of *d* that is determined by some true proposition Occ(... *d* ...), where the dots signify that *d* can be embedded in a composed temporal description. In the following, when referring to Int(*d*), it will be assumed that Occ(... *d*...) is true.

If *d* is a description of the process of painting something red, then the interval associated to an instant *now1* may contain the interval of *d* corresponding to an instant *now2* later than *now1* :



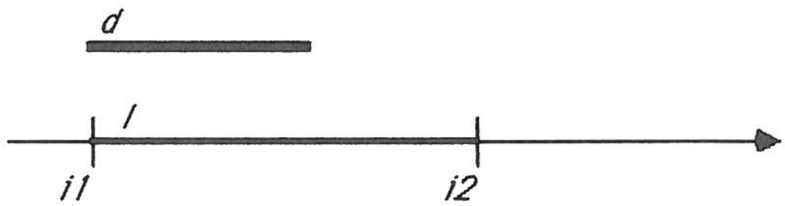
This happens because, the process of painting occurring at *now1* lasts longer than the process of painting at *now2*.

If, however, *d* is a static property such as something is red, then a later interval overlapping with an earlier interval, will finish the earlier interval ("finish" as Allen has defined it):



(This is due to the fact that $d = \{\langle r\rangle, \langle r,r\rangle, \langle r,r,r\rangle \ldots\}$ where *r* is the instant proposition that something is read).

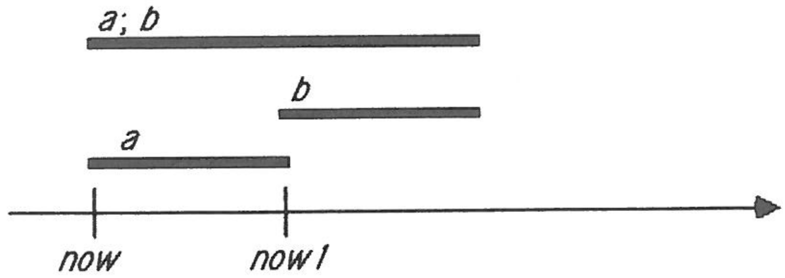
Often, the a temporal description is referred to relative to an already mentioned interval. If *I* is an interval and *d* is a temporal description then we write Occ(*d* <u>in</u> *I*) to denote the *d* occurs in *I*:



i.e. *d* occurs at the instant *i1* (the first instant in *I*) and there is a sequence $\in$ *d* of length not greater than the length of *I*, such that the instant propositions of this sequence are fulfilled along *I*. The length of the interval associated the occurrence of *d* <u>in</u> *I* is the length of the largest such sequence.

<u>And then</u>

An operator ";" is used to compose descriptions corresponding to "*...and then...*". It is called the succession operator. Occ(*a*;*b*) occurs if *a* occurs, and if *b* occurs when the interval of the occurrence of *a* ends. If the occurrence of *a* ends at *now1* , then interval associated with the occurrence of *a*;*b* at *now* is the union of the interval of the occurrence of *a* at *now* and the interval of the occurrence of *b* at *now1*:



We have the following axiom:

$$\mathrm{Occ}(a;b) \supset \mathrm{Occ}(a)$$

meaning that if *a* occurs and then *b* occurs, then also *a* occurs.

<u>It will be</u>

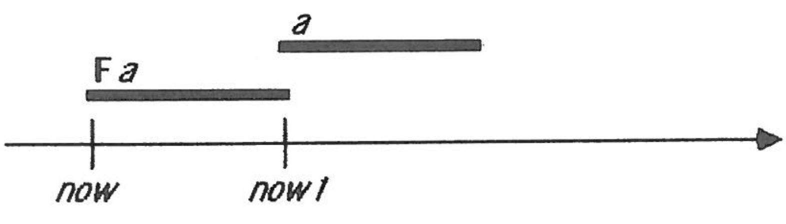
F is the future operator. Occ(**F** *a*) means that *a* will occur now or sometime in the future. (It may be argued that we should not allow Occ(**F** *a*) to be true now as a consequence of *a* occurring now, but this is easy to repair with another future operator.) Clearly,

$$\mathrm{Occ}(a) \supset \mathrm{Occ}(\mathbf{F}\,a)$$

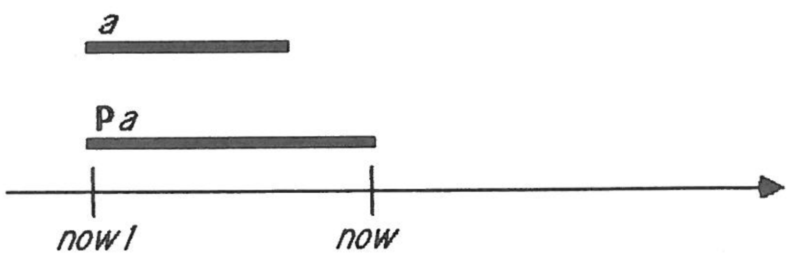
and

$$\mathrm{Occ}(a;b) \supset \mathrm{Occ}(\mathbf{F}\,b)$$

meaning that if *a* occurs and then *b* occurs, then *b* will occur now or in the future (if Int(*a*) is a singleton set then *b* occurs now). The interval associated with **F** *a* is defined to end at the first instant at which *a* occurs:



We also employ a past operator **P** symmetric to the future operator:



The interval of **P** *a* begins at the last occurrence of *a* and lasts until *now*. If *a* occurs at *now* then Int(**P** *a*) = {*now*}. We have:

$$Occ(a) \supset Occ(\mathbf{P}\, a)$$

Temporal descriptions as names

The sentence "*I was a sailor during the war.*" states 1) there was a war, 2) in the interval over which the war occurred I was a sailer. (There is some ambiguity whether this means that I was a sailor during all of the war, or just in a period of the war. We assume the latter interpretation.) Let w be the temporal description of the war, and let s be the temporal description of me being a sailor. Then 1) is translated into Occ($\mathbf{P}\, w$) and 2) is translated into Occ(s in Int(w)). Then the whole sentense "*I was a sailor during the war.*" is translated into:

$$Occ(\mathbf{P}\, w) \wedge Occ(s \ \underline{in}\ Int(w))$$

Note that the interval Int(w) has been determined by the statement Occ($\mathbf{P}\, w$). "*I was a sailor*" is interpreted in the temporal context determined by "*during the war.*" Hence, there is an invisible link from w in Occ($\mathbf{P}\, w$) to w in Occ(s in Int(w)) so that Int(w) is determined by Occ($\mathbf{P}\, w$). If d denotes the temporal description of a dangerous time, then "*I was a sailor during the war. It was a dangerous time.*" is translated into

$$Occ(\mathbf{P}\, w) \wedge Occ(s \ \underline{in}\ Int(w)) \wedge Occ(d \ \underline{in}\ Int(w))$$

where w determined in Occ($\mathbf{P}\, w$) is used twice.

If we want to talk about several distinct occurrences of a temporal description d, an index must be added to each textual occurrence of d. If r is a temporal description which gives the interval consisting of the rest of the day, and c is a temporal description that occurs if the alarm clock rings today, then $r; \circ c$ is the temporal description that the alarm clock rings tomorrow. The temporal description

$$r_1; \circ c_1; r_2; \circ c_2$$

occurs if the alarm clock rings tomorrow (c_1) and if it rings the day after tomorrow (c_2). The indices of r and c are necessary only to distinguish the different occurrences.

Processes
In English, "*a plane takes off*" may denote the whole process of take-off, i.e. as consisting of the phases: the plane moves at a great velocity at the runway, the nose lifts, the body lifts and the plane is in the air. In the sentence "*Before a plane takes off, security checks have to be carried out*", the construct "*a plane takes off*" denotes the whole take-off process. If s denotes the process of making security checks, and b denotes the process of taking off, then the sentence above can be formalized:

$$\mathrm{Occ}(b) \supset (\mathrm{Occ}(\mathbf{P}\, s) \wedge \mathrm{Int}(s) < \mathrm{Int}(b))$$

meaning that if b occurs now, then the last occurrence of s was entirely before b.

But "*a plane takes off*" may also denote a time interval during that process as in "*Just as we arrive, a plane takes off*". This means that a plane is *in* the process of taking off, when we arrive. If a is a temporal description of "we arrive" and b is a temporal description of "a plane takes" of, then "*Just as we arrive, a plane takes off*" should not be translated to

$$\mathrm{Occ}(a) \wedge \mathrm{Occ}(b)$$

as $\mathrm{Occ}(b)$ means that the take-off process begins. What we mean is that a plane *has* begun the take-off process . It would be better to write

$$\mathrm{Occ}(a) \wedge \mathrm{Occ}(\mathbf{P}\, b)$$

meaning that a is occurring and that sometime in the past b was occurring. This is not sufficient, as we have not captured that the process of take-off was still in progress. Instead, in

$$\mathrm{Occ}(a) \wedge \mathrm{Occ}(\mathbf{P}\, b) \wedge \; NOW \in \mathrm{Int}(b)$$

the proposition $NOW \in \mathrm{Int}(b)$ denotes that the occurrence of b has not yet finished. NOW is a function of time, which at instant i has the value i.

Next time
As the time is supposed discrete we may define a next time operator $\circ$, so that

$$\mathrm{Occ}(\circ\, d)$$

means that d occurs at the next moment.

<u>Tensed proposition</u>
We will also allow tensed propositions as e.g. $\mathbf{G}(p)$ to denote that p is true now, and p will always be true:

$$\mathbf{G}(p) \equiv p \wedge \circ p \wedge \circ\circ p \wedge \ldots$$

where $\circ p$ means that the proposition p is true at the next moment.

<u>Simultaneous occurrences</u>
If both a and b occur, we write $\mathrm{Occ}(a \parallel b)$. The meaning of the $\parallel$ operator is formally given by:

$$\mathrm{Occ}(a \parallel b) \equiv \mathrm{Occ}(a) \wedge \mathrm{Occ}(b)$$

Another true formula is:

$$\mathrm{Occ}(a; (b \parallel c)) \equiv \mathrm{Occ}(a;b) \wedge \mathrm{Occ}(a;c)$$

Another example: that $(\mathrm{F}\,a);b$ occurs, means that a will occur and as soon as the interval associated with $(\mathrm{F}\,a)$ ends, then b occurs, i.e. b occurs at the same time as a. Formally,

$$\mathrm{Occ}((\mathrm{F}\,a);\ b) \supset \mathrm{Occ}(\mathrm{F}(a \parallel b))$$

If we want to state the intervals of the occurrences of a and b are the same (end at the same time), we use the operator &, defined by:

$$\mathrm{Occ}(a\ \&\ b) \equiv \mathrm{Occ}(a) \wedge \mathrm{Occ}(b) \wedge \mathrm{Int}(a) = \mathrm{Int}(b)$$

The following should be true:

$$\mathrm{Occ}(a;\ b\ \&\ c) \equiv$$
$$\mathrm{Occ}(a;\ b) \wedge \mathrm{Occ}(a;c) \wedge \mathrm{Int}(a;b) = \mathrm{Int}(a;c)$$

$$\mathrm{Occ}(a \parallel (c;\ d)) \wedge (\mathrm{Int}(a) \cap \mathrm{Int}(d) = \emptyset) \supset \mathrm{Occ}(a;\ \mathrm{F}\,d)$$

The latter equivalence says that if a occurs, and c occurs followed by d, and if the occurrence of a does not overlap with the occurrence of d then the d occurs after (or at the instant where) the occurrence of a has finished.

Or

That it is now true that either the description a or the description b occurs (or both) is denoted Occ(a, b). We have

$$\mathrm{Occ}(a, b) \equiv \mathrm{Occ}(a) \vee \mathrm{Occ}(b)$$

Strict until

If the occurrence of a ends just before the first occurrence of b begins, then we write Occ($a\mathbf{U}b$) which is read that a lasts until b occurs. If b does not occur then there is no condition on how long the occurrence of a lasts. When Occ(a **U** b) then a occurs now and if b is going to occur then it will be just after the occurrence of a finished:

$$\mathrm{Occ}(a\ \mathbf{U}\ b) \equiv \mathrm{Occ}(a) \wedge (\mathrm{Occ}(\mathrm{F}b) \supset \mathrm{meets}(\mathrm{Int}(a), \mathrm{Int}(b)))$$

where meets($I1$, $I2$) means that $I1$ ends just before the instant where $I2$ begins, i.e. if $I1 = \{i1, \ldots, j1\}$ and $I2 = \{i2, \ldots, j2\}$ then $i2 = j1 + 1$.

Repetition

That an occurrence of a lasts precisely until just before the next occurrence of a etc. *ad infinitum* is written Occ(a*):

$$\begin{aligned}\mathrm{Occ}(a^*) \equiv\quad & \mathrm{Occ}(a) \wedge \\ & \mathrm{Occ}(a\ \mathbf{U}\ a) \wedge \\ & \mathrm{Occ}(a\ \mathbf{U}\ (a\ \mathbf{U}\ a)) \wedge \\ & \ldots\end{aligned}$$

Counting

For a description d, #F(d) denotes the number of times d will occur in the future (inclusive now). We have:

$$\mathrm{Occ}(d) \supset \#\mathrm{F}(d) \geq 1$$
$$\mathrm{Occ}(d \parallel \mathrm{oF}\, d) \supset \#\mathrm{F}(d) \geq 2$$

An example

Let us take the example from [Allen & Kautz 85] that has already been studied in chapter 3. The dynamics, goal and strategy are formalized in the following.

Dynamics

It is always the case that an occurrence of puton(A, B) takes place if and only if A is clear, B is clear, and at the next moment, then A is on B:

$$\mathbf{G}(\, Occ(\text{"puton(A,B)"}) \equiv$$
$$Occ(\, \text{"clear(A)"} \parallel \text{"clear(B)"}) \parallel \circ\text{"on(A,B)"} \,)$$

Block A is initially clear, and stays clear until C is on A, or B is on A. Then either B is on A until A becomes clear, or C is on A until A becomes clear, etc. The conditions for B and C are analogous:

$$Occ((\text{"clear(A)"} \ \mathbf{U} \ (\text{"on(C,A)"}, \text{"on(B,A)"})^{*})$$
$$Occ((\text{"clear(B)"} \ \mathbf{U} \ (\text{"on(A,B)"}, \text{"on(C,B)"})^{*})$$
$$Occ((\text{"clear(C)"} \ \mathbf{U} \ (\text{"on(A,C)"}, \text{"on(B,C)"})^{*})$$

That there is at most one block on each block can be formalized:

$$\mathbf{G}(Occ(\text{"on(B,A)"}) \supset \neg Occ(\text{"on(C,A)"})$$

etc.

Goal

It will be the case that A is on B, and B is on C:

$$Occ(\mathbf{F}(\text{"on(A,B)"} \parallel \text{"on(B,C)"}))$$

Strategy

There will be exactly one occurrence of puton(A,B) and exactly one occurrence of puton(B,C):

$$\#F(\text{"puton(A,B)"}) = 1 \wedge \#F(\text{"puton(B,C)"}) = 1$$

Fullfilment of goal

That the strategy fullfils the goal is simply written

(SYST) *Dynamics* $\wedge$ *Strategy* $\supset$ *Goal*

where *Dynamics*, *Goal* and *Strategy* are the formulas given in the three preceding paragraphs. The axioms of the logic should allow (SYST) to be