A THESIS

entitled

A GENERAL APPROACH TO TEMPORAL REASONING ABOUT ACTION AND CHANGE

by

Taoxin Peng

Submitted in partial fulfilment of the requirements for the award of the

DEGREE OF DOCTOR OF PHILOSOPHY
in Computer Science

Supervised by

Prof. Brian Knight
Dr. Jixin Ma

Sponsored by

The University of Greenwich

Acknowledgements

I would like to express my sincere thanks to my supervisors, Professor Brian Knight and Dr. Jixin Ma, for their guidance, and assistance during the development of this research. They have had a deep impact on my research career that cannot be fully expressed. Among many other things, I have learnt the value of clarity and precision in scientific research.

I would like to thank Dr. Don Cowell for valuable comments. I would also like to express my gratitude to my friends who have supported me throughout this process and the staff at the School of Computing and Mathematical Science for their affability and support.

Last and obviously not least, I would like to thank my wife Yanhong, my son Liam and my supportive parents.

The financial support from the University of Greenwich is gratefully acknowledged.

Abstract

Reasoning about actions and change based on common sense knowledge is one of the most important and difficult tasks in the artificial intelligence research area. A series of such tasks are identified which motivate the consideration and application of reasoning formalisms. There follows a discussion of the broad issues involved in modelling time and constructing a logical language. In general, worlds change over time. To model the dynamic world, the ability to predict what the state of the world will be after the execution of a particular sequence of actions, which take time and to explain how some given state change came about, i.e. the causality are basic requirements of any autonomous rational agent.

The research work presented herein addresses some of the fundamental concepts and the relative issues in formal reasoning about actions and change. In this thesis, we employ a new time structure, which helps to deal with the so-called *intermingling problem* and the *dividing instant problem*. Also, the issue of how to treat the relationship between a time duration and its relative time entity is examined. In addition, some key terms for representing and reasoning about actions and change, such as states, situations, actions and events are formulated. Furthermore, a new formalism for reasoning about change over time is presented. It allows more flexible temporal causal relationships than do other formalisms for reasoning about causal change, such as the situation calculus and the event calculus. It includes effects that start during, immediately after, or some time after their causes, and which end before, simultaneously with, or after their causes. The presented formalism allows the expression of common-sense causal laws at high level. Also, it is shown how these laws can be used to deduce state change over time at low level. Finally, we show that the approach provided here is expressive.

Contents

CHAPTER 1
INTRODUCTION

Artificial Intelligence is a field of research that aims to understand and build intelligent machines, especially intelligent computer programs. Beginning with John McCarthy's classic paper, "Programs with Common Sense" [McC59], one of the main goals of artificial intelligence research has been to develop a computer program capable of common sense reasoning about action and change -- specially, a program that is able to perform such tasks as prediction, explanation, and planning. McCarthy envisioned that a program of this kind might "reason" by manipulating explicit, declarative representations of the relevant knowledge. Following this logical approach, the central problems are to discover what this knowledge is and how it might be formalised. In this chapter, the role of common sense reasoning in the research field of artificial intelligence and the motivation of this work are discussed.

1.1 Common-sense Reasoning

Common sense reasoning plays an important role in Artificial intelligence. Intuitively, common sense is the practical, basic knowledge and inference technique that we, as human beings, use every day. It provides us with a general feel for how the objects and people around us behave, tells us what is likely to happen if we perform certain actions, explains how simple physical systems behave, and generally helps us function in the day-to-day world. Common sense knowledge includes the basic facts about events (actions) and their effects, facts about knowledge and the way to obtain it, facts about beliefs and desires. It also includes the basic facts about the material objects and their properties. It is generally regarded that by "common sense" in the physical world, we adopt the view of the classical Newtonian universe. Quantum physics seems to be contrary to common sense in some ways, e.g. the result that a particle can be in two places at the same time, or not at any place with certainty. We shall not consider these "strange" effects in the thesis. Similarly, the Newtonian view is that there is a time defined for all bodies in the universe. So by "common sense" we exclude the concepts of relativity, where twins can age differently and gravity affects time duration. The examples in this thesis all assume a Newtonian universe. One of the features of common sense reasoning that makes it different from traditional mathematical proofs is the use of defaults. The default is a proposition that is postulated to be true in the absence of information to the contrary. The theory of common sense reasoning provides an axiomatic basis for reasoning about the world inhabited by "agents" like us--by agents who have beliefs and goals, who perform actions in order to reach these goals and, by doing so, change the state of the world.

As claimed by McCarthy, one path to human-level Artificial Intelligence uses mathematical logic to formalise common sense knowledge in such a way that common sense problems can be solved by a logical reasoning approach [McC89]. This methodology requires understanding the common sense world well enough to formalise facts about it and ways of achieving goals in it. Formalising common sense knowledge brings with it two benefits. Firstly, the task of actually writing down appropriate axioms, formulae and inference rules forces us to think about how common sense reasoning works -- something we normally just take for granted. Secondly, by producing a formal theory of common sense reasoning we are laying a foundation upon which implementations can be done. An AI system capable of achieving goals in the common sense world will have to reason about what it and other actors can and cannot do. For instance, consider a robot that must act in the same world as people and perform tasks that people give it. In this case, we need to answer the following question: What view shall we build into the robot about its own abilities, i.e. how shall we make it reason about what it can and cannot do?

It has been stated that the common sense knowledge and reasoning is the area in which AI is farthest from human-level, in spite of the fact that it has been an active research area since the 1950s [McC98]. Common sense covers all aspects of the world of human experience. Many rules for different types of knowledge interact with each other in complex ways. To produce a universal theory for common sense knowledge to model the human beings' behaviour would be a vast undertaking. However, it is possible to identify smaller, more manageable, specialised areas of common sense that can be formalised in isolation, e.g., common sense about time, space, actions and change etc. Most of these special theories of common sense have immediate applications in artificial intelligence. More importantly, they could be employed as building blocks with which to construct a full account of common sense reasoning in future. That is the final objective for which the researchers working in this area do their best. The roots of today's research on common sense reasoning are found in the seminal paper by John McCarthy [McC59]. This paper contains the first proposals to use logic in AI for representing and reasoning common sense knowledge, outlines the major objectives of common sense reasoning research, and points out the difficult problems associated with it.

1.2 The Role of Temporal Reasoning in Artificial Intelligence

Temporal information plays a very important role in communication between people as well as in many specialised domains of human activity, such as common sense reasoning. Therefore, work on artificial intelligence also comprises research on representing and reasoning about temporal interdependencies.

Shoham [Sho88] claims that temporal reasoning can be divided according to four main roles:

- Prediction -- to determine the state of the world at a given future time or, more generally, the evolution of the world until a given future time;
- Planning -- given a description of a state at a given time and a goal, to produce a sequence of actions which can be invoked to bring about the goal;
- Explanation -- to produce a description of the world at some past time which accounts for the world being the way it currently is;
- Learning about the physical behaviour of the world -- given a description of the world at different times, to produce a set of rules governing change which account for the regularities in the world.

It is clear that all these four tasks are valuable components in the goals of common sense reasoning and would be equally valuable tools with which to provide AI application systems. For the purpose of developing a common sense theory, general or specialised, such as temporal reasoning about action and change, there are mainly two tasks we have to deal with. The first is to construct a formalism for reasoning about time. This formalism must reflect our own general common sense approach to temporal reasoning and, at the same time, support more detailed temporal inferences that might be needed in specialised AI domains. The second is to use this formalism to formalise common sense notions, such as facts, actions, events and change etc.

Within the last three decades, there has been a strong interest in proposing approaches that can formalise common sense, in terms of reasoning about action and change. A great number of formalisms have been developed for this purpose. In many of these approaches, temporal features are added for the purpose of enriching expression. The key intuitions motivating this development are the following:

1) States persist over time (situations take time). Very few states are instantaneous. Normally a state will hold over a period of time until some action occurs to change it. For example, if John parks his car in the university's car park, the state that his car is in the car park should persist over a period of time until John drives it away or someone steals it.

2) Actions take time. Very few actions are instantaneous: a moment's reflection reveals that the effects of many actions result from applying some force over a period of time. For instance, to lift a quite heavy object up to a van from the ground, it will take some time with positive duration for people to do it.

3) The relationship between actions/events and their effects is complex. Some effects become true immediately after the end of the event and remain true for some time. For example, when John parks his car in the street, this has the effect that the car is in the street for at least a short time after the action/event. Other effects only hold while the event is in progress. For instance, consider a flashlight with a button for flashing it. The light is on only when the button is being pressed down. Finally, some effects might start to hold some time after the event. For example, If someone presses the button at the crosswalk, 25 seconds later the pedestrian

crossing light will turn to yellow from red. That is there is a delay time between the event and its effects.

4) Actions/events may interact in complex ways when they overlap or occur simultaneously. In some cases, one action might interfere with or change the effects of another action while it is occurring. In other cases, the effect of performing two actions may be completely different from the effects of each in isolation. As a radical case of this, consider a wristwatch that is controlled by two buttons A, and B. Pressing A changes the mode of the display, while B shows the alarm time setting. Pressing A and B simultaneously, however turns the alarm on or off. The effect of performing the actions simultaneously has no causal relation to the effects of the actions performed in isolation. There may be additional effects, or some of the usual effects of one of the actions may not be realised. In fact, the effect of two actions together could be virtually independent of the effects of each action done alone. Many activities (e.g., carrying a piano) cannot be accomplished without close co-ordination and simultaneous actions by multiple agents.

5) Change may be continuous. The change caused by actions/events can be divided into two classes: discrete change and continuous change. It is easier to represent a discrete change. However sometimes we need to represent continuous change, such as the height of a falling object, the level of water in a filling tank etc.

6) Actions may interact with external events beyond the agent's direct control: for example, to sail across a lake, the agent may put up the sail and so on, but the action will not succeed unless the wind is blowing.

7) Knowledge of the world is usually incomplete and unpredictable in detail. Thus prediction can only be done on the basis of certain assumptions. Also, some effects of actions may become uncertain, and actions themselves may occur uncertainly. For example, consider a robot with a sensor. Since its sensor data may be both incomplete, due to the robot's limited window, and uncertain, due to sensor noise, the robot's activities can be uncertain as well.

1.3 Logic-based Representation

Logic is a precise formalism that allows one to assign a clear meaning to the elements of a theory through formal semantics, thus removing scope for the possibility of ambiguity. By its very nature it encourages rigorous treatment of semantic issues that might otherwise be overlooked. Logic does not commit us to using a particular programming technique or language; instead it provides a general-purpose declarative representation scheme. The argument for the fundamental importance of logic in knowledge representation is very simple. Pat Hayes sums it up in the following two questions.

> *One of the first tasks which faces a theory of representation is to give some account of what a representation or representational language means. Without such an account, comparisons between representations and languages can only be very superficial. Logical model theory provides such an analysis*. [Hay77]

> ... *virtually all known representational schemes are equivalent to first-order logic* (*with one or two notable exceptions, primarily to do with nonmonotonic reasoning*). [Hay85]

As point out by Moore [Moo82], the overall claim of such an argument is weaker than the claim that the language of formal logic itself should be used as a representational formalism in AI programs. In fact, the underlying meaning of any representational scheme is supplied by logical model theory. The closer the actual representational formalism is to the language of logic, the easier it is to reveal this meaning.

To make the argument more clearly and precisely, Shanahan summarises it as the following three main steps [Sha97]:

- We need to supply an account of what the representations used in AI programs mean. In logic, model theory provides the basis for such an account.

- Almost all representational formalisms are equivalent to first-order logic.
- A model-theoretic account of the meaning of a representational formalism pins down the notion of correct representation for that formalism. It can also play a central role in pinning down the notion of correct reasoning, even if that reasoning is non-deductive.

The formalisms of logics have been used to differing extents in AI. The use of logic can be distinguished as four levels following John McCarthy's catalogue [McC89]:

- A machine may use no logical sentences--all its "beliefs" being implicit in its states.
- The second level of use of logic involves computer programs that use sentences in machine memory to represent their beliefs but use rules other than ordinary logical inference to reach conclusions.
- The third level uses first order logic and also logical deduction.
- The fourth level is a goal. It involves representing general facts about the world as logical sentences. Once put in a database, the facts can be used by any program.

A key problem for achieving the fourth level is to develop a language for a general common-sense database. According to the above argument for logic, how to represent the effects of actions and events in logic is a natural question. Therefore, developing an approach for reasoning about action and change becomes one of the basic and main tasks for achieving the fourth level of the use of logic.

1.4 The Motivation

To model the dynamic world, reasoning about action and change based on common sense knowledge is one of the most important and difficult tasks in the Artificial Intelligence research area. The problems that arise in, for example, either classical mechanics or the historical framework, are symptomatic of all systems for reasoning about action and change. The general problem is how to express effectively what a program knows and how it should reason.

Over the past half century, numbers of approaches have been proposed for dealing with this problem, including McCarthy and Hayes' framework of the situation calculus [McC63, McH69], which is probably the most influential formalism regarding this area. It is an early proposal that takes evolution over time into account. It is originally introduced as a general framework for the modelling of the dynamic worlds; i.e., worlds that change over time. Since then several extensions to the framework have been proposed to add temporal features into the situation calculus (e.g., [Sch90, GLR91, PiR93, 95, MiS94]) in order to enrich the temporal ontology of the formalism. Some other influential formalisms for dealing with this problem include [McD82, All84, KoS86, Sho88, AlF94, StM94, San94, Sha95, GoG96]. However, these approaches have not gone as far as one would like for dealing with temporal issues in representing and reasoning about actions and their effects, and there are still some problematic issues that have not been satisfactorily solved.

Generally speaking, it is assumed that the world persists in a given state until some action is carried out to change it into another state; also, while some actions may be instantaneous, most of them perform over some interval of time. Hence, intervals are needed for expressing the time spans of situations and actions. Most existing formalisms usually associate entities such as fluents, situations/states and actions with some special time, where time elements are characterised as points and intervals are constructed out of points. For instance, in Pinto and Reiter's formalism [PiR93, 95], the time span of a given situation is characterised in terms of its starting point and ending point during which no fluents change truth values. Some problematic issues will arise when one is going to enrich the temporal ontology by adding a time line to the situation calculus as suggested by Pinto and Reiter [PiR95]. As for the time structure, there are mainly two problematic issues.

One issue is that only points are addressed as primitive and hence time intervals have to be constructed from points. Therefore, for representing actions and situations (or, in the terminology of the event calculus, events and processes) which may last for some possible duration, one has to explicitly express their corresponding starting time and ending time. This may lead to the so-call *dividing instant problem* (DIP) [Vil94], i.e., the problem of specifying whether time spans of situations or actions are closed or open at their starting/ending points. To deal with this problem, there are two approaches. One is to take intervals including their ending-points, so that the adjacent intervals would have ending-points in common. Hence, if two adjacent intervals correspond to states of truth and falsity of a given fluent, there will be a point at which the fluent is both true and false. Similarly, if all intervals don't include their ending-points, there will be points at which the truth or falsity of some fluents is undefined. Another approach is to take point-based intervals as semi-open (e.g., all intervals include their left ending-points, and exclude their right ones) so that they may sit conveniently next to one another. However, on the one hand, since this approach insists that every interval contain only a single ending-point, the choice of which ending-point of intervals should be included/excluded seems arbitrary, and hence unjustifiable and artificial. On the other hand, although the approach may offer a solution to some practical applications, there are some other critical questions which remain problematic (examples are given in [Gal90, KnM92]). The fundamental reason is that in a system where time intervals are all taken as semi-open, it will be difficult to represent time points in a consistent structure so that they can stand between intervals conveniently.

Another issue is that the negation of a given fluent, and the relationship between a negative fluent and sentences which involve the fluent have not been formally addressed. This is a very important issue that needs some careful treatments if time intervals are allowed to be arguments to some *global predicates* [Sho87a] such as “TRUE”, “HOLDS”, etc. In fact, we may face the possibility that some fluents might be neither true nor false *throughout* some specified intervals. Additionally, in a logic where some time intervals are characterised as infinitely decomposable, the so-called *intermingling problem* [Ham71, Gal96] may arise, that is, the possibility of indefinitely intermingled time intervals within each of which a fluent *f* takes both true and false values. This will lead to some difficulties in characterising the relationships between the negation of a fluent and the negation of the corresponding sentence involving that fluent.

Beyond the time structure itself there are some other problematic issues that have been mostly neglected in most of the existing systems [GLR91, PiR93, MiS94, Sha95 etc.]. One of them is to express the temporal relationships between actions and their effects. This is in fact quite complex and interesting. Normally, the effects of an action become true immediately after the end of the action. However in some cases, the effects of an action may start to hold true some time after the end of the action. That is there is a delayed time between an action and its effects. In most of the existing formalisms, only the former case has been dealt with. Gelfond *et al*. [GLR91] propose an approach using the notation *Duration of Actions* and *Actions that Involve No Activity* to describe an action with delayed effects. This action normally is called *Wait*. Its main function is to fill the gap between action *a* and its effect, in case there is a time delay there. That is: $Result(S_0, a+Wait)$, where the duration of *Wait* equals the time delay. However the null actions only move time forward but have no effects. As pointed out by Allen and Ferguson, it is "neither convenient nor intuitive" [AlF94].

Also, the treatment of Gelfond *et al* to the delayed effects of actions raises another issue in temporal reasoning of actions and change. This issue is how to deal with the relationship between a time duration and its relative time entity. To see this issue more clearly, let us consider the example a ball thrown into the air. While the ball is going up (say for just 8 seconds), the velocity is not zero (and again, not zero when the ball is going down). Only at the apex (the stationary point) where the ball is neither going up nor going down, the velocity becomes zero. Now, how to express the (delayed) effects of the action of throwing the ball? If we take the approach proposed by Gelfond *et al*, using an action called *Wait* to fill the duration, one may use $Result(S_0, Throw+Wait)$ to represent the situation 8 seconds after the action *Throw* (here, we assume the duration of *Wait* is 8 seconds). However, in this result situation, is the velocity zero or not? The answer is not unique.

In fact, there are two situations, one is the situation where the ball is at the stationary point, another one is the situation immediately after the stationary point, where the ball is going down. Both of them satisfy that the *Wait* action lasts for 8 seconds. Gelfond's approach seems unable to distinguish these two delayed effects. This issue has been in fact neglected in the existing temporal reasoning systems.

Additionally, in existing systems, there is a restriction on causal relations, e.g. *causes strictly precede their effects*, or similarly, *there is no change during the performance of actions*. This restriction also implies that there are no situations defined during the time an action is executing. This means that one can't assert anything about the world during the time while an action is being performed. Rather, all effects of an action must be packaged into the resulting situation. As a consequence, all effects start simultaneously. This avoids the simultaneity of a cause and its effects. However, "it may be argued that this is an overkill" [Sho88]. Consider a scenario in which the status of a light is controlled by a button. The light is on only when the button is being pressed down. Intuitively, it seems that the "cause" (pressing the button) coincides with the "effects" (the light is on), rather than preceding it. How do we reason about this scenario formally? Systems with such a restriction can not offer successful solutions for this question.

The other problematic issue beyond the time structure is about the expression of situations. In the various versions of the situation calculus, the word "situation" is used just as a kind of context or time label referring to the set of facts which describe the corresponding state of the closed world of discourse; and, sometimes, it is even used interchangeably with the word "state" (e.g., [GeN87]). However, there are indeed some fundamental differences between the concept of situations and the concept of states of the world if temporal issues are taken into account. Hence, in order to fully capture the temporal aspects of the dynamic world, the following separation between the notion of states and situations is expected:

- The state of the world, which may be described by a certain set of facts, is time independent - the state at a given time does not necessarily have to be different from the state at another time.
- Situations are time dependent - a situation is an association of a given state of the world with a particular time over which the world holds in that state. Hence, each situation must be taken as unique. In other words, two situations that refer to two states of the world associated with two different times, respectively, must be taken as different, no matter whether the states of the world at these two times are the same or not. Additionally, since each situation is associated with a particular time,

there should be, on intuitive grounds, a constraint that excludes the case that an action may change a situation to an earlier situation.

Another issue concerns is how to characterise the notion of actions and events. In the literature, the two words, "action" and "event" are used in many different senses by various researchers; sometimes they are also used interchangeably. In fact, similarly to the separation between states and situations, as argued in chapter 6, actions and events should be treated as different entities:

- Actions are time independent - they are simply some names denoting certain useful and relevant activities that may be conducted over some time by the agents to accomplish changes of the state of the world. E.g., "Shooting", "Striking", "Running", "Sleeping", and so on. Following Gooday and Galton's idea [GoG96], in this thesis, we shall call the association of a given action with a certain temporal duration an action type, e.g., "Striking for 24 hours", "Running for three minutes", and so on. It is important to note that the same action can contribute various action types. For instance, "Running for three minutes" and "Running for two hours" denote two different action types. Action types are high level entities - they are still time independent: for any given action type, it may perform once, more than once over different times, or may not even perform at all.

- Events are time dependent - each individual performance of an action type over a given time constitutes an event at low level. E.g., the performance of action type, "Striking for 24 hours", over the whole day of the 16th of September 1994 constitutes the event, e = <Striking, 16/09/1994>.

In addition, generally speaking, to reason about action and change, there are four main questions that need to be answered. One is whether a proposition is true or not. The second is when it becomes true. The third is how long it may persist. The final one is what causes it to be true. Most of the existing formalisms only deal with some of these questions. For instance, the situation calculus and its extensions answer the first, the second and the fourth questions while the event calculus and its extensions answer the first three questions. Therefore, an approach that may reduce such restrictions and provide more power of expression, for instance, to deal with all four above questions, is very desirable. This dissertation provides such an approach.

Finally, following Allen and Ferguson's view [AlF94], most of the approaches for temporal reasoning about actions and change can be classified into two classes, constructive models (state-transition-based) [McH69, Sch90, Lif87, PiR93, 95, MiS94, GoG96, etc.] and non-constructive models (temporal-logic-based) [McD82, All84, KoS86, Sho88, AlF94, StM94, etc.]. Each class has its own advantages. However, one may ask the following questions: Is it possible to develop a new formalism that combines most of the benefits of both of those two classes? If yes, how to do it? This is one of the problems that this thesis will deal with.

1.5 Contributions of the Work

The main contribution of this thesis is the development of a general formalism for temporal reasoning about action and change. This new formalism allows expression of high level common-sense knowledge about action and change, and also supports explicit representation of time and occurrence of events at low level. The contributions of this work can be detailed as follows:

- The formalism proposed here allows a comprehensive characterisation of the relationship between the negation of a given fluent and that of involved sentences.

- A new time theory is introduced which can be seen in fact as a special model of the time structure previously proposed by Knight and Ma in [KnM92, MaK94]. Based on this time structure, a discrete model has been presented. This model overcomes/bypasses the so-called *intermingling problem* and the *dividing instant problem*, which beset most existing temporal theories involving time intervals, while retaining most of the advantages of the general theory.

- For temporal reasoning about actions and change, the issue of how to deal with the relationship between a time duration and its relative time entity is quite interesting and important. The notion of duration type is introduced. The definition of duration types specifies the end points of a time duration, in order to map one temporal duration to a unique time entity. Using this method, some problems, such as representing delayed effects, coincident effects etc., which affect some of existing systems, are successfully overcome.

- The formalism formulates some key terms for representing and reasoning about actions and change, such as states, situations, actions and events. The distinction between states and situations is formally made by defining a situation as a pair of a state and a time over which the world holds in the state. In an analogous way, a formal distinction between actions, action-types and events is proposed, which allows the expression of common-sense causal laws at high level. It is shown how these laws can be used to deduce state change over time at low level, when events occur under certain preconditions. By using this formalism, it is possible to represent truth value of any given fluent over various times, and model the concurrency of actions and events.

- To deal with the frame problem, one of the most important and difficult problems in reasoning about actions and change, two of the conventional nonmonotonic reasoning techniques, i.e. causal minimisation and state-based minimisation are correctly extended. As side effects, the other two related problems, e.g. the ramification problem and the qualification problem, are also addressed.

- The formalism combines many of the existing techniques into a unified, formal framework. It allows more flexible temporal causal relationships than do other formalisms for reasoning about causal change, such as the situation calculus or the event calculus. It includes effects that start during, immediately after, or some time after their causes, and which end before, simultaneously with, or after their causes. The causal axioms guarantee the common-sense assertion that "*the beginning of the effect cannot precede the beginning of the cause*".

1.6 Outline of the Thesis

The outline of the rest of this thesis is as follows. In chapter 2, some basic and major issues about temporal knowledge representation are addressed, such as times, states/situations, actions/events and the causal relationships between actions/events and their effects etc. A review of some representative temporal reasoning systems is given in chapter 3. The main features of these systems and comments upon the relative merits and disadvantages of these are examined. In chapter 4, the discussion is focused on the nonmonotonic techniques for the three classical problems, i.e. *frame problem*, *ramification problem* and *qualification problem*. Firstly, what are these problems and the brief discussion on each problem are described. Then the discussion about the existing solutions to the frame problem in detail is presented. Chapter 5 introduces a time theory, which can be seen as a special model of the time structure previously proposed by Knight and Ma in [KnM92, MaK94]. Based on such a time structure, in chapter 6, a discrete model called *Temporal State Transition Calculus* is presented, which gets some benefits from the time structure. Some key terms regarding knowledge representation, such as duration types/action types, states/situations, actions/events are formulated and the temporal relationships between actions/events and their effects are explicitly examined. Chapter 7 presents the expressive power of the formalism in terms of providing some example applications. Finally, chapter 8 provides a summary and some concluding remarks.

In this thesis, I will be using the first-order predicate calculus with equality throughout, with the following conventions:

$\wedge$	and,
$\vee$	or,
$\Leftrightarrow$	equivalence,
$\Rightarrow$	implication,
$\models$	entailment,
$\exists$	existential quantifier,
$\forall$	universal quantifier,
$\neg$	negation,

Also, generally, a variable is any string of alphanumeric characters beginning with a lower case letter, while a constant, function symbol, or predicate symbol is any string of alphanumeric characters beginning with an upper case letter. In any formula, for the purpose of simplicity, unbounded variables are universally quantified.

CHAPTER 2
MAJOR ISSUES IN
TEMPORAL KNOWLEDGE REPRESENTATION

An understanding of the elementary concepts of temporal knowledge representation has been proved important for many AI applications, e.g., planning, diagnosis, intelligent database and natural language understanding, etc. Elementary terms include times, fluents, actions, change, effects and causation and so on. Among them, time plays a fundamental role. However, the manner in which time is represented and used in AI differs from that in rigorous formal disciplines such as mathematics and physics. In the mathematician's approach, time is usually treated as an independent variable expressed explicitly. AI researchers, on the other hand, sometimes avoid the use of isolated times and instead use times with respect to some other elementary terms. The advantages to the AI approach include expressive power and flexibility but it has proved hard to produce a formal theory of common sense temporal reasoning without compromising the explicitly expressed time. Typically, a trade-off must be reached in which the expressiveness of the temporal language is balanced against its compactness and implementability. This means we are facing the following choices: What primitive temporal objects should be represented? What kind of time structure should be introduced? What mechanisms should be used to relate facts to times? What specialist features should be built in the theory? These issues will be examined in this chapter and some further discussions will be given in chapter 6.

2.1 Intervals and points

In order to design a system for temporal reasoning, a selection of the underlying time structure is necessary. This is the issue of what should be taken as the primitive elements of time. The question of whether to represent time in terms of intervals, points or both is an open one. Practically, the choice depends on the nature of the intended application. However, in many domains both intervals and points are required and so even if we employ only one as a temporal primitive it will still be necessary to have some method of deriving the other. As we shall see, this is not always straightforward and can result in a number of practical and philosophical difficulties.

2.1.1 Time intervals

The main feature of time interval is *duration*. In real world, most of temporal information involves duration. For example, Allen [All83] gives the following story:

> *Ernie entered the room and picked up a cup in each hand from the table. He drank from the one in the right hand, put the cups back on the table, and left the room.*

In this account we can identify several time intervals, e.g. the time Ernie was in the room, the time between entering the room and picking up each cup, the time between putting down the cups and leaving the room, and many others. Therefore, the claim is that intervals are sufficient for modelling all the temporal references in human accounts such as this.

In fact, the observation is that the only times we can identify are times of occurrences and properties. For any such time, say the time when Ernie opens the door, it appears to be possible to look more closely at the occurrence and decompose it. For instance, Ernie turns the handle first and then pushes the door to open. Hence, times can be decomposed into subtimes. In other words, it seems that there is always a more detailed causal explanation if one cares, and is able, to look for it. A good analogy, then, is that times correspond to duration, i.e. to intervals on real line. For this reason it is essential that we have some way of representing time intervals.

For the purpose of knowledge representation, an interval is a continuous chunk of time with duration. Normally, there are two alternative ways of incorporating intervals into a formal knowledge representation scheme. In the first, intervals are assumed to consist of points, and hence, the corresponding systems may be considered as models of point-based time theories. An example of this kind of interval is the *time-segment* of Bruce's model for temporal references [Bru72]. However, as Allen has commented [All81, 83], modelling intervals by taking their ending-points can lead to problems: the annoying question of whether ending-points are in the interval or not must be addressed, seemingly without any satisfactory solution. The second treatment takes intervals as primitive objects without any definitions of the "ending-point" and "internal-point" structures. Allen's interval logic [All81, 83, AlH89], Vilain's temporal system [Vil82, ViK86], Knight and Ma's extended temporal model [KnM92, KnM93], are examples that treat intervals as primitive.

As an example of the first choice, let us look at Bruce's temporal model. Bruce's model is an early attempt at mechanising part of the understanding of time within artificial intelligence. In this system a formal framework, based upon first-order logic, is established for the analysis of tenses, time relations, and other references to time in natural language. The axioms of the framework are based on the following definitions: A *time-system* is a pair, (*time*, ≤), where *time* is a set whose elements are called *time-points*, and ≤ is a partial order over *time*.

Bruce then defines point-based intervals, termed *time-segments,* as chains which are convex in the sense that there are no points missing within the chains, where a chain is a totally ordered subset of *time-points*. The related issues about time-segments, such as: density and linearity, may hence be derived from the corresponding issues of the time-points which make up the time-segments. The ordering relations between segments are also inherited from the partial order over the time points. Bruce gives seven binary relations between *time-segments,* which can be derived from the ordering relations over their greatest lower bounds and the least upper bounds: *Before, During, Same-time, Overlaps, After, Contains* and *Overlapped*.

Following Bruce's idea, briefly speaking, we can define an interval like this: given times t_1 and t_2 (integers, rationals or reals), the endpoints of an interval i, the interval i can be defined as the set of all time points between t_1 and t_2 (it will be convenient to write such an interval as (t_1, t_2), where $t_1 < t_2$). If we choose to include the endpoints within the interval itself then we say that it is closed. If not, then the interval is open. If the interval includes t_1 but not t_2, we say it is left-closed. Similarly, we can define the right-closed intervals.

Additionally, as argued by Allen (see chapter 5), there are some problems in dealing with the treatment of open or closed intervals. Mechanisms for duration reasoning are not specified, although these may be defined by introducing a mapping from the time-points to the reals.

Allen's model is a typical example of the second choice. Instead of adopting time points, Allen takes intervals as the primitive temporal quantity, as being the natural means of human reference to time. Allen argues that even references to apparent point events, such as the time Ernie entered the room, or the time that he put down a cup, are best modelled as small time intervals.

In order to express temporal relationships over time intervals, Allen took originally as primitive a set of nine (mutually exclusive) basic binary relations between any two intervals [All81], extended later to 13 [All83]: *Equal, Before, Meets, Overlaps, Starts, Started-by, During, Contains, Finishes, Finished-by, Overlapped-by, Met-by, After*. These are based on Bruce's seven relationships, but whereas Bruce's relations are derived from the partial order within a point-based theory, Allen's are taken as primitive.

The most disputed aspect of Allen's system is its exclusion of time points as primitive, although in the later paper [AlH89], Allen and Hayes define a point as the "*meeting place*" of intervals, or as a maximal set termed "*nest*", of intervals that share a common intersection, at a subsidiary status within the theory; and use the concept of a "moment", i.e., a very short interval which is non-decomposable, to model some instantaneous events. The contention is that nothing can be true at a point, for a point is not an entity at which things happen or are true [All83]. Except for the assumption that moments have positive length, while points have zero length, another obvious structural difference between points and moments is that moments are treated as primitive objects, and hence can meet other intervals (although they are not allowed to meet other moments), while points are not treated as primitive objects and cannot meet anything at all [AlH89].

2.1.2 Time points

A time point differs from an interval in that it has no duration. Although most of the temporal information that we wish to represent can be expressed in terms of intervals there are some critical cases that require points. For example, when two billiard balls are rolled towards one another the exact moment at which they first touch (change from being not in contact to being in contact) is clearly a time point rather than an interval. In addition, sometimes we practically treat intervals as points. For instance, consider an integrated circuit system, when we close a switch there will be a time delay before current flows through the entire device, but it is simpler to ignore the delay and assume that the circuit functions immediately or, in other words, the circuit can be switched on or off at a point in time rather than over an interval.

McDermott's temporal logic gives an example of this case. McDermott [McD82] develops a first-order temporal logic to provide a versatile "common-sense" model for temporal reasoning. In accordance with the "naive physics" advocated by Hayes [Hay78], McDermott adopts an infinite collection of states (points) as the primitive temporal elements and adds several crucial axioms. Every state has a time of occurrence, *d*(s), a real number called its *date*. Time is assumed to be a continuum, with an infinite numbers of states between any two distinct states, where states are partially ordered by the "*no later than*" order relation "≤". The future (not the past) is branching, that is, there are many possible futures branching forward in time from the present. Each single branch, called a "*Chronicle*", consists of a connected series of states and is isomorphic to the real line.

As argued by many researchers [e.g. Vil82, MaK94], for general treatments, both time points and intervals are needed to represent the full spectrum of common sense temporal knowledge. We have already examined the consequences of adopting points as primitive objects and defining intervals in terms of these in last section. On another hand, in the literature, there are many examples of points defined in terms of interval structures. In different interval structures, time points can be defined by different ways [Haj96].

2.2 Fluents, States and Situations

The treatment of temporal information from the artificial intelligence point of view is concerned principally to represent the knowledge, especially the description domains. Whatever domain is concerned in this kind of treatment, the first stage always consists of defining a *representation* for temporal information, and it is at this level that we find the fact that time is one of the important features. However, additionally, there are some other features need to be declared. For example, to give a complete description of the world, other than the relative times, we need to express the features of status of the world, in terms of, e.g., facts, propositions, states, etc. In this section, we will examine the definition and properties of these notions.

2.2.1 Fluents

An atemporal knowledge representation system can be used to represent facts about an (unchanging) world in terms of the objects of that world and the corresponding properties. Informally, we refer to these properties as fluents. Predicate calculus has proved to be an idea vehicle for knowledge representation of this sort. For example, in the Blocks world, "block A is on the table" and "block A is not on block B" are two fluents in the Blocks world and can be represented as:

$$On(A, Table)$$

$$\neg On(A, B)$$

This kind of fluent denotes the relationship between two different objects. Here the objects are blocks A, B and *Table*. Sometimes a fluent may just denote a status of one object. For instance, $On(Light)$ means that the *Light* is on.

The above discussion just talks about fluents as being simply true or false. This is not the whole thing about fluents. We can extend this sort of representation scheme to the temporal domain, therefore, the truth values of fluents are supposed to be dependent on times and we need a method to associate fluents with times at (over) which they hold true. One way of doing this is to introduce a special temporal incidence predicate, *Holds*, which takes a fluent and a time as arguments. This predicate evaluates to true or false according to whether the fluent is true or false at (over) the given time. One immediate problem is that we have already defined fluents to be predicates. First order logic does not allow us to use predicates and functions as variables. Fortunately, the logical device of reification can provide a successful solution to this problem.

In non-reified formulations of situation calculus, fluents correspond to predicates that take one or more arguments. A fluent, according to McCarthy and Hayes' definition [McH69], is " a function whose domain is the space of situations (we will introduce notion *situations* soon). They distinguish various types of fluents. If the range of the function is $\{true, false\}$, then it is called a *propositional fluent*. If its range is the space of situations, then it is called a *situational fluent*. Normally, in the evolutionary versions of the situation calculus, *fluents* usually are *propositional fluents*. Suppose we want to represent the fact that it is raining in situation S_0. One way to do this, using the notion of a propositional fluent above, is to write,

$Raining(S_0)$.

Here, following McCarthy and Hayes, the fluent *Raining* can be thought of as a function whose domain is the space of situations and whose range is $\{true, false\}$. Strictly speaking, it is a predicate whose argument is a situation. An alternative way of expressing the same fact is to write,

$Holds(Raining, S_0)$.

Here, the fluent Raining has been reified. It is awarded the status of object. Consider another example, suppose we want to express the fact that block A is on the table in situation S_0. In non-reified version, it can be written as:

$OnTable(A, S_0)$.

Again *Ontable* is a function whose range is {*true*, *false*}. On the other hand, in a reified language, fluent *OnTable* is treated as an object, and *OnTable*(*A*) would be a fluent term and instead, we write

$$Holds(OnTable(A), S_0).$$

In what follows, we will adopt the reified expression for fluents and other similar terms. Specially, in temporal reified logics [Sho87b, MaK97], propositional terms are related to times or other propositional terms through an additional sort of "truth predicate", such as *Holds* (or *True*), *Causes* etc. For example, one may use *Holds*(*OnTable*(*A*), *T*) to represent the assertion "Block *A* is on the table at time *T*".

Compared with some non-reified approaches such as the method of temporal arguments, it is argued that reified logics have the disadvantage of being more complex in expressing assertions about some given object with respect to different times e.g. an assertion such as "the President of 1962 died in 1963" [BTK91]. However, as summarized by Ma and Knight [MaK97], from the point of view of expressiveness, since reified temporal logics accord a special status to time and allow one to predicate and quantify over propositional terms, they are more expressive for classifying different types of temporal occurrence and representing both non-temporal and temporal aspects of causal relationships. For instance, Bacchus *et al* [BTK91] and Vila [Vil94] claim that while it is difficult or even impossible to express assertions such as "effects cannot precede their causes" in a non-reified logic like that of Bacchus *et al* [BTK91], one can easily express such a statement in a reified logic. A formal demonstration of this claim is provided in chapter 6 of this thesis. As further evidence for the convenience of using reified fluents, notice that several approaches to dealing with the frame problem appeal to the minimisation of the predicate *Ab* (details can be seen in chapter 4). For instance, Baker's solution to the frame problem [Bak91] uses a circumscription policy in which the predicate *Ab* is circumscribed with the interpretation of the function symbol *Result* allowed to vary. Such a policy cannot be expressed without reifying fluents.

2.2.2 States

A fluent is a notion relative to some individual facts. For example, in the blocks world, "Block *A* is on the table" and "Block *A* is on block *B*" are two fluents. There is no information about the relationships between these two fluents. Also, there is no information about whether they can be both true or not at the same time. At a specified time, without further information, one can not make any definitive conclusion about these two fluents. Therefore, to describe a status of the world, we need to use fluents together with the information about the relations among fluents such as whether some of the considered fluents can be true concurrently at a given time or not. For instance, "Block *A* is on the table" together with the fluent "Block *B* is on the table" describe a status of the two blocks world. But fluent "Block *A* is on the table" together with the fluent "Block *A* is on block *B*" can not compose a state in the real world. Therefore, to represent the relations between fluents and describe the world in terms of specifying the truth values of a set of fluents, the notion of states is employed.

In literature, the word "state" is used in various senses by different researchers. For instance, McDermott defines a *state* as "an instantaneous snapshot of the universe" [McD82]. This definition is very close to McCarthy and Hayes' definition of situations. In [McH69], McCarthy and Hayes define a *situation* as "the complete state of the universe at an instant of time", which "can be thought of as snapshot of the world" [Sha97]. Sometimes, the word "state" is even used interchangeably with the word "situation", e.g., Genesereth and Nilsson define that "a *state*, or *situation*, is a snapshot of the world at a given point in time" [GeN87].

According to Lin and Shoham's idea of epistemological completion [LiS95], which is presented based on the conventional situation calculus [McH69], a state can be defined as follows. Let $\underline{S}$ be a fixed set of fluent constants, which includes all the fluents in which we are interested. A set *Sl* of fluents is a state of the situation *Sit* (with respect of $\underline{S}$) if

$$Sl = \{ f \mid Holds(f, Sit), f \in \underline{S} \} \cup \{ \text{not}(f) \mid \neg Holds(f, Sit), f \in \underline{S} \}$$

where *Holds*(*f*, *sit*) asserts that fluent *f* holds in situation *sit*.

Therefore, if *Sl* is a state of situation *Sit*, then for any fluent $f \in \underline{S}$, either *Holds*(*f*, *Sit*) is true or $\neg$*Holds*(*f*, *S*) is true. Intuitively, states completely characterize situations with respect to the fluents in a given set of fluents, such as $\underline{S}$. In other words, a state defined above completely specifies the truth values of fluents in a situation with respect to $\underline{S}$. For example, in the Two Blocks World, there are in total 4 fluents in which we may be interested.

OnTable(*A*) *OnTable*(*B*)

On(*A*, *B*) *On*(*B*, *A*)

Let $\underline{S}$ include these 4 fluents, and let

$$S_1 = \{OnTable(A), OnTable(B)\} \cup \{\text{not}(On(A, B)), \text{not}(On(B, A))\}$$

$$S_2 = \{On(A, B), OnTable(B)\} \cup \{\text{not}(On(B, A)), \text{not}(OnTable(A))\}$$

$$S_3 = \{On(A, B), OnTable(A)\} \cup \{\text{not}(On(B, A)), \text{not}(OnTable(B))\}$$

be three sets of fluents with respect to the set $\underline{S}$. In what follows, for simplicity, we only list the positive fluents. As default their negatives are also considered. If it is necessary, we can make the negatives mentioned. Therefore, the above three sets can be expressed as:

$$S_1 = \{OnTable(A), OnTable(B)\}$$

$$S_2 = \{On(A, B), OnTable(B)\}$$

$$S_3 = \{On(A, B), OnTable(A)\}$$

It is easy to see that for some given situation, sets S_1 or S_2 may contributes a state with respect to $\underline{S}$, since all the fluents in S_1 (or S_2) may hold in a given situation. But S_3 can not, because fluents *On*(A, B) and *OnTable*(A) can not hold true in the same situation (at the same time). Also, in any given situation, there should be one and only one state that can hold true with respect to a fixed set of fluents. This definition has some advantages, as Lin and Shoham pointed out that this definition captured an epistemological complete representation. Also, in actual applications, it is most convenient to talk about whether a description of a world is complete with respect to a set of fluents in which we are interested. However, there are some limitations regarding to this definition. For instance, since it associates a state with a given set of fluents, it is difficult to cope with incomplete information.

The other way is defining a state simply as a set of fluents without any constraints [Sha95]. Only the truth values of fluents in this set are specified, others are treated uncertain. The reason for this treatment is simple as well. Since a world is too large for complete description, it is impossible to completely describe a status of a world. What one can do is only to give facts (fluents) about a relative part of the world. Without having information other than that about this known relative part, one can not make further conclusion. For example, in the above two Blocks world, all the three sets of fluents, S_1, S_2 and S_3 are states with respect to this definition. One of the differences between these two definitions is that S_3 can not contribute any state with respect to the former definition, but it can compose one according to this definition. Therefore, for this definition, a question is raised: Is it possible for the two fluents in S_3 to both hold true at the same time? This is a question about the notion of consistency in states. The other difference between these two definitions is that with respect to the latter any subset (including the trivial one - null) of the four fluents in the two Blocks world can compose a state. This makes it possible to cope with incomplete information. For example, Let S_4 be a state which includes only one fluent, *OnTable*(A). Suppose this fluent holds true at time t, that is:

True(*OnTable*(A), t).

Where *True*(f, t) represents that fluent f holds true with respect to time t. Then we know the information about the position of block A. However, in the absence of further information about block *B*, our conclusion does not include any statement about block, *B*. If we add the knowledge that fluent *OnTable*(*B*) holds true at time *t* as well, that is

$$True(OnTable(A), t) \wedge True(OnTable(B), t),$$

then our conclusion may include the information about the positions of both block *A* and *B*. The main advantage of the definition is that it can handle incomplete information. However, according to this definition, with respect to a given time, there may exist more than one state. In chapter 6, for the purpose of developing a formalism to reason about action and change, we are going to define states in a different way, which gets benefits from both of the above definitions.

As shown above, the notion of fluents is independent of time. It claims a proposition, but does not tell when the proposition holds true or false. For example, "Block *A* is on table" is a fluent. However, given a time *t*, whether this fluent holds true over *t* is unknown. The notion of states inherits this property of fluents. From this point of view, the notions of *states* and *situations* have to be separated.

2.2.3 Situations

A situation differs from a state in the way that it has a strong temporal aspect. In the original version of situation calculus, McCarthy and Hayes define a situation as "the complete state of the universe at an instant of time". Following the discussion about states, normally it is impossible to collect all the information to completely describe a situation. This means one never knows a situation completely-- instead, one only knows some facts about a situation. In fact, we only need to deal with the information in which we are interested. The basic mechanism used in situation calculus to define a new situation is the *Result* function,

$$sit' = Result(a, sit)$$

In this formula *sit* is a situation, *a* is an action, and *sit'* is the result situation of performing action a in situation s.

There are different ways to interpret the idea of a situation in the situation calculus. One of them is that a situation can be defined by the set of fluents that hold in it. According to this interpretation, there is an axiom as follows, which is called *situation-state axiom*, since some researchers reserve the term "state" for a set of fluents [Sha97].

$$sit_1 = sit_2 \Leftrightarrow \forall f(Holds(f, sit_1) \Leftrightarrow Holds(f, sit_2))$$

The other one is thinking of a situation as a unique node in the tree of situations defined by the *Result* function. According to this interpretation, the following axioms, which is called *axioms of arboreality*, can be written, since they insist that the space of situations is tree-like:

$$Result(a_1, sit_1) = Result(a_2, sit_2) \Rightarrow a_1 = a_2 \wedge sit_1 = sit_2$$

$$Sit_0 \neq Result(a, sit).$$

The above two interpretations are not inconsistent with each other. But the axioms of arboreality for the second interpretation do rule out certain formulae that are compatible with the first interpretation, such as,

(Tog) $Result(Toggle, Result(Toggle, sit)) = sit.$

This formulae's intended meaning is that toggling twice has no atemporal effect. The inclusion of any such formula introduces a cycle to the structure which the *Result* function superimposes on the space of situations, making it a graph rather than a tree.

The situation calculus carries no inherent commitment to time. Time can be introduced as a fluent. In McCarthy and Hayes' original paper about situation calculus [McH69], fluent $Time(sit)$ is used to associate a time with the situation. In reified logic, we can use $Holds(Time(t), sit)$ to denote that the time is t in situation sit.

However, the two interpretations of situations lead to two alternative ways in which this can be done.

In the second interpretation, the axioms of arboreality arise from the implicit assumption that the *Result* function is essentially a temporal successor function. The introduction of time as a fluent obviously would be incompatible with a formula like (Tog), but such formulae are ruled out by the axioms of arboreality anyway.

The first interpretation also agrees to the introduction of time as a fluent. However, if we abandon the axioms of arboreality to allow formulae like (Tog), a formula that seems to make sense only in the context of the situation-state axiom, a different approach is needed. This is one of the motivations for Miller and Shanahan to propose another approach.

To incorporate narratives in the situation calculus, some of the researchers have extended the original situation calculus by means of introducing a time line into it [PiR93, 95, MiS94]. Pinto and Reiter associate a time point with each situation, while Miller and Shanahan associate a situation with each time point in the time line. For instance, Pinto and Reiter define a situation which has a starting time point and an ending time point, and the ending time point of a situation *sit* is axiomatised as identical to the starting time of the new situation *Result*(*a*, *sit*), that is the time point at which action *a* occurs. During the time span of a situation no fluents change truth values. They use *Starts*(*sit*, t_1) to represents that the situation *sit* starts at time t_1. In Miller and Shanahan's formalism, to make their approach to narratives work, a function *State* is introduced. The term *State*(*t*) denotes the situation at time *t*. This kind of approach does not separate a state from a situation clearly, although it successfully deals with narratives (See next chapter for the comparison of these two approaches). However, their treatment of linking a situation with a time suggests a way to distinguish a state from a situation more explicitly by means of further combining a state with a time together to represent a situation. In chapter 6, this idea will be discussed in detail.

2.3 Actions and Events

Actions and events are so inextricably bound up with change that we can not seriously exclude them from a temporal reasoning formalism. The differences between events and actions have not yet been fully clarified. In this section, we are going to introduce the basic features for each notion. In chapter 6, we will clarify them more formally.

2.3.1 Actions

Whereas the intuition behind the notion of state (situation) is persistence, the intuition behind the notion of action is change. In common sense, the world persists in one state until an action is performed to change it to a new state. In the formalism of situation calculus, an action can be seen as a function from one situation to another, and can be described by a set of preconditions on the initial situation and a set of effects that will hold in the final situation.

Compared with states and situations, actions have some additional characteristics, such as effects, preconditions etc. Whenever we want to talk about an action, first of all, we have to examine whether it can be successfully performed or not. This means we need to check the precondition of the action. In most cases, the execution of a given action may be constrained by some precondition that must hold. For example, in the Blocks World, suppose we want to perform the action that moves block A onto block B. To guarantee this action can be successfully performed, the condition that both block A and B are top-clear must be satisfied. As the effect of the successful moving action, block A should be on the top of block B.

In addition, sometimes more than one action may perform at the same time. These actions are called concurrent actions. For example, John wants to go for a walk, and is trying to open the door; Mary wants him to stay home, and is trying to close the door. In this case, there are two actions that occur concurrently. One is opening the door, which is performed by John. The other is closing the door, which is performed by Mary. To express this kind of actions, we have to separate the notion of actions to two different notions: primitive or atomic action and global action. A global action may be conceived as the net result of the operation of several atomic actions. Practically, whether two or more actions can be executed concurrently, and what their joint effects would be, depend on how they interact. There are a variety of concurrent interactions, e.g. *Independent*, *Cancellation*, *Synergy* and *Conflicting*.

- **Independent** In this case, the effects of a global action are the aggregation of its subactions. The effects of a subaction can be simply added to the set of the effects of the other subactions. For example, Mary and John are lifting a desk. Mary is lifting the left-side of the desk, while John is lifting the right-side. The effect of the global action is that the desk is in the air. This effect is the aggregation of left-side in air and right-side in air, which are effects of performing subactions by Mary and John respectively.

- **Cancellation** One of the subactions may cancel the effect of another subaction, which may take place if the subaction execute by itself. For instance, while Mary tries to open a door by performing a *Push* action, simultaneously, John pushes on the door the other way. Then the door will not be open as expected by Mary. Thus, we say that the two *Push* actions, performing in the opposite directions, cancelled each other out. The effect of the global action is that the door remains closed.

- **Synergy** Subactions may have synergistic effects, i.e., two subactions, when performed concurrently, will cause effects that would not be caused by any of the subactions performed isolation. For example, consider a wristwatch that is controlled by two buttons A and B. Pressing A changes the mode of the display, while B shows the alarm time setting. Pressing A and B simultaneously, however, turns the alarm on or off. The effect of performing the two actions simultaneously has no causal relation to the effects of the actions performed in isolation. Another example is the soup example. Whenever Mary tries to lift the bowl with one hand, she spills the soup. When she uses both hands, she does not spill the soup. The effect of the global action is that the bowl is lifted by Mary without any spilling.

- **Conflicting** One of the subactions may conflict with another subaction. For example, John wants to go for a walk, and is trying to open the door; Mary wants him to stay home, and is trying to close the door. There are two subactions: opening the door and closing the door. Intuitively, since we don't know whose strength is bigger, we could not deduce any definite conclusion. So we'd better leave the result of performing these two subactions undecided.

So far, much work has been done to deal with the concurrent actions, such as the work of Gelfond, Lifschitz and Robinov [GLR91], the work of Lin and Shoham [LiS92], the work of Pinto [PiR95] and the work of Miller and Shanahan [MiS94]. For example, Lin and Shoham propose an approach to represent concurrent actions in situation calculus [LiS92]. They introduce the notions of *global actions*, *primitive actions*, and the binary predicate *In*. A global action is a set of primitive actions, and *In* expresses the membership relation between global actions and primitive actions. When a global action is performed in a situation, all of the primitive actions in it are assumed to be performed simultaneously.

In the programming language community, several approaches have been developed based on modal logics, which incorporate models of concurrent computation [BKP86, Sho90b, FiB91 etc.]. For example, the Concurrent METATEM is a language based upon the direct execution of temporal formulae [Fis93]. It consists of two distinct aspects: an execution mechanism for temporal formulae in a particular form; and an operational model that treats single executable temporal logic programs as asynchronously executing objects in a concurrent object-based system. The detailed discussion about this language and its relevant work can be found in [Fis94].

Although, approaches developed in common sense reasoning and programming language communities are normally for different purposes, there must be some techniques which can be used in both. Also the exchange of ideas between these two can benefit both.

2.3.2 Events

As mentioned above, an action can be seen as a mapping that maps a situation to another situation. With respect to this explanation, actions can be seen without time structure. Consider the usual emphasis in studies based on McCarthy and Hayes' situation calculus [McH69, LiS92, Lif86, 90, PiR93, 95, MiS94 etc.]. In these systems, an action like "moving A on to B" is reasoned about in terms of a mapping that maps an initial situation, S_0 to another situation: $Result(Move, S_0)$. The axioms of the calculus deal entirely with the different facts which are true or false in the initial situation S_0 and the result situation $Result(Move, S_0)$. There is no mention of the infinite number of states occurring during the move. In this case, actions just bring about fact changes, which is just a list of two situations, i.e., $\{S_0, Result(Move, S_0)\}$. How long they took to reach the result situation is not describable. Further, it is meaningless in those formalisms to ask what happens during the execution of an action.

In the literature, many researchers use events to represent the entity that can occur and cause some change with respect to time [e.g., McD82, All84, KoS86, Sho88, AlF94, MiS94, Sha95 etc.]. As McDermott shows, events are more difficult to handle than fluents. Unlike fluents, the defining features of an event are the changes in facts that the event brings about. Also a deeper problem is that many events are simply not fact changes. Events take time: they can take either time points or time intervals. For example, “John arrived at the train station” is an event with respect to a time point. “John wrote a letter in an hour” is another event with respect to a time interval. In this case, one may raise some questions, such as what happens in the middle of a fact change? Consider the sentence "John ran around the track 3 times." The fact change that occurs is that John is more tired. The amount of fatigue is not terribly different from the amount ensuing on running around 4 times. In [All84], the predicate *Occurs* is introduced to represent the occurrences of events, which takes an event and a time as its arguments and is true only if the event happened over time t, and if time t is an interval, then there is no proper part of t over which the event happened. For example, suppose $ArriveAt(John)$ denotes *John arrived at the train station* and P is a time point, then the occurrence of the event that "John arrived at the train station at time P" can be expressed as:

$$Occurs(ArriveAt(John), P).$$

Noticing that states describe aspects that do not change and events describe change, the most distinction between states and events exists in their relation to time. On one hand, when a state holds over an interval i, one can conclude that the state also holds over all subintervals of i. For example, if a ball is red during the entire day, then it is also red during the morning and afternoon of that day. This property is termed homogeneity in the temporal logic literature. Events, on the other hand, generally have the opposite property and are anti-homogeneous: If an event occurs over an interval i, then it doesn't occur over any subinterval of i, as it would not yet be completed. For example, if a ball dropped from the table to the floor over interval i, then over any subinterval of i, it would just be somewhere in the air between the table and the floor. Thus for any event e, and times t and t', we have the axiom

$$Occurs(e, t) \wedge In(t', t) \Rightarrow \neg Occurs(e, t')$$

where In is a predicate that denote the relationship between two times, in which one time is wholly contained in another.

An additional aspect of events is their hierarchical nature. Consider the event of going to work in the morning. This event contains a number of other simpler events, e.g., getting out of bed, taking a shower, having breakfast etc. Each of these may also contain further, more basic events. Finally, similar to actions, some events may occur concurrently. For example, John wrote a letter over time T_1 while Mary cooked the dinner over the same time. There are two events in this case: John wrote a letter over time T_1 and Mary cooked the dinner over time T_1. The ability to construct structures of this form and reason about them is also an important requirement.

2.4 Causal relations between events and their effects

Fluents, actions and events are central issues in temporal reasoning. To reason about change, the causation of the change is crucial. The fact that people freely speak of A

causing *B* is so common in everyday life such that it has convinced many researchers that modeling causation is a necessity. Drawing conclusions about dynamically changing worlds is grounded on formal specifications of what effects are caused by the execution of a particular action/event. Temporal ambiguity arises when multiple sequences of situations are consistent with the description presented. For example, leaving an unlocked car with the keys in the ignition raises the question of whether it will be there in an hour, particularly in the presence of known car thieves. The space of possible interpretations, then, is the set of situation sequences consistent with the problem description. Temporal reasoning provides an implicit preference over these sequences of situations in the form of *causation*: we prefer sequences of situations in which one situation leads causally to the next, rather than sequences in which one situation follows another at random and without causal connection.

In addition to causation, temporal relationship between events and their effects is another important issue in temporal reasoning. Some effects become true at the end of the event and remain true for some time after the events. For example, in the blocks world, as soon as action "moving a block from the top of another block onto the table" completes, the block being moved should be on the table (immediately), and it should remain on the table for some time after the action is completed. In some cases, there may be a time delay between an action/event and its effect(s). For instance, if a pedestrian presses the button at the crosswalk, the pedestrian crossing light will be caused to turn to yellow from red. It is known that the change (from red to yellow) is caused by the execution of pressing. Additionally, there is a delay time, e.g., 25 seconds, between the event and its effect. In other cases, effects only hold while the event is in progress. For example, consider the flashlight with a button for flashing it. The light is on only when the button is pressed down. Also there are some other cases, e.g., effects might start to hold true sometime after the beginning of the event and to stop being true before the end of the event.

In system/control theory there is a principle normally called the "*causality principle*" which basically says that "actions cannot affect the past". If a model of a dynamic system does not comply with this principle, it's considered "faulty" [Gef98]. Meanwhile, in some philosophical accounts of causation [Sup70], there is a similar property of causality that says: causes precede their effects. In temporal reasoning about actions and change, we have a similar "*causality principle*", that is the most general temporal constraint: *the beginning of the effects can not precede the beginning of the cause* [McD82, All84, Sho88, TeT95, KPM98]. Theoretically, there are in total 8 possible qualitative temporal relationships between actions/events and their effects, which can be briefly classified as follows:

- Case 1 -- the effect becomes true after the beginning of its causal event, and the end of the effect coincides with the end of its causal event;
- Case 2 -- the effect becomes true at the same time as the beginning of its causal event, and only holds true while the event is in progress;
- Case 3 -- the effect becomes true at the same time as the beginning of its causal event, and remains true for some time after the event;
- Case 4 -- the effect becomes true after the beginning of its causal event, and stops being true before the end of its causal event;
- Case 5 -- the effect becomes true at the same time as the beginning of its causal event, and stops true before the end of its causal event;
- Case 6 -- the effect becomes true immediately after the end of the event and remains true for some time;
- Case 7 -- the effect becomes true after the beginning of its causal event, and remains true for some time after the event;
- Case 8 -- the effect becomes true some time after the end of its causal event, and remains true for some time.

In most existing formalisms for representing causal relationships between events and their effects, the effect of an event is represented by the result immediately after the occurrence of the event. Expression of flexible temporal relationships between events and their effects, which is in fact quite interesting and complicated, has been mostly neglected. This issue will be discussed in detail in chapter 6.

CHAPTER 3
PREVIOUS WORK

Since the introduction of McCarthy and Hayes' situation calculus, which is one of the most influential formalisms for representing change in AI, many temporal approaches have been proposed to address the problem of representing action and change in a natural way by enriching the temporal ontology. These systems are similar in many respects, but there are subtle differences in terminology and basic theory that derive from the differences in approach. In this chapter we take a close look at a number of formalisms that have proved popular within the AI community.

3.1 Situation calculus

Situation calculus (SC) is an aspect of the logic approach to AI. It was first discussed in [McC63], but was not widely studied (did not become influential) until [McH69] was published. The aim of the situation calculus is that from facts about situations and general laws about the effects of actions and events, it is possible to infer something about future situations.

Formally, A language of the *situation calculus* can be defined as: a language of many-sorted first-order predicate calculus with equality, which includes,

- A sort for *situations*,
- A sort for *fluents*,
- A sort for *actions*,
- A function *Result* from actions and situations to situations,
- A predicate *Holds* whose arguments are a fluent and a situation.

The basic notions in situation calculus include *situation*, *fluent*, and *action*. A situation is a snapshot of the world at some instant. Situations are rich objects in that it is not possible to completely describe a situation, only to say some things about it. A fluent is a function whose domain is the space of situations. If the range of the function is (*true*, *false*), then it is called a *propositional fluent*, or simply a *fluent*. For example, $Raining(S_0)$ is a fluent, which says that it is raining in situation S_0. As mentioned in previous chapter, this is an unreified representation, and we can express the same fact in a reified way:

$$Holds(Raining, S_0).$$

In what follows, I will adopt the reified representation for the discussion.

There is a key function in situation calculus, *Result*, which takes as its arguments an *action* and a *situation* and returns the new situation resulting from applying the action in the previous situation.

Let's use a simple example illustrate the general idea of situation calculus. Consider a variation of the Blocks World in which there are just two blocks. Each block can be somewhere on the table or on top of exactly one other block. Each block can have at most one other block immediately on top of it. Different states of this world correspond to different configurations of blocks. In this example, we have some fluents such as:

$$On(A, B)$$
$$Table(B)$$
$$Clear(A).$$

These fluents say that block A is on block B, block B is on the table and there is no block on block A respectively. Action $Move(x, y)$ means that block x is moved onto block y. Normally, there are some preconditions for actions. That is the execution of a given action may be constrained by some pre-conditions that must hold. For instance, if we want to perform the action $Move(A, B)$, the precondition $\{Clear(A), Clear(B)\}$ must hold. Suppose in original situation S_0, we have:

$Table(A)$

$Table(B)$

$Clear(A)$

$Clear(B)$.

Then action $Move(A, B)$ is performed. What will happen as the result of the execution of action $Move(A, B)$? According to the commonsense knowledge, for any two blocks x and block y, if both of them are clear, and action $Move(x, y)$ is performed, then in the result situation block x should be on the top of block y. This can be written as:

$$Holds(Clear(x), s) \wedge Holds(Clear(y), s) \wedge Move(x, y)$$
$$\Rightarrow Holds(On(x, y), Result(Move(x, y), s)$$

By this axiom and the knowledge about situation S_0, we obtain

$$Holds(Clear(A), S_0) \wedge Holds(Clear(B), S_0) \wedge Move(A, B)$$
$$\Rightarrow Holds(On(A, B), Result(Move(A, B), S_0))$$

Then we reach the expected conclusion that in situation $Result(Move(A, B), S_0)$, block A is on block B.

McCarthy and Hayes' primary interest is in planning. They treat change through functions from situations to situations, representing actions. This treatment allows us to consider the construction of plans as the development of a series of situation-changing actions that proceed from the initial situation to one in which the desired final properties hold for the system.

Although the original situation calculus is now more than three decades old, it has been studied and used as the basis for a number of recent reasoning formalism [Lif87, Bak91, LiR94, MiS94, PiR93, PiR95]. For the past several years, the Cognitive Robotics Group at the University of Toronto has been exploring the feasibility of the situation calculus as a theoretical and computational foundation for modelling autonomous agents dwelling in dynamic environments. Their purpose is to enrich the situation calculus to make it to be well suited to the general problem of providing a formal and computational account of complex dynamic domains and agent behaviours. Some modest success has been achieved in the research [Rei91, LRL97, LLR99, Lin96, PiR95].

As discussed in [GLR91, AlF94, Sha97], since the original situation calculus is a point-based temporal logic with branching time model, there are five most frequently alleged drawbacks of the situation calculus.

- It cannot handle actions whose order of occurrence is unknown.
- It cannot handle concurrent actions.
- It cannot handle actions with duration.
- It cannot handle actions with delayed effects.
- It cannot handle continuous change.
- It cannot handle external events.

To overcome those drawbacks of the original situation calculus, many researchers have proposed extensions by enriching the primitive objects and the temporal ontology. In Gelfond, Lifschitz and Robinov's paper, "What are the limitations of the situation calculus?" [GLR91], they challenge most of those issues. They suggest extensions to the original situation calculus that enable the formalism to represent some of the phenomena mentioned above. Their suggestions are taken up and amplified by various authors [LiS95, MiS94, PiR95, Sha97]. Most of their suggestions are acceptable and lead to some successful extensions. Lin and Shoham tackled the concurrent issue by enriching primitive objects, actions, in their extension. Based on the argument that "While situations are 'rich' objects, as manifested by the various fluents that are true and false in them, actions are 'poor', primitive objects." [LiS95]. They introduce the notions of *global actions*, *primitive actions*, and the binary predicate *In*. *In*(a, g) means that the primitive action a is an element of the global action g. They successfully integrate their approach to concurrent actions with their attempt at the frame problem. However, they did not concern actions with duration.

Pinto and Reiter, and Miller and Shanahan's approaches [PiR95, MiS94] provide a richer temporal ontology. In [PiR95], Pinto and Reiter extend the ontology of the situation calculus for the representation of time and event occurrences. They do this by defining a time line corresponding to a sequence of situations (called *actual* situations) beginning with the *initial* situation. Actual situations are totally ordered and the actions that lead to different actual situations are said to have *occurred*. In their formalism, they endow a branching structure of time with a time line and identify one path through the tree of situations as the actual course of events. Pinto and Reiter define a situation which has a starting time and an ending time, and the ending time of a situation *sit* is axiomatised as identical to the starting time of the new situation *Result*(a, *sit*), that is the time point at which action a occurs. During the time span of a situation no fluents change truth values. This extension enriches the ontology of the situation calculus to provide for the representation of time and event occurrences. However, their treatment of time may lead to the so-called Divided Instant Problem (DIP) [Van83, Vil94], i.e., the problem of specifying whether time spans of situations are closed or open at their starting/ending points.

In the extension of Miller and Shanahan [MiS94], they focus their attention on the incomplete narrative information. The objective of their paper is to provide an approach to bridge the gap between the situation calculus and the narrative-based approach, such as event calculus. Comparing with the method by Pinto and Reiter, they introduce a narrative time line, such as reals or the naturals, while each situation is associated with a time point in the narrative time line by the introduction of a function, *State*. The term *State*(t) denotes the situation at time point t. This is one of the main differences between the approaches of Pinto and Reiter [PiR95] and Miller and Shanahan [MiS94]. In the formalism of Miller and Shanahan, the frame problem is overcome by extending the idea of Baker's solution. Also, it is able to cope with actions with duration and overlapping actions in these extensions. In order to specify that an action occurs over a real time interval (t_1, t_2), they use a three-argument version of the predicate *Happens*. Formula *Happens*(a, t_1, t_2) means that action a starts to occur at time point t_1 and finishes occurring at time point t_2. By this interpretation of the predicate *Happens*, it is not clear how to express the occurrence of an action during a real time interval, which includes the right end point. Since their treatment of actions with duration follows the idea of that in Gelfond *et al* [GLR91], the DIP and the problem of the uniqueness of the actual time with respect to a given duration may arise. In chapter 6, a detailed discussion about these issues will be given.

There is a clear correspondence between these two extensions. In fact, the mapping Sit of Miller and Shanahan's can be equated with the actual situations of Pinto and Reiter's. Furthermore, Shanahan has shown that, under suitable conditions, both extensions yield the same results [Sha97].

In addition, some sketchy accounts of how to represent continuous change in the situation calculus were published in the early Nineties [Sch90, GLR91]. Recently, some new formalisms based on situation calculus have been proposed in order to represent continuous change [Pin94, Rei96, Mil96]. In all those representations, time line plays an important role. However, Reiter [Rei96] shows that it is possible to do without a distinguished narrative time line and still represent continuous change in the situation calculus.

In most of these extensions of the situation calculus, there is a restriction on causal relations, that is *causes strictly precede their effects* or similarly, *there is no change during the performance of actions*. This omits some cases in which the causal action and its effects occur simultaneously, or there may be a delay time between the end of an event and its effects.

3.2 McDermott's temporal logic

In order to deal with the problem of reasoning with and about time and temporal primitives in the context of planning, McDermott [McD82] proposes a first-order temporal logic to provide a versatile "common-sense" model for temporal reasoning, with a similar flavour to Hayes' naive physics [Hay78]. In this thesis, this logic is stated as McDermott's Temporal Logic (MTL).

Like McCarthy and Hayes [McH69, Hay78], McDermott adopts states as the primitive temporal element. He proposes several crucial axioms: Every state has a time of occurrence, $d(s)$, a real number called its date. Time is assumed to be a continuum, with an infinite number of states between any two distinct states, where states are partially ordered by the "*no later than*" order relation "$\leq$"; and the future (not the past) is branching, that is, there are many possible futures branching forward in time from the present. Each single branch, called a "*Chronicle*", consists of a connected series of states and is isomorphic to the real time line.

There are two key ideas in McDermott's logic. One is the "openness" of the future. This is that more than one thing can happen starting at a given instant. McDermott introduces a time structure with branching to treat this idea. The other idea is the continuity of time. This idea is based on the fact that many things do not happen discontinuously.

As pointed out by McDermott, his approach is not equivalent to that of McCarthy and Hayes [McH69]. The main difference that he claims is in the treatment of actions and change. McDermott argues that change is often a continuous process and the continuous nature is fundamental to many processes of change. Thus, the treatment that the Situation Calculus offers, in which change is instantaneous (the transition from one state to the next) is inadequate. McDermott also argues that events brought about by actions are not always best characterised by a transition between states, or fact changes; he cites as examples such things as running around a track three times, eating a gourmet meal and so on, where the discrete changes brought about by the action are, at best, only partially relevant to the action. This is the main reason McDermott made a distinction between facts and events. In his logic, a fact is defined as a set of propositions that hold true at some indicated time. An event is defined as a set of intervals over which one occurrence of the event takes place.

However, as Long pointed out [Lon89], McDermott's treatment of events (or facts) as sets of intervals will face the following three problems:

1. The Dividing Instants Problem. Intervals are denoted by the states marking their end points—the question of whether the end points are included in the intervals or

not remains undetermined, with a rather inconclusive argument favouring closed intervals [Van83, Vil94].

2. The Multimingling Problem. An event may consist of an infinite number of intervals within a finitely bound super-interval [Ham71, Gal96]. McDermott employs an axiom to ensure that no state can change infinitely often during a finite interval.

3. In order to define an event in the way proposed by McDermott, one must begin by knowing all the intervals over which it occurs, in every chronicle. This is important, since it is only by considering the event over every chronicle that one is able to discard irrelevant features of an event. In fact, it would appear that the only way to ensure that the intervals chosen actually do characterise the required event is to have the event defined in advance. Thus, this construction of events does not allow an internal characterisation. Hence, it neither corresponds with intuition nor offers any explanation of how events might be constructed initially.

Having developed a logic of time, McDermott examines three major problems that temporal reasoners must face: reasoning about causality and mechanism, reasoning about continuous change and planning actions. The examination can be briefly described as follows:

- The treatment of causality. McDermott asserts that events can cause two kinds of things: other events or facts. He introduces two basic predicates to deal with the causality, predicate *ECAUSE* and *PCAUSE*. For instance, formula *ECAUSE*(p, e_1, e_2, rf, i) says that event e_1 is always followed by event e_2 after a delay in the interval i, unless p becomes false during interval i. Argument rf controls the beginning of the delay. The delay interval i starts from the beginning of e_1, if $rf = 0$; from the end of e_1, if $rf = 1$.

- The treatment of persistence. Naturally, the truth value of a fact in the real world is supposed to keep unchanged until some action happens to force it to change. Many facts remain true for long enough to be depended on. For example, as the result of the event "a boulder fell down to the bottom of a mountain", that boulder will probably stay at the bottom of the mountain for years. Based on this view, he introduces a predicate *persistence* to deal with the persistence of facts. Formula *persistence*(s, p, r) says that fact p persists from state s with a lifetime r. The reason that McDermott does so is to take into account the fact that in many cases, we simply lose information about a system for moments too far from our last observation. However, the argument r seems not necessary here. According to the explanation above, some problems may arise. For instance, normally, we don't know how long the lifetime should be with respect to a given fact and a give state.

- The treatment of continuous change. McDermott has paid a great deal of attention to the subject of continuous change. He claims that a system cannot reason about time realistically unless it can reason about continuous change. McDermott uses the term *fluent* to model continuously changing variables. A fluent is a thing whose value changes with time. This term is used in almost exactly the same sense as in situation calculus. Fluents are valuable, because physical quantities may be thought of as fluents. For example, "the temperature in London" is a fluent, which takes on values in temperature space. The changes of the fluent can then be reasoned about. Then McDermott enforces an axiom which states that when a quantity is changing continuously, then during any interval for which the values of the quantity at the end point are known, the quantity must take all intermediate values. McDermott also shows how this property of continuity can be used in reasoning about the behaviour of a tank filling with water.

- The treatment of planning. McDermott notes the importance of choice in planning -- choice of actions. To do this, he introduces a branching time structure. He also provides axioms that ensure that chronicles branch only into the future, although this limits the expressiveness of the logic. In addition, it is arguable whether we need to consider time as branching in order to model possible worlds. In fact, it is possible to conceptualise the world number, or chronicle, as related to the event data, and not to the time. [KnM94]

3.3 Allen's interval calculus

In [All84], Allen develops a formalism for reasoning about actions based on temporal logic for the naive treatment of two major subareas of artificial intelligence: natural language processing and problem solving. His interests include many topics, such as actions, causation, intentions and planning.

Allen's interval calculus (IC) offers a very different approach to modelling time as compared with that of McDermott. Instead of adopting time points (or states which are associated with time points) he takes intervals as the primitive temporal quantity, the natural means of human reference to time. In order to express temporal ordering of these intervals, Allen takes as primitive a set of thirteen (mutually exclusive) basic binary relations between any pair of intervals [All83]: *Equal*, *Before*, *Meets*, *Overlaps*, *Starts*, *Started-by*, *During*, *Contains*, *Finishes*, *Finished-by*, *Overlapped-by*, *Met-by*, *After*. Not only does Allen take intervals as his primitive temporal quantity, but he also specifically excludes time points in claiming that any quantity of time must be subdivisible. This ruling eliminates the possibility of instantaneous events from Allen's treatment. It seems strange that a theory intended to support the expressive power of natural language does not support the expression of instantaneous events, particularly instants such as "now" or the commencement or termination of intervals. Allen's original contention is that nothing can be true at a time point, for a point is not an entity at which things happen or are true [All83].

To characterise the times that some "*instant-like*" events occupy, in a later paper [AlH89], Allen and Hayes define time points as the "*meeting places*" of intervals or as a maximal set, termed "*nest*", of intervals that share a common intersection, at a subsidiary status within the theory; and use the concept of "*moments*", i.e., very short intervals which is non-decomposable, to model some instantaneous events. Except for the assumption that moments have positive length, while points have zero length, another obvious structural difference between points and moments is that moments are treated as primitive objects, and hence can meet other intervals (although they are not allowed to meet other moments), while points are not treated as primitive objects and cannot meet anything [AlH89].

However, as Galton shows in his critical examination of Allen's interval logic, Allen's theory of time is not adequate, as it stands, for reasoning correctly about continuous change [Gal90]. This problem stems from Allen's determination to base his theory on time intervals rather than on time points, either banishing points entirely, or relegating them to a subsidiary status within the theory.

To reason about action and change, Allen introduces a fairly rich ontology of temporal primitives: properties and occurrences. Further, Allen divides the class of occurrences into two subclasses, processes and events. These notions may be distinguished by the way in which they hold or occur in time. A property is described to hold over every subinterval of any interval over which it holds. For example, if I was in London all last week, then I was in London all of last Tuesday. For an event, on the other hand, each occurrence defines a unique interval over which it occurs, and it does not occur over any subinterval of that interval. Processes refer to some activity not involving a culmination or anticipated result, such as the process denoted by the sentence, “I am running”. Processes fall between events and properties. To see this, consider the process “I am running” over interval *i*. Unlike events, this process may also be occurring over subintervals of *i*. Unlike properties, however, it is not the case that the process must be occurring over all subintervals of *i*. For example, if I am running over interval *i*, then I am running over the first half of *i*, however, there may be some subinterval of *i* when I paused for a brief rest.

Allen uses three different predicates to relate elements from his three ontological categories to the times over which they hold or occur. For a property, he uses the predicate $Holds(p, t)$ to state that property *p* holds true throughout the interval *t*; for an events, $Occur(e, t)$ to state that event *e* happens over interval *t* and there is no subinterval of *t* over which event *e* happens; and for a process, $Occurring(p, t)$ state to that process *p* is occurring over interval *t*.

In order to deal with causation, Allen employs one predicate and one function in his theory. Event causality is denoted using the *Ecause* predicate. $Ecause(e_1, t_1, e_2, t_2)$ says that the occurrence of event e_1 at time t_1 causes the occurrence of event e_2 at time t_2.

According to Allen, an action is an occurrence caused in a "certain" way by an agent. So the action causality, termed agentive causality by Allen, is characterised by the function *Acause*(*agent*, *occurrence*), which for any agent and occurrence produces the action of the agent causing the occurrence. As pointed out by Derek Long [Lon89], there is a strange asymmetry in the treatment of these two forms of causation, taking one form to be predicate and the other a function, and Allen offers no reason for this treatment.

3.4 Event Calculus

The event calculus was first introduced by Kowalski and Sergot [KoS86], which is an approach for representing and reasoning about time and events within a logic programming framework. It is based in part on the situation calculus [McH69], but focuses on the concept of event as highlighted in semantic network representations of case semantics. As its name suggests, in this calculus, the notion 'event' is central. Event descriptions imply the existence of time periods for which certain relationships hold. The holding of relationships for periods of time, in turn, implies the holding of relationships at time points. Events start and finish periods of time, during which states are maintained. Events are considered to be after the time periods that they finish and before the time periods that they start, not fully contained within either of these periods.

The event calculus incorporates the idea of a distinguished narrative of events. Event occurrence is described by predicates *Act* and *Time*. In later versions of event calculus, which employed time points instead of time periods and which is normally called simplified event calculus [Sad87, Sha89, Kow92], the *Act* and *Time* predicates are combined into a single predicate *Happens*. *Happens*(a, t) means that an event of type a occurs at time t. The effects of actions are described by predicates *Initiates* and *Terminates*. Formula *Initiates*(a, f, t) presents that an event of type a initiates the property f at time t, and *Terminates*(a, f, t) presents that an event of type a terminates the property f at time t. For instance, suppose the event "Peter moves from room A1 to room A2", denoted by Peter_Move, happens on 1/1/90, denoted by T_1 causes the property f, "Peter is in room A2", to be true starting from time T_l. Then the fact after the occurrence of this event can be described as

$$Initiates(Peter\_Move, f_l, T_l).$$

Events, in the event calculus, are considered to be structureless "points" in time, where "points" is used here only to convey the lack of internal structure. Events start and end periods of time, during which properties are maintained. The event calculus makes use of negation as failure to deal with persistence. Kowaski and Sergot show that this can, in the case of incomplete information about events starting and ending time periods, lead to incorrect results about the equality of different time periods. However, this problem has been bypassed in the later simplified version of event calculus. As mentioned by Shanahan, "Periods are eliminated altogether. A simplified set of axioms bypasses the need for them" [Sha97].

In [Sad87], Sadri illustrated a number of the general characteristics of the event calculus:

(1) Event descriptions can be assimilated in any order, independent of the order in which events actually take place;
(2) Events can act as temporal references and need not be associated with absolute times.
(3) Events can be simultaneous.
(4) Events can be partially ordered.
(5) All updates are additive. The effect of deletion is obtained by adding information about the end of periods.
(6) The event calculus rules are in Horn clause logic augmented by negation as failure. These rules can be run as a logic program in Prolog.
(7) The event calculus allows events to be input with incomplete descriptions.

The event calculus is developed primarily as a means of supporting a temporal database, rather than immediately concerning with the problem of modelling change. Change is assumed to be associated with events and is reflected in the discrete transition from one state to another, punctuated by an intervening event. Shanahan [Sha90] extends the event calculus with the ability to represent continuous change in a manner that entails relatively little modification. He does this principally by the introduction of a new predicates, *Trajectory*(q, t_1, p, t_2) which describes the evolution of a property, p, by giving its precise value at time t_2, when the property (continuous change), q, was initiated at time t_1. This predicate has been revised later as *Trajectory*(f_1, t, f_2, d), which represents that if discrete fluent f_1 is initiated at time t then continuous fluent f_2 holds at time $t+d$, where d is a time duration. This treatment satisfactorily handles the phenomenon of events that are triggered when a continuously varying quantity reaches a threshold value. For example, consider a kitchen sink that's filling with water. When the water reaches the brim, the sink overflows. In other words, an event occurs which is triggered because the water attains a particular level. This event in turn initiates and terminates various fluents.

More recently, some researches have been done for purpose of either examining the variants of the event calculus [SaK95], or studying the difference between event calculus and situation calculus [VDD95, KoS97]. Sadri and Kowalski [SaK95] argue that the original event calculus has the advantages of being more general than the later simplified event calculus, but the disadvantage of being too complex and in some cases erroneous. The simplified event calculus has the advantage of simplicity, but the disadvantage of being too specialised. They also show that, in certain cases of incomplete information about events, both original event calculus and simplified event calculus give incorrect results. In [SaK95], Sadri and Kowalski claim that they present a new variant of event calculus, which combines the generality of original event calculus with the simplicity of the simplified event calculus and avoid the above mentioned incorrect results.

3.5 Shoham's theory of time and causation

Shoham has developed a theory of time and causation (TTC) based on a first order modal logic, for representing and reasoning about temporal information. He claims that "a prerequisite for reasoning about change is a language for representing temporal information" [Sho88]. The logic that Shoham develops has a number of interesting features, including a temporal nonmonotonic reasoning component, which will be discussed in section 4.4.3.

Shoham takes time points as his fundamental temporal primitive objects, which are related by <, the precedence relation. An interval is defined as an ordered pair of points (its begin-point and end-point) such that the first either precedes or is equal to the second (and thus a point P is identified with the interval $<P, P>$). So intervals with zero duration are allowed. As pointed out by Shoham, the reason that the time intervals are defined in term of time points is "any property of intervals can be expressed as a property of their end points, and in fact we will see that the point-based formulations are more concise as well as more intuitive." To associate an atemporal assertion, such as "the house is red," with a time interval, following McDermott's and Allen's lead, he adopts a predicate *True*. Formula $True(t_1, t_2, p)$ states that proposition p holds true over interval $<t_1, t_2>$. For example, the assertion "the house is red during the year 1990" can be expressed as:

True(*1/1/90*, *31/12/90*, *Color*(*House*, *Red*))

To extend Allen's three distinct entities: *properties*, *events*, and *processes*, and McDermott's *facts* and *events*, Shoham provides a new categorisation scheme which includes *facts*, *properties* and *events*. He claims that the assertions "I ran more than two miles" and "I ran less than two miles" do not fit into either of Allen's or McDermott's categorisation schemes. However, within his new categorisation scheme, these assertions can be classified clearly.

Rather than introduce facts, events, properties and so on as separate objects, Shoham starts with the primitive notion of a temporal proposition, and then devises categorisations of proposition types, relying on the relation between the truth of the proposition over one interval and its truth over others. The main advantage of this approach is that one is not compelled to make any distinction when he has no need for it. For example, when it is said that "x causes y", x and y may be facts, events, or something else.

To specify how the truth of the proposition over one interval is related to its truth over other intervals, Shoham adopts the notion of homogeneity: a homogeneous proposition is true of an interval if and only if it is true over all its proper subintervals. In addition, he extends the hierarchical scheme, providing six main classes:

1. Downward-hereditary ($\downarrow p$): a proposition which whenever it holds over an interval, it also holds over all of its subintervals.
2. Upward-hereditary ($\uparrow p$): a proposition which whenever it holds for all proper subintervals of an nonpoint interval, it also holds over the nonpoint interval itself.
3. Liquid ($\updownarrow p$): a proposition that is both upward-hereditary and down-hereditary.
4. Concatenable: a proposition which whenever it holds over two consecutive intervals, it also holds over their union.
5. Gestalt: a proposition that never holds over two intervals one of which properly contains the other.
6. Solid: a proposition that never holds over two properly overlapping intervals.

According to this classification, Liquid proposition types coincide with Allen's properties and McDermott's facts. Allen's and McDermott's events correspond either to gestalt propositions, or to solid ones, or to both.

Shoham extends his logic to handle causality by introducing a modal operator, $\Box$, which he suggests should be read as either "it is known that ..." or "it is believed that ..." depending on the context that they appear in. A causal statement in Shoham's theory is written in the form:

$$\phi \wedge \theta \Rightarrow \Box\varphi$$

where;

1. The latest time point of a formula, ltp, is the latest time point that appears in it;
2. φ is a (positive or negative) atomic base sentence with ltp t_l;
3. ϕ is a conjunction of sentences $\Box\varphi_i$ where φ_i is a (positive or negative) base sentence with ltp t_i such that $t_i < t_l$;
4. θ is a conjunction of sentences $\neg\Box\neg\varphi_j$ where φ_j is a (positive or negative) base sentence with ltp t_j such that $t_j < t_l$.

Here, ϕ is the causing and φ the caused part of a sentence. θ represents the set of background conditions that must be true in order that ϕ causes φ. The idea here is that the background conditions which Shoham refers to as the "causal field" are assumed to be true unless there is explicit evidence to the contrary. This embodies the theory with a kind of nonmonotonicity at the language level. It should be noted that if all of the sentences in θ were required to be false, rather than true, then the same effect could be achieved using negation as failure.

As pointed out by Shoham [Sho88], his formulation can be viewed as a suggestion to generalise McCarthy's approach in three ways:

1. Start with any standard logic, not necessarily first-order predicate logic. For example, Shoham bases his formulations on a standard modal logic.

2. Allow any partial order on interpretations, not only the one implied by a particular circumscription axiom. For example, he suggests a preference criterion that relies on temporal precedence.

3. Shift the emphasis to the semantics, stressing the partial order on models and not the particular way of defining that partial order. The various circumscription axioms, either McCarthy's original ones or Lifschitz's more recent ones, are one way of doing so, and they are more elegant. It remains to be seen whether that

particular way of expressing the preference criterion on models, using a second-order axiom, has additional advantages. In his own formulations he chooses other means of defining preference criteria, such as chronological ignorance.

Shoham's basic temporal logic is an elegant, straightforward formalism with the advantage of increased expressive power via his meta-level temporal classification. However there are still some technical limitations in his theory. Firstly, it is less straightforward to relax the discreteness requirement for time in his logic to deal with the continuous change. Secondly, in both causal theories and inertial theories it was assumed that, intuitively speaking, causes strictly precede their effects. This restricting avoids some problems, but "it may be argued that this is an overkill" [Sho88]. Finally, in Shoham's theory, two types of causation are recognised: potential causation, e.g., "X generally causes Y", and actual causation, e.g., "Some X actually caused some Y" [Sho90]. Shoham gives no explanation for incorporating such distinctions into his theory other than to enable causal statements to be classified. It is worth mentioning here that in order to be able to describe simple planning operators, Bell extended Shoham's causal theories and defined interval theories and extended causal theories [Bel91].

3.6 Allen and Ferguson's interval temporal logic

Allen and Ferguson [AlF94] present a representation of events and actions based on interval temporal logic. The goal of their work is the development of a general representation of actions and events that supports a wide range of reasoning tasks, including planning, explanation, prediction, natural language understanding and commonsense reasoning in general. An approach to the frame problem based on explanation closure is proposed. Some difficult problems in the area of knowledge representation such as external events and simultaneous actions have been discussed in their framework.

In their paper, Allen and Ferguson claim that in order to adequately represent actions and events, one needs an explicit temporal logic, and that approaches with weaker temporal models, such as state spaces (e.g., STRIPS-based approaches) and the situation calculus, either cannot handle the problems with complicated temporal ontology or require such dramatic extensions that one in effect has grafted an explicit temporal logic onto the earlier formalism.

The basic temporal structure used in Allen and Ferguson's logic, namely the interval representation of time was developed by Allen [All83, 84] and discussed in detail in [AlH89]. They introduce a notion named time periods. Intuitively, a time period is the time associated with some event occurring or some property holding in the world. For this reason semi-infinite and infinite periods are not allowed in their time structure. Also there is no beginning or ending of time. The relationships between two periods are defined in terms of the single primitive relation "*Meets*". There are thirteen possible relations between time periods. A period is either a *moment* or an *interval*. A period is a moment if it has no subperiods, an interval if it has subperiods. The temporal structure they assume is a linear model of time.

Allen and Ferguson discuss two important open issues. One is how to deal with temporal durations. For instance, it is not realistic to say that if an agent tries to turn the ignition on the car for any length of time, then the engine will start. If the action is tried for too short a time, the engine probably won't catch. Also, if the action is tried for too long a time, the starting motor will burn out. The other issue is the introduction of probabilistic knowledge. By staying within standard first order logic, for example, one is restricted to saying that an event will definitely occur, or that it might possibly occur.

To reason about change, Allen and Ferguson make a distinction between the notion of actions and the notion of events. Events are described as the way by which agents classify certain useful and relevant patterns of change. As such, there are very few restrictions on what an event might consist of except that it must involve at least one object over some stretch of time, or involve at least one change of state. An action refers to something that a person or robot might do. It is a way of classifying the different sorts of things than an agent can do to affect the world. By performing an action, an agent causes an event to occur, which in turn may also cause other desired events to occur.

Based on the interval time structure and the classified notions of actions and events, it is possible for us to represent some temporal relationships between actions and their effects. Allen and Ferguson claim that the simultaneous effects and delayed effects can be expressed in their formalism. However, no examples for representing such cases are provided. More attention has been paid to the temporal relationships between actions and events. Additionally, as mentioned by Allen and Ferguson, "it needs to be acknowledged that formalising knowledge using the more expressive temporal representation can be difficult. Subtle differences in meaning and interactions between axioms may be more common than in less powerful representations, and more experimentation is needed in building knowledge bases based on our representation" [AlF94]

Since their formalism is within the first-order logic, Allen and Ferguson provide an approach to handle the frame problem based on the explanation closure approach. However, it is arguable that this approach has some limitations (see next chapter for the detailed discussion).

3.7 Discussion

It is difficult to say what the ideal characteristics of a general purpose AI temporal reasoning system should be. Also, it is not easy to evaluate which system is the best. AI has been becoming a far reaching discipline covering everything from natural language processing to automated theorem proving and the temporal requirements of its many component parts are incomparable. For different purposes, the system may focus on different interests. For example, a computational linguist might consider some notion of tense to be of the most importance, whereas this is of little practical use to the qualitative physics researchers. In fact, it seems unlikely that a general system capable of satisfying all of AI's temporal requirements will be built (or even needed) in the foreseeable future. However, within any sub-area of AI, e.g. language processing, automated reasoning, qualitative reasoning or temporal reasoning about actions or change etc. it seems possible or necessary to build a general system for the treatment of general purpose of these sub-areas.

The AI approaches to time are normally different from the program verification and specification approaches. In the former, it is usual to use a first-order formalism, whereas, in the latter, modal approaches are very popular. For example, Barringer, Kuiper and Pnueli developed a modal logic, called ***Temporal*** *Logic of Reals* (TLR), which is stuttering robust, yet possesses a general *next-time* operator. It is a dense time model and interpreted over sequences of sampling points, alternating between persistent and transient sample points. This logic provides a more abstract description of concurrent and reactive systems [BKP86]. Further details about using TLR in verification and specification of systems can be found in [Pnu92, KMP94]

Interval Temporal Logic (ITL) is a modal approach proposed by Moszkowski for specifying and reasoning about computer programs, digital circuits and message-passing systems [MoM84, Mos85], in which an interval is considered to be a (in)finite sequence of states, where a state is a mapping from variables to their values. Moszkowski then introduced a programming language called **TEMPURA** based on ITL [Mos86]. TEMPURA provides an executable framework for developing and experimenting with suitable ITL specifications. Some recent work on system specification and simulation based on ITL and Tempura can be found in [CaZ97, ZCM00, etc].

For the same purpose, Barringer, Fisher *et al* developed a temporal programming language, called **METATEM**, which shares the most advantages with TEMPURA [BFG89]. The temporal logic employed in METATEM can be seen as classical logic extended with various modalities representing temporal aspects of logical formulae. As stated by Fisher and Owens, "The difference between TEMPURA and METATEM is analogous to the difference between PASCAL and PROLOG: the former represents a lower-level approach amenable to efficient execution, the latter represents a higher-level approach more suited to describing symbolic computation." [FiO92]. Although METATEM has a relatively simple and useful model of computation, it suffers from several problems. Having investigated these problems, Fisher extended it to Concurrent METATEM [Fis93], which consists of two distinct aspects: an execution mechanism for temporal formulae in a particular form (basically the METATEM execution mechanism); and an operational model that treats single executable temporal logic programs as asynchronously executing objects (agents) in a concurrent object-based system. Each agent executes its own set of temporal formulae and, in doing so, (effectively) generates an infinite sequence of states. As each agent executes asynchronously, it constructs a separate execution sequence, while a mechanism is provided for communication between separate agents. Further work on Concurrent METATEM and relevant executable modal and temporal logics can be found in [Fis94, 96, Gab91, 96, KeF00 etc.] Although, TEMPURA and METATEM were designed particularly for the specification and verification of systems, such as computer programs, digital circuits and message-passing and reactive systems, it is interesting to investigate its possible applications to the common-sense reasoning area.

In this thesis, our main interest is in temporal reasoning about actions/events and change so we will confine our comments to this more limited, but realistic, sub-area.

The majority of the systems that we have looked at are designed for coping with action and change problems. Here, at least, we can suggest a number of minimum requirements. For the temporal representation, it seems essential that both time points and intervals can be handled. The former are a prerequisite for modelling continuous change while the latter greatly simplify the representation of duration processes and events. For the causal expressiveness, the basic issues, such as the frame problem (persistence), ramification problem and qualification problem etc. must be efficiently handled. The ability to differentiate between states (situations) and actions (events) that change them must also be of importance. This distinction is of particular concern in the application of such formalisms, such as planning, diagnosis etc. The ability to represent complex relationships between actions/events and their effects is necessary if realistic problems are to be solved, although most of the existing systems deliberately choose to treat this relationship as the effects becoming true at the end of the event and remaining true for some time after the event. For example, when I put a book on the desk, this has the effect that the book is on the desk for at least a short period of time after the action is completed.

The systems surveyed in this chapter are quite popular in the community of temporal reasoning. However, most of these systems only fulfil some of the above requirements. The situation calculus, for instance, is a point-based temporal logic. It is difficult to handle actions with duration in it. On the other hand, interval calculus is interval-based. It eliminates the possibility of instantaneous events. Table 2.1 shows the comparison of these systems surveyed in this chapter, where the meanings of the abbreviations are as follows:

CS: Situation Calculus;
MTL: McDermott's Temporal Logic;
IC: Allen's Interval Calculus;
EC: Event Calculus;
TTC: Shoham's Theory of Time and Causation.
ITC: Allen and Ferguson's Interval Temporal Logic

	SC	MTL	IC	EC	TTC	ITC
Intervals			•			•
Points	•	•		•	•	
States		•		•	•	
Situations	•					
Actions	•	•	•			•
Events		•	•	•	•	•
Causation		•			•	•
Concurrency		•		•	•	•
Delayed Effects		•	•			•
Synchronous Effects						•

Table 2.1: Comparison of temporal reasoning systems

In chapters 5 and 6, we will provide a temporal reasoning system that attempts to overcome some of the drawbacks of the existing systems and try to combine some of the benefit from them.

CHAPTER 4 NONMONOTONIC REASONING ABOUT ACTION AND CHANGE

An important aspect of common sense reasoning is the ability to reason about actions and change. For example, imagine a robot responsible for various household chores. It must be able to infer that if it moves the lamp, the shade and cord move, too. But when it tries to move the bookcase, with the lamp on top, the cord will come tight, and the lamp will fall off and break. On the other hand, a napkin, or a piece of paper on top of the bookcase might flutter off, but probably wouldn't cause any harm. Then too, the robot must know where to find things, like the vacuum, the carpet, the closet, or the bookcase. This requires the ability to infer that furniture and appliances stay in the same place unless someone deliberately modifies them. There are effectively an infinite number of such reasoning situations that a household robot might come up against. It is therefore essential to its function that the robot be able to reason about actions and their effects on the world.

There are three classical problems in reasoning about actions and change: *frame problem*, *ramification problem* and *qualification problem*. Since these problems were nominated, they have attracted much interest in the artificial intelligence community and many solutions to them have been proposed [Haa87, Lif87, LiS95, Rei91, Sch90, Sho88, Sha95, 97, StM94, Thi96, 97, McT95, KaM97 etc.]. Some of them are based on monotonic logics, such as explanation closure. However, among them, the most successful and popular methods are based on nonmonotonic logics. In this chapter, our discussion is focused on the nonmonotonic techniques for these problems. We will introduce what are these problems and briefly discuss about each problem first. Then we will present examinations of the existing solutions to the frame problem in detail. For convenience, we employ the situation calculus to represent the problems.

4.1 Frame Problem

The frame problem was first introduced and named by McCarthy and Hays [McH69]. The difficulty is that of indicating and inferring all those things that do not change when actions are performed and time passes. For example, when the lamp is moved from the bookcase to the table, we need to determine that the vacuum cleaner and the carpet remain stationary, and do not change shape and colour. Formalising this notion of *inertia* or *persistence* has turned out to be a notoriously difficult problem to which numerous solutions have been proposed. In literature, the most recent and generous contribution to the frame problem is Shanahan's book, "*Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*". In his book, Shanahan discusses the various approaches to the frame problem. Also in the book's concluding chapters, he offers his own work on Event Calculus, which Shanahan claims "comes very close to a complete solution to the frame problem".

The original proposal for the situation calculus [McH69] was to use frame axioms, which explicitly stated which properties are not changed by the actions. To write down the frame axioms in the above example, we use $Move(u, v, w)$ to denote the action of moving object u from v to w, $On(x, y)$ denotes that object x is on object y, $Colour(o, c)$ denotes that object o has the colour c. For fluent On, we need:

$$Holds(On(x, y), s) \wedge x \neq u \Rightarrow Holds(On(x, y), Result(Move(u, v, w), s))$$

This axiom says that if x was on y before moving u from v to w (so long as x isn't the object being moved) and x is not equal to u, then x is still on y after performing action moving v to w. For fluent $Colour$, we need a similar frame axiom:

$$Holds(Colour(o, c), s) \Rightarrow Holds(Colour(o, c), Result(Move(u, v, w), s)).$$

Explicit frame axioms have come under criticism, primarily because there are too many of them, both to write down explicitly and to reason with efficiently. There are several ways to try to overcome this problem. Haas [Haa87] first introduced a technique called *explanation closure*. To explain this technique, consider the fluent

Holding, and suppose that both $Holds(Holding(r, o), s)$ and $\neg Holds(Holding(r, o), Result(a, s))$ are true, where $Holding(r, o)$ states that the robot r holds the object o. How can we explain the fact that *Holding* ceases to be true? If we assume that the only way this can happen is if the robot r put down or dropped the object o, we can express this with the explanation closure axiom:

$$Holds(Holding(r, o), s) \wedge \neg Holds(Holding(r, o), Result(a, s))$$
$$\Rightarrow a = PutDown(r, o) \vee a = Drop(r, o).$$

To see how this functions as a frame axiom, rewrite it in the logically equivalent form:

$$Holds(Holding(r, o), s) \wedge a \neq PutDown(r, o) \vee a \neq Drop(r, o)$$
$$\Rightarrow Holds(Holding(r, o), Result(a, s))$$

This says that all actions other than $PutDown(r, o)$ and $Drop(r, o)$ leave fluent *Holding* invariant, which is the standard form of a frame axiom.

The explanation closure axioms signifies that if ever a fluent changes its truth value, then the corresponding alternative actions given provide an exhaustive explanation for that change. Explanation closure axioms are an effective substitute for frame axioms, and are much more succinct [Sch90, Rei91]. They form the basis of a whole class of so-called monotonic solutions to the frame problem.

However, many researchers abandon frame axioms altogether, and have built models that use persistence or inertia assumptions (e.g., [Lif87, Sho88]). These approaches *assume that all changes caused by an action are specified, and every property not asserted to change does not change*. This technique has a significant advantage, i.e. it eliminates the need to enumerate frame axioms. Other approaches work instead by minimizing event or action occurrences. Properties are assumed to change only as a result of events defined in the representation, and logically unnecessary events do not occur (e.g., [MoS88, StM94, GoG96 etc.]).

In Shanahan's book [Sha97], he suggests three criteria that a satisfactory solution to the frame problem should meet.

- Representational parsimony
- Expressive flexibility
- Elaboration tolerance

Shanahan claims that the explanation closure technique meets the criterion of representation parsimony. However, it does not meet the other two criteria very well. For the criterion of expressive flexibility, the difficulty is to handle some more complicated domains that may have features like ramifications, concurrent actions, continuous change etc. For the criterion of elaboration tolerance, the drawback of this technique is that the acquisition of new knowledge about the domain necessitates the complete reconstruction of the domain theory. This made researchers move to nonmonotonic logics. In section 4.4, some popular nonmonotonic techniques for the frame problem will be examined. Also in chapter 6, the conventional circumscriptive technique is going to be revised for dealing with the frame problem within the proposed formalism.

4.2 Ramification problem

The second problem is the ramification problem (named by Finger [Fin87]), which is to specify everything that changes. The essence of this problem for reasoning about actions and change has been described by Ginsberg and Smith as follows: "The difficulty is that it is unreasonable to explicitly record all of the consequences of actions. For example, if we move the bookcase, all of the books inside move along with it. Also the papers and doilies on top move. For any given action there are essentially an infinite number of possible consequences that might occur, depending upon the details of the situation in which the action occurs." [GiS87a]

The frame problem is largely a representational problem: the idea is to predict that properties remain the same. In the case of predicting that things change as actions occur, however, there is no objection to causal axioms stating that an action causes a particular effect to take place. In addition, there is a computational problem: figuring out everything that has changed when an action is performed can be a very time-consuming task. This is specially true if the world is very interconnected or if there are causal chains. For example, if John carries his briefcase into his office, then everything in the briefcase will also be in his office. So the result of the carrying action is not only that John and his briefcase are currently in his office, but also that his pens, his blue pad, the draft paper that he is writing up are in his office. The problem of how to represent all consequences of an action is known as the ramification problem [Fin87].

Early systems such as STRIPS completely ignored ramification considerations, but it quickly became clear that as soon as toy domains, such as STRIP's block worlds, were replaced by more realistic ones the ramification problem needed to be tackled. Ginsberg and Smith [GiS87a] proposed that, rather than attempting to write causal axioms in such a way as to capture most of the potential effects of an action, only the immediate, forseeable result should be represented in this way. Any indirect effects should be expressed in terms of domain constraints.

Along with the study of the frame problem, its dual problem, the ramification problem is unavoidable. A number of solutions have been proposed [e.g. LiR94, KaL94, San95, 96, Thi97, McT95, KaM97, Sha97 etc.]. As mentioned by Sandewall [San96], most solutions require causal laws to specify the most significant effects of the action, and to rely on domain constraints for specifying additional changes that are due to the action. However, the use of domain constraints alone is not sufficient. Two methods have been taken in order to overcome the drawbacks of the approaches that rely on domain constraints too much. One is *minimisation based*, the other is *causation oriented*. Minimisation based approaches assume that the total set of those changes in fluents that result from the action, consists of the changes that are explicitly stated in the causal law, and a minimal set of other changes while satisfying the domain constraints. Causation oriented approaches require instead that some information is available about how changes in one fluent may "cause" changes in another fluent, and accept those models which satisfy causal laws and domain constraints, where only correctly "caused" changes are present. In chapter 7, the application of the formalism proposed in this dissertation to the ramification problem will be demonstrated.

4.3 Qualification Problem

The third problem in reasoning about actions and change is the so-called qualification problem, also named by McCarthy [McC77]. It differs from the frame problem and the ramification problem as it is about the qualifications of the performance of actions, rather than change itself. In fact, it has the distinction of being the only one of the three classical problems that has meaning outside the temporal domain. Put simply, it states that any action has a number of preconditions that must be satisfied if it is to succeed. The problem is that the number of such preconditions for each action is immense. Imagine all of the things that could prevent the robot from moving the bookcase: it could be too heavy with all of the books in it, it could be too fragile to survive the move, it could be fastened to the floor, the floor where we might want to put it might be too weak to hold it, the door might be too small, or the house might catch on fire. Computationally, we cannot afford to check all of these unlikely possibilities explicitly.

As described by Ginsberg and Smith [GiS87b], the qualification problem consists of three distinct difficulties:

1. The language or ontology may not be adequate for expressing all possible qualifications on the action,

2. It may be infeasible to write down all of the qualifications for an action even if the ontology is adequate, and

3. It may be computationally intractable to check all the qualifications for every action that is considered.

These issues prompted many researchers, i.e. McCarthy [McC80] and Reiter [Rei80] to develop nonmonotonic logics that would allow one to ignore most qualifications unless they could be shown to actually be relevant to a particular application of an action. This technique has been widely adopted and extended to cope with the qualification problem and seems to solve this problem adequately [e.g. GiS87b, Sho88, San94, Lif94, Elk95, Tri97b etc.].

The qualification problem is easily muddled up with the frame problem itself since, like the frame problem, it concerns the effects of actions, and default reasoning techniques seem to be needed to overcome it [Sha97]. The essence of the qualification problem is: How can we be sure that all the preconditions that we have built into our causal axioms are all the preconditions there are in the world one wants to model?

In chapter 6, we will briefly show how to express the preconditions of actions/events, and therefore, to use the nonmonotonic reasoning technique such as circumscription to formalise the assumption that the known preconditions of each action are the only preconditions. Since the qualification problem has little to do with the temporal domain, we are not going to discuss it further in details in this dissertation.

4.4 Nonmonotonic reasoning

As it is known, the frame problem is a crucial problem in reasoning about actions and change [Sha97]. Techniques for the solutions to the frame problem can be revised for dealing with the other two relative problems, ramification and qualification problems. Additionally, in section 4.1, we have already mentioned that to meet the three criteria for a satisfactory solution to the frame problem set by Shanahan, nonmonotonic reasoning appears to be the best candidate. In this section, I look more closely at this subject. The discussion starts by describing what we mean by nonmonotonic reasoning in more precise terms. To do that, let us introduce a definition of monotonicity first. Consider the relationship between sets (not conjunctions) of sentences Γ and the sentences ϕ that follow from Γ in a logical system. If ϕ follows from Γ according to that system, we will write $\Gamma \models \phi$. A logical system is *monotonic* if for any two sets of formulae Γ_1 and Γ_2 where $\Gamma_2 \supset \Gamma_1$ and any formula ϕ, $\Gamma_1 \models \phi$ implies $\Gamma_2 \models \phi$. A logic is termed nonmonotonic if it does not obey the principle of monotonicity.

Nonmonotonic reasoning first gained serious attention in 1980 when two of the most influential formalisms were introduced in a special edition of *Artificial Intelligence Journal*. McCarthy's circumscription [McC80] and Reiter's default logic [Rei80] both provide a way of overcoming the qualification problem by extending first order logic. Circumscription works by minimizing the extent of qualifications - allowing one to ignore qualifications unless they were actually relevant. Default logic provides a nonmonotonic inference procedure based on a special kind of meta-rule. Both systems proved extremely popular until Hanks and McDermott demonstrated that they were unable to solve a relatively simple example of the frame problem: the so called Yale Shooting Problem [HaM87]. This startling result proved to be the death knell of default logic as a vehicle for reasoning about actions and change. However, after the publication of the Yale Shooting Problem, a number of extensions of circumscription have been proposed that seek to solve the frame problem and the ramification problem and have been proved to gain a great success. For this reason we will start by providing a thorough description of circumscription and then go to the extended variants.

4.4.1 Circumscription

Circumscription is a form of nonmonotonic reasoning, introduced by McCarthy [McC77] as a way of characterizing defaults using second order logic. It is the first application of the idea of relative likelihood, or preference between models, to nonmonotonic reasoning. In [McC77], McCarthy recognized that a natural way of representing defaults was to order states of the world, according to what we thought was the case, and then to choose the sentences true in the minimal models in this order as our current beliefs. In fact, circumscription refers to a group of broadly similar systems (domain, predicate, formula, pointwise and prioritized circumscription) all of which make use of a *circumscription policy*. Circumscription allows us to infer that unless something is explicitly known to be the case (or must follow from what is known) then we may effectively ignore it by assuming it to be false. In what follows, we introduce the basic concept of circumscription first and then discuss some of its extensions.

Intuitively, as its name suggests, the basic idea of circumscription is to limit the set of objects of which a predicate is true. Such a process is termed *minimising* the predicate. Circumscribing a predicate in a sentence means assuming that the extent of the predicate is as small as possible.

Before giving the formal definition of circumscription, let's explain the following notion first: for any predicate symbols P, Q of the same arity, let $P = Q$ stand for $\forall x(P(x) \Leftrightarrow Q(x))$ and $P \leq Q$ stand for $\forall x(P(x) \Rightarrow Q(x))$. Let Z stand for the tuple Z_1, Z_2, ..., Z_m of object, function, and/or predicate constants.

Definition 4.1 Let A(P, Z) be a sentence containing Z and a predicate constant P. The circumscription of P in A with Z varied, denoted by CIRC(A; P; Z), is the sentence

$$A(P, Z) \wedge \neg\exists p, z(A(p, z) \wedge p < P)$$

Here p is a predicate variable of the same arity as P, z stands for an m-tuple of variables which matches the m-tuple Z in arity and type, and $p < P$ stands for $(p \leq P) \wedge \neg(p = P)$. The object, function and predicate constants that are not included in Z and are different from P are said to be fixed in CIRC(A; P; Z). This abbreviates a formula in second order logic that selects those models of A in which the extension of the predicate P is minimal with Z varied (in the set inclusion sense).

Generally we can extend this definition to one which allows P be a tuple of predicates. These predicates can be circumscribed "in parallel" or assigned different "priorities". The exact choice of which predicates are minimised in a circumscription, the order in which they are minimised, and which predicates are allowed to vary, is called the circumscription policy.

To see how this technique works, let's look at the following example:

Consider a variation of the Blocks World in which there are more than two blocks. Each block can be somewhere on the table or on top of exactly one other block. Each block can have at most one other block immediately on top of it. To express the information about the world, we introduce three unary predicate constants: *Block*, *Ontable* and *Ab*. *Block*(x) means that object x is a block. *Ontable*(x) means that object x is on table. If x is a block, *Ab*(x) means that block x is an abnormal block with respect to the normal block. Normally we have that a block is on table. That says we call a block a normal block if it is on table. We would like to represent information about the locations in the blocks world, using the standard default rule:

(Block1) $\quad Block(x) \wedge \neg Ab(x) \Rightarrow Ontable(x)$

Then information about the location of a specific block will have to be included in the database only if that block is an exception to this default. For instance, block B_1 is one of such blocks. We have

(Block2) $\quad \neg Ontable(B_1)$

If B_2 is another block, in addition, we may use the fact that B_1 and B_2 are two different blocks,

(Block3) $\quad Block(B_1) \wedge Block(B_2) \wedge B_1 \neq B_2$

then we expect that a general logical mechanism of default reasoning will allow us to conclude:

(Block4) $\quad Ontable(B_2)$

This conclusion can not be achieved by classical logic, as (Block4) is not a logical consequence of the axioms (Block1)-(Block3). However, the formal definition of circumscription gives the conclusion we expect.

Proposition 4.1 [Lif94] Let Σ be the conjunction $\Sigma_1 \wedge \Sigma_2 \wedge \Sigma_3$, where

$$\Sigma_1 = Block(x) \wedge \neg Ab(x) \Rightarrow Ontable(x),$$

$\Sigma_2 = \neg Ontable(B_1)$,

$\Sigma_3 = Block(B_1) \wedge Block(B_2) \wedge B_1 \neq B_2$,

then

$$\mathrm{CIRC}(\Sigma; Ab; Ontable) \models Ab(x) \Leftrightarrow x = B_1.$$

Circumscription can also be presented using model theory. Many people find the model theoretic version easier to understand than the second order presentation. It is certainly sometimes easier to prove properties of a circumscription by appealing to the model theory. What exactly are we doing when we circumscribe a predicate? It seems that we are limiting the ways in which we complete a theory by using a set of preference criteria to select models that close the world with respect to that predicate. This process is known as *minimization* and we say that the circumscribed predicate is minimal in extent. McCarthy gives a formal definition of these preference criteria and the version below is an adaptation of Shanahan's reworking of these preference criteria [Sha97].

Definition 4.2 Let M_1 and M_2 be interpretations of a theory. M_1 is *as small as* M_2 *with respect to* a predicate ρ allowing a tuple σ of predicate, function, and constant symbols *to vary*, written as $M_1 \sqsubseteq_{\rho;\sigma} M_2$, if

- M_1 and M_2 agree on the interpretation of everything except possibly ρ and/or zero or some members of σ, and
- The extension of ρ in M_1 is a subset of its extension in M_2.

Definition 4.3 A model M_1 of a theory Σ is *minimal with respect to* $\sqsubseteq_{\rho;\sigma}$ if there is no models M_2 of Σ such that $M_2 \sqsubseteq_{\rho;\sigma} M_1$ and not $M_1 \sqsubseteq_{\rho;\sigma} M_2$.

Two features of this preference relation among models are worth noting:

- The comparison between models is based on set inclusion, and not on set cardinality. Although they may agree on the interpretation of everything except ρ and/or zero or some members of σ, and the extension of ρ may have fewer members in M_1 than in M_2, it is not necessarily the case that $M_1 \subseteq_{\rho;\,\sigma} M_2$ (not a subset).

- As a consequence of this, it is not always the case that two models are comparable. In other words, it is possible neither to have $M_1 \subseteq_{\rho;\,\sigma} M_2$ nor to have $M_2 \subseteq_{\rho;\,\sigma} M_1$.

Definition 4.4 Let M_1 and M_2 be interpretations of a theory. M_1 is *preferable to* M_2 with respect to a circumscription policy if M_1 is as small as M_2 but M_2 is not as small as M_1 with respect to that policy.

Probably the most attractive feature of the circumscriptive approach is that it proceeds simply by adjoining the sentence: $A(P, Z) \wedge \neg\exists p, z(A(p, z) \wedge p < P)$ in definition 4.1 to the original theory. No fixed-point equations need to be solved, as in the consistency-based approaches (such as default logic, autoepistemic logic and model approaches). This tends to make the semantics of this approach cleaner than that of others.

The other feature we want to mention is, to deal with the default rules we have to face the question: how are we going to capture formally the notion that one default rule may be more important than another? For instance, there may be two conflicting default rules. To address this issue, McCarthy [McC80] introduces a notion of prioritized circumscription. This idea is generalised by Lifschitz in his work on pointwise circumscription [Lif87], and further by Baker [Bak91] and Shanahan [Sha97] on state-based circumscription.

To summarise the discussion of circumscription, as many researchers point out, circumscription is a simple and natural extension of classical logic. It makes a clear separation between the classical aspects of a representation - those facts that are known to be true, and from which we can draw deductively valid inferences - and the nonmonotonic aspects - default assumptions whose consequences we would be willing to give up in the face of contradicting evidence [Sha97]. However, there are still some unsolved problems of this approach. The complexity of circumscription is one of them. It has resulted in a certain reluctance on the behalf of researchers to use it with working AI applications. Various application areas have been suggested but none has resulted in a real program. Of course, none of these proposed applications will amount to much in the absence of an effectively implemented nonmonotonic reasoning system. The difficulty in developing such an implementation is that none of the formal descriptions provides a constructive definition of a valid nonmonotonic derivation. The consistency-based approaches require solving a particular fixed-point equation; there are no known general-purpose techniques for solving such kind of equation. Circumscription, meanwhile, uses a second-order axiom; once again, there are no complete proof procedures for second-order theories. Any success on these areas may lead to significant progress.

4.4.2 The Yale Shooting Problem

The crucial part of the application of circumscription is to select the circumscrition policy. Consider the standard *common-sense law of inertia* in situation calculus:

(CLI) $\neg Ab(f, a, s) \Rightarrow Holds(f, s) \Leftrightarrow Holds(f, Results(a, s))$.

This says that the value of a fluent persists from one situation to the next situation unless something is abnormal. The policy of minimising *Ab* while allowing *Holds* to vary is an obvious one, and it seems as if it should solve the frame problem. In 1986, however, Hanks and McDermott showed that this policy fails to generate the conclusions we expect even with extremely straightforward examples. They distilled the essence of the difficulty into a single, simple example, the now famous Yale Shooting Problem.

This Yale Shooting Problem was discovered when Hanks and McDermott attempted to integrate temporal and nonmonotonic logics. It can be described as follows: We are told that a gun is loaded at the original situation, and that the gun is fired at Fred after waiting for a while. Firing a loaded gun at an individual causes the person to be dead. In addition, the fluents *Alive* and *Loaded* persist as long as possible; that is, these fluents stay true unless an action that is abnormal with respect to these fluents occurs. Thus, a person who is alive tends to remain alive, and a gun that is loaded tends to remain loaded. What can we conclude about Fred's status after firing action occurs? Assume that the original situation is S_0. We can ask the following question: Is

$$Holds(Alive, Result(Shoot, Result(Wait, S_0)))$$

true?

Although common sense argues that Fred is dead after shooting because we have the knowledge that the gun is loaded at the original situation and there is nothing to tell it is unloaded somewhere, the facts support two models. In one model (the expected model), the fluent *Loaded* persists as long as possible. Therefore, the gun remains loaded until it is fired at Fred, and Fred dies. In this model, action *Shoot* is abnormal with respect to fluent *Alive*. In the other model (unexpected model), the fluent *Alive* persists as long as possible, i.e., Fred is alive after shooting. Hence, the fluent *Loaded* did not persist: somehow the gun must be unloaded by *Wait* action. In this case, action *Wait* is abnormal with respect to fluent *Loaded*. Each model includes one abnormality. It is as if we have traded one abnormality for another, and both models are minimal with respect to the extension of predicate *Ab*.

Here, the existence of multiple minimal extensions is a genuine problem. If, given the simple set of assumptions above, one cannot even conclude that the gun stays loaded and Fred is dead after shooting, in what sense can we say that the circumscription with the policy of minimising *Ab* while allowing *Holds* to vary has solved the frame problem? Hanks and McDermott in fact argued that the existence of the Yale Shooting Problem underscored the inadequacy for reasoning in temporal domains. This discovery revolutionized the area of reasoning about actions and change. People began to design systems specifically for the temporal domain, either by extending circumscription to take into account temporal considerations or by proposing entirely new nonmonotonic reasoning formalisms. The resulting work mainly fell into three categories: systems based on chronological minimisation, systems based on causal minimisation and systems based on state-based minimisation. (The third category, state-based minimisation is named by Shanahan [Sha97]). In the remainder of this chapter we will examine the most important issues of these systems.

4.4.3 Chronological minimisation

Hanks and McDermott [HaM86, 87] argue that the problem with general nonmonotonic logics is their failure to incorporate the notion of time. In particular, they claim that time creates an explicit ordering, and temporal reasoning is inherently biased towards that ordering. That is the main reason for the failure. To see this, let's look at the Yale Shooting Problem, the expected model arises when we reason about world states in temporal order: *Alive* and *Loaded* holds at the beginning S_0, so they will (by default) hold after waiting action. This means that after waiting for a while, *Loaded* still holds, when the gun is fired, *Alive* does not hold any more. That is

$$\neg Holds(Alive, Result(Shoot, Result(Wait, S_0))).$$

In contrast, the unexpected model arises when we apply persistence to Alive:

$$Holds(Alive, Result(Wait, S_0)) \wedge Holds(Alive, Result(Shoot, Result(Wait, S_0)))$$

before we have reached any conclusion about Loaded after waiting action. This persistence to *Alive* force us to reason backwards about *Loaded*. It must be the case that:

$$\neg Holds(Loaded, Result(Wait, S_0)).$$

The first solutions to this problem work by imposing a forward-in-time order on reasoning. They are thus known as the *chronological minimisation*. Each of these solutions -- Hanks and McDermott's program [HaM86], Shoham's logic of chronological ignorance [Sho86, 87a], Kautz's logic of persistence [Kau86], and temporal applications of Lifschitz's pointwise circumscription [Lif86] -- describes a reasoning system with an inherent forward temporal bias. Each works by considering world states in their chronological order, extending as much persistence as possible through earlier world states before addressing later world states. This approach gives a particular preference over sequences of world states: we prefer that *as little happens* *for as long as possible.* In the Yale shooting case, in the initial situation, S_0:

$$Holds(Alive, S_0) \wedge Holds(Loaded, S_0).$$

By default, *Alive* and *Loaded* persist after waiting action since they are not forced to change:

$$Holds(Alive, Result(Wait, S_0)) \wedge Holds(Loaded, Result(Wait, S_0)).$$

Then by the causal law that "Firing a loaded gun at an individual causes the person to be dead" and the fact that the gun is loaded, we reach the expected conclusion:

$$\neg Holds(Alive, Result(Shoot, Result(Wait, S_0))).$$

The main idea of this approach is to postpone changes until they are forced; or to allow persistence to continue for as long as possible. This avoids the unexpected model that arises for the standard nonmonotonic logics.

The three chronological logical approaches [Sho86, Kau86, Lif86] essentially mimic the behaviour of Hanks and McDermott's program. Kautz and Lifschitz use circumscription to fix state values in one world's state before considering the next; Shoham defines a model preference criteria with the same properties. This approach prefers sequences of world states that minimise changes to the world; persistences apply whenever possible. As summarized by Bell [Bel89], chronological minimisation captures two principles of reasoning about change:

(1) All change is rule governed: nothing changes unless it is caused to. To allow otherwise is to allow miracles, and allowing miracles involves a more complex form of reasoning.
(2) The causal relation is temporally directed; that is, when reasoning causally we reason forwards in that time from cause to effect.

Problems with Chronological Minimisation For several reasons, forward reasoning solutions are not entirely satisfactory. The most obvious one is that causation is not merely time-moving-forward. The applicability of this kind of solutions seems to be limited to what Hanks and McDermott call temporal projection problems, or in other words, problems in which given initial conditions, we are asked to predict what will be the case at a later time. One can also consider temporal explanation problems [HaM87], i.e., problems requiring reasoning backwards in time.

For example, consider the Stolen Car Scenario, which is best thought as an explanation problem. Suppose that John parks his car in the morning and goes to work. According to his knowledge at lunch time, common sense should allow him to infer by default that the car is still there where he left it. However, when John returns to the car park in the evening he finds that it has gone. Its disappearance requires an explanation. That is, we have to reason backwards in time to the possible causes of the car's disappearance. In this case, the only reasonable explanation for the car's disappearance is that it was stolen some time between morning and evening. The car may have been stolen any time after John parked it and before John observed that it was gone, so he can not say anything about its whereabouts at lunch time.

Kautz first noted this point when he presented his solution to the YSP [Kau86]. As he observed, the trouble with chronological minimisation is that, since it postpones change until as late as possible, it will yield the conclusion that the car was stolen immediately before John's return to the car park in the evening. This conclusion is clearly too strong. It may have been stolen at lunch time or just after John parked it.

In addition, other shortcomings of chronological minimisation in its original form have been addressed by Sandewall [San93, 94]. Sandewall introduces two significant innovations: *filtered preferential entailment and occlusion*. He employs a narrative-based temporal logic rather than the situation calculus and event calculus. However, as Shanahan [Sha97] points out, variants of these two ideas are applicable to both situation calculus and event calculus. In chapter 6, we will show how to apply those ideas to our formalism for dealing with the explanation problem.

4.4.4 Causal Minimisation

Unlike chronological minimisation, causal minimisation approaches [Hau87, Lif87] are not concerned with the time at which things change, but with the reason why the changes occur. Rather, they employ a predicate, *Causes* to avoid the temporal projection problem by circumscribing *Causes*. Circumscribing *Causes* means that we prefer sequences of world states in which actions have fewer effects. In causal minimisation, every change has to be caused by actions, and all changes that can not be explained in terms of the effects of known actions are minimised.

As in Yale Shooting Problem, one can observe that in the anomalous model there is a change that is not caused by any action. The gun becomes unloaded spontaneously, for no reason. This observation motivated the causal minimisation. Consider the YSP as an example, in the initial situation, S_0, we have the fact:

$$Holds(Alive, S_0) \wedge Holds(Loaded, S_0),$$

and a causal axiom:

$$Causes(Shoot, Alive, False).$$

This axiom says that the execution of action *Shoot* will cause the fluent *Alive* having truth value *False*. There are no explicit causal consequences for action *Wait*. The circumscription policy is circumscribing *Causes* while allowing *Holds* to vary. Circumscribing *Causes* now yields no implicit causal consequences for action *Wait*, so we can say that nothing changes during the *Wait* action. Thus, *Loaded* persists:

$$Holds(Loaded, Result(Wait, S_0))$$

and therefore

$$\neg Holds(Alive, Result(Shoot, Result(Wait, S_0))).$$

N.B. In Lifschitz's Formal Theories of Action [Lif87], there is a predicate, *Precond*, which tells when an action can be successful. This predicate can be used to deal with the *qualification problem*.

Involving a new way of writing effect axioms: *using* predicate *Causes* instead of *Ab*, causal minimisation represents causality explicitly by stating that a fluent changes its truth value if and only if a successful action causes it to do so. This leads to the expected conclusion.

But what lies behind the success of the causal minimisation? In the anomalous model arising with the naive approach, the gun becomes unloaded during the *Wait* action. In other words, we have *Ab*(*Wait*, *Loaded*, *Result*(*Loaded*, S_0)). In terms of causal predicates, an anomalous model would have to have *Causes*(*Wait*, *Loaded*, *False*). In the naive approach, the presence of the *Wait* abnormality avoids the need for *Shoot* to be abnormal with respect to *Alive* when a shooting takes place after a waiting. Naturally, circumscription will prefer models in which *Ab*(*Wait*, *Loaded*, *Result*(*Loaded*, S_0)) is absent. Hence we may have the anomalous model, i.e., we are not able to deduce the intuitive conclusion that Fred dies.

However, with causal minimisation, this trading of abnormalities is not possible. The formula *Causes*(*Shoot*, *Alive*, *False*) is an axiom. We can't get rid of it from our models, whatever else we add. This is an important idea of causal minimisation, called the *principle of separation* by Shanahan [Sha97]. It is a key to causal minimisation. Causal minimisation differs from the chronological minimisation in two major ways: (a) there is an explicit attempt to make sure that causal minimisation works for both forward and backward reasoning, and (b) there is an explicit attempt to base the solutions to the Yale Shooting Problem on a strong intuition about temporal reasoning.

Problems with Causal Minimisation Causal minimisation is a very simple but effective way of handling the Yale Shooting Problem. However, there are some weaknesses with this approach. Firstly, it does not allow us to write our domain axioms in unrestricted situation calculus. Instead, we must use the *Causes* predicate. There is simply no way to use this predicate to express general context-dependent effects, domain constraints, or ramifications [Bak91]. In particular, it is difficult to represent actions with context-dependent effects, that is, actions whose effects dependent on the fluents that hold at the time the action is performed. For example, if I toggle a light switch, the light comes on if it is off at the time, but goes off if it is on. This knowledge can be expressed using the following axioms:

$$\neg Holds(On, s) \Rightarrow Holds(On, Results(Toggle, s))$$

$$Holds(On, s) \Rightarrow \neg Holds(On, Results(Toggle, s))$$

The difficulty for causal minimisation to express such knowledge is that no corresponding set of *Causes* sentences is possible.

Secondly, it does not allow concurrent actions and uncertain actions. These constraints limit the expressive power of the approach. Finally, like chronological minimisation, it has trouble with the explanation problem. For example, in the stolen car scenario, since the *Causal* predicate is independent of time, and every change should have a causation, the causal minimisation will lead to an unintuitive consequence. That is a *Wait* action *always* makes fluent *Stolen* hold, whatever the circumstances. Even if I recover my car, next time when I perform a *Wait* action, the car will be stolen again.

Motivated Action Theory (MAT) Stein and Morgenstern [StM94] present a more advanced causal minimisation system, which is partially motivated by the examination of the deficiencies of Haugh and Lifschitz' formalisms. As Stein and Morgenstern point out in their paper, "from the chronological approaches, and from Haugh's causal minimisation, we adapt the idea of minimising actions; from the causal approaches, we borrow the idea of minimising causes." Unlike most previous solutions to the Yale Shooting Problem, MAT is not based on the situation calculus. Instead it is based on a simple, interval-based theory of time. In MAT, concurrent actions and uncertain actions are allowed.

Motivated Action Theory is based on the principle that an agent typically knows all that he needs to know in order to make predictions about the world in which he lives.

In particular, if he is reasoning with an underconstrained set of facts about the world, he will tend to prefer models that have fewer unexpected, *unmotivated* actions.

A description of a problem scenario in MAT is termed as a theory instantiation, consisting of a theory *T* and a (partial) chronical description *CD*. Intuitively, *T* gives the rules governing the world's behaviour and contains *causal rules* and *persistence rules*. Causal rules describe how actions change the world; persistence rules describe how fluents remain the same over time. *CD* describes some of the facts that are true and the actions that occur during a particular interval of time.

Central to MAT is the concept of motivation. Intuitively, an action is motivated with respect to a theory instantiation if there is a "reason" for it to happen. The most important types of motivation are *strong* and *weak* motivations. An action is strongly motivated if it "has to happen" in all the models (i.e., it is a theorem that the action happens). An action is weakly motivated if it "has to happen" with respect to a particular model (i.e., if it must occur given the particular way the model is set up). The notion of weakly motivated can be used to treat uncertain actions.

To deal with the frame problem, Stein and Morgenstern develop a model theoretic *preference relation* that is in a similar manner to Shoham's preference relation. The preference relation is based on the idea of motivated actions. Given a theory instantiation $TI = T \vee CD$, a statement ϕ is said to be strongly motivated in $M(TI)$, a model for a theory *TI*, if it is present in all models of *TI* . A statement ϕ is said to be weakly motivated in $M(TI)$ if there exists a causal or persistence rule of the form $\alpha \wedge \beta \Rightarrow \phi$, where α is motivated in $M(TI)$ and $M(TI) \models \beta$. A model $M(TI)$ is preferred to a model $M'(TI)$ if any action that occurs in $M'(TI)$ is motivated in $M(TI)$. In other words, MAT prefers models in which *as few unmotivated actions as possible occur*.

Let's use the Yale Shooting Problem to show how this preference relation works. Here we simply use integers 1, 2, 3, ..., as the time index. The scenario can be described as follows:

chronicle description, *CD*:

$Holds(Alive, 1)$

$Holds(Loaded, 1)$

$Occurs(Shoot, 3)$

causal rules and persistence rules, *T*:

$$Occurs(Shoot, t) \wedge Holds(Loaded, t) \Rightarrow \neg Holds(Alive, t{+}1)$$

$$Holds(Alive, t) \wedge ((\neg Occurs(Shoot, t) \vee \neg Holds(Loaded, t)) \Rightarrow Holds(Alive, t{+}1)$$

In the expected model, *Loaded* persists from time 1 to time 3 and Fred dies at time 4:

$\neg Holds(Alive, 4)$.

In the unexpected model, *Alive* persists from time 1 to time 4, that is:

$Holds(Alive, 4)$.

In the first model, every change is motivated. There are no unmotivated actions. In the second model, fluent *Loaded* changes its truth value from time 1 to time 3. This change is unmotivated. That means there must be an unmotivated action occurring between time 1 and time 3. Then by minimising unmotivated actions, MAT prefers the first model over the second one.

Unlike many other causal (and all chronological) minimisation based systems, MAT is capable of handling explanation problems. As we mentioned above, Haugh and Lifschitz's causal minimisation failed to solve the stolen car problem, since the *Causal* predicate in their formalism is independent of time. The causal rules in MAT do depend on time, so it can solve the stolen car problem correctly. In fact, the stolen car scenario can be described as:

chronical description, *CD*:

$\neg Holds(Stolen, 1)$

$Holds(Stolen, 4)$.

There are no causal rules. At time 4, the car is observed to have disappeared. What has happened then? We prefer sequences in which the thieves steal the car to sequences in which the car disappears without explanation. That means some action must happen at time 2, or 3 which causes the car to be stolen:

$Occurs(Steal, 2) \vee Occurs(Steal, 3)$.

MAT tell us nothing more than this. In fact, there are two models that satisfy MAT, one in which the *Steal* occurs at time 2, and the other in which the *Steal* occurs at time 3. Intuitively, there does not seem to be a reason to prefer one of these two models to the other, since we have not had any information when the car is stolen. *Steal* is not a motivated action. However, it does not violate the MAT's minimising policy: *as few unmotivated actions as possible occur*.

Problems with MAT MAT has many advantages over Haugh and Lifschitz' formalisms. It allows for forward and backward temporal reasoning, supports a flexible temporal ontology. However, there are significant problems with this formalism. The persistence rules are essentially frame axioms. Although Stein and Morgenstern have suggested that there may be some way of automatically generating these, they do not provide a practical method of doing this. In fact, Sandewall [San92] casts doubt on the existence of an automated persistence rule generating procedure.

4.4.5 State-based minimisation

As Hanks and McDermott pointed out in [HaM86], the original circumscription policy of minimising predicate *Ab* while allowing predicate *Holds* to vary fails to deal with the projection problem. They propose the Yale Shooting Problem, which motivates the following chronological and causal minimisations. Later on, Baker reasoned that as all attempts to use chronological or causal minimisation had produced systems that were flawed to a greater or lesser extent, the obvious course to take was to try an entirely different approach. In 1989, Baker described a formalism that implemented this philosophy. Like many other systems, Baker used the situation calculus as the representation language but he did not make use of causal predicates. Unlike chronological minimization, it does not require a new default reasoning formalism. Unlike causal minimization, it does not involve a new way of writing effect axioms. Instead, Baker's approach simply requires the adoption of a few extra axioms, and a slight modification of the original circumscription policy.

Baker's approach has two key components. First, he changed the circumscription policy. Instead of varying the *Holds* predicate, the *Result* function is allowed to vary. Second, an axiom is added that guarantees the existence of a situation for every possible combination of fluents. Let's use the Yale shooting scenario to see how Baker's approach works.

As stated in section 4.4.2, in the anomalous (unexpected) model of the Yale shooting scenario, one abnormality is traded for another. At the cost of gaining the abnormality $Ab(Loaded, Wait, S_0)$, we have managed to shed the abnormality $Ab(Alive, Shoot, Result(Wait, S_0))$. This trade is possible because in the anomalous model $Result(Wait, S_0)$ does not denote a situation in which *Alive* and *Loaded* hold. However, $Result(Wait, S_0)$ does denote a situation in which *Alive* and *Loaded* hold, and indeed every model that is a minimal model with respect to the new circumscription policy satisfies $Ab(Alive, Shoot, Result(Wait, S_0))$.

To guarantee the presence of every possible combination of fluents in every model, Baker introduces an important axiom, the existence-of-situations axiom. In the Yale shooting case, the existence-of-situations axiom can be expressed as the following:

$$\exists s(Holds(Alive, s) \wedge Holds(Loaded, s)) \wedge$$
$$\exists s(Holds(Alive, s) \wedge \neg Holds(Loaded, s)) \wedge$$
$$\exists s(\neg Holds(Alive, s) \wedge Holds(Loaded, s)) \wedge$$
$$\exists s(\neg Holds(Alive, \mathrm{s}) \wedge \neg Holds(Loaded, s))$$

This axiom is crucial to Baker's approach. However, it seems not easy to write this kind of axiom if a domain comprises a large number of fluents. Fortunately, Baker provides a means of automatically producing these axioms.

In addition, to get rid of all anomalous models, another axiom called *domain closure axiom* is needed. In this case, it is

$$f = Alive \vee f = Loaded.$$

That means the only fluents in the Yale shooting scenario are *Alive* and *Loaded*, and there can't be any mysterious nameless fluents to bother us. So the existence-of situations covers all possible cases.

Therefore, the new circumscription policy, together with the two extra axioms, provides a proper solution to the Yale Shooting Problem. Baker also shows that this simple modified approach is enough to solve the temporal projection problem and a number of other troublesome examples. By introducing some simple constraints, it can be applied to wide range of different problems, including those involving ramifications and certain explanation problems.

Baker's approach has proved to be the most effective to date as it solves a much wider variety of action and change problems than other formalisms. As mentioned above, one of the key components of this approach is to modify the original circumscription policy and reserve the most features of circumscription. This gives us opportunities to extend further. Recently, Shanahan and Miller [MiS94, Sha95, Sha97] extend this approach to more general cases, and make it more expressive. In what follows, we will adopt Shanahan's notion, call this approach the *state-based minimisation*.

The state-based minimisation is so successful that the only valid criticism that can be applied to it is that, like many other formalisms, it is based on circumscription and seems unlikely to be used as the basis for an implementation.

CHAPTER 5
TIME STRUCTURE

The notion of time is ubiquitous in most activities that require intelligence. Firstly, these activities mostly concern the real world, which is a dynamic world. The facts and phenomena that happen in it occur over time. Secondly, the human perception and understanding of the real world deeply incorporate the concept of time. Everything appears related by its temporal reference. Usually, event occurrences are temporally related ('during 1998', 'in May of 1989', 'on the day of 22 April, 1997 ', ...). Things remain in a certain state for a while until a certain event happens. Time seems to be a fundamental entity with which the rest of the objects in the world are related, and therefore appears to play the role of a common universal reference.

In particular, time is fundamental for reasoning about change and action. The existence of time allows one to describe change and the characteristics of its occurrence (shape, interaction with other occurrences, etc.). The temporal reference is an idea deeply integrated in human commonsense. It is notable how naturally and efficiently humans are able to manage this during everyday life when interacting with the environment.

A representation structure of time must choose the primitive temporal entities that it will use. Structures based on points are more widespread because they unconsciously introduce an association with numerical structures: integers, rationals or reals. However, a representation based on point entities takes into account the concept of duration with difficulty. Structures based on intervals avoid this difficulty. But they have some other limitations. As Galton shows in his critical examination of Allen's interval logic [Gal90], the theory of time based on intervals is not adequate, as it stands, for reasoning correctly about continuous change. Hence for the purpose of reasoning about actions and change, sometimes we may need a special designed time structure.

In this thesis, a discrete time structure will be proposed, which can be seen as a special model of the time theory previously proposed by Ma and Knight in [MaK94]. The theory treats both points and intervals as primitive on an equal footing, and is shown to be powerful enough to cope with dividing instant problem and intermingling problem, and deal with knowledge with duration. The fundamental temporal structure we assume is a simple linear model of time. Although branching time models have been proposed as a useful means of handling possible worlds, uncertainty about the past or the future and the effects of alternative actions when planning, it is argued that it is not necessary to have a theory that assumes that time itself branches. In fact, notions of possibility can be handled by introducing a separate model operator to represent possibility explicitly [AlF94, KnM94].

5.1 Prime Times

The fundamental time structure starts with an ordered set **P** of objects named *prime times*. From **P** to $\mathbf{R}_0^+$, the set of non-negative real numbers, there is a function, *Dur*, which assigns each prime time a non-negative real number denoting its temporal duration. If $Dur(p) > 0$, p is called a (time) *moment*, otherwise, p is called a (time) *point*.

Additionally, we assume that **P** is *similar to* **Z**, the set of integers. In other words, there exists a one-to-one mapping between the elements of **P** and **Z** that preserves the order relation. Such a mapping is called a *similar function* [Lip64]. In what follows, we shall use $Meets(p_1, p_2)$ to denote that prime time p_1 is the immediate predecessor of prime time p_2. Also, we impose the following axiom that ensures that two points can't meet each other:

(5.1.1) $Meets(p_1, p_2) \Rightarrow Dur(p_1) > 0 \vee Dur(p_2) > 0$

From the property of the *similar function*, we have:

- The fundamental time structure is *linear*, not branching from any time into either the past or the future;

- The fundamental time structure is *unbounded* both in the past and future;

It is important to note that prime times have no internal structure. In another words, the elements of **P** are all non-decomposable, even though some of them may have positive duration, i.e. moments [AlH89].

5.2 General Times

The fundamental time structure described above is not adequate enough for general knowledge representation. For instance, since the prime times have no internal structure, knowledge like "on 1st of May, 1998, John was working in his office. In the morning of that day, he was writing a research paper, and in the afternoon, he was preparing for next day's lecture" cannot be expressed. Therefore, we need to extend the concept of prime times.

Based on the set of prime times, **P**, we define the set of (general) *times*, denoted by possibly indexed *t*, **T**, as the minimal set satisfying the following axioms:

(5.2.1) $\forall p \in \mathbf{P}(p \in \mathbf{T})$

That is **P** a sub set of **T**, $\mathbf{P} \subseteq \mathbf{T}$;

(5.2.2) $Meets(p_1, p_2) \Rightarrow Meets_T(p_1, p_2)$

Where $Meets_T$ is the binary relation over **T** and $Meets_T(t_1, t_2)$ denotes that time t_1 is the immediate predecessor of time t_2. Therefore, axiom (5.2.2) states that $Meets_T$ is the extension of *Meets* from **P** to **T**.

(5.2.3) $Meets_T(t_1, t_2) \Rightarrow \exists t \forall t', t'' \in \mathbf{T}(Meets_T(t', t_1) \wedge Meets_T(t_2, t'')$
$\Rightarrow Meets_T(t', t) \wedge Meets_T(t, t''))$

That is, if two times are separated by a sequence of times, then there is a time which connects them.

(5.2.4) $t_1 = t_2 \Leftrightarrow \exists t_1', t_2'(Meets_T(t_1', t_1) \wedge Meets_T(t_1', t_2) \wedge Meets_T(t_1, t_2') \wedge Meets_T(t_2, t_2'))$

That is, two times are identical if and only if they have the same immediate predecessor and the same immediate successor. Hence, by axioms (5.2.3) and (5.2.4), for any two times, t_1 and t_2, such that $Meets_T(t_1,t_2)$, we shall denote their *adjacent union* as a new time (interval), $t = t_1 \oplus t_2$. N.B. $t_1 \oplus t_2$ always implies that $Meets_T(t_1,t_2)$.

(5.2.5) $\exists p_1, \ldots, p_n(t = p_1 \oplus p_2 \ldots \oplus p_n)$

That is every general time is an adjacent union of limited number of prime times. This axiom is crucial for dealing with the *intermingling problem* [Ham71].

In addition, as the extension of the duration assignment function, *Dur*, from **P** to $\mathbf{R_0}^+$, we define Dur_T as the function from **T** to $\mathbf{R_0}^+$, which assigns each element in **T** a non-negative real number, such that:

(5.2.6) $Dur_T(p) = Dur(p)$

(5.2.7) $Meets_T(t_1,t_2) \Rightarrow Dur_T(t_1 \oplus t_2) = Dur_T(t_1) + Dur_T(t_2)$

That is, the duration of the combined times $t_1 \oplus t_2$ is identical to the sum of duration of t_1 and duration of t_2, where "+" is the conventional arithmetic addition operator.

In summary, **T** is in fact the closure of **P**, under operator $\oplus$. Elements in **T** are not necessary non-decomposable. A time t is called a (time) interval if only if there exist time t_1 and time t_2, such that $t = t_1 \oplus t_2$. By (5.1.1), (5.2.2) and (5.2.7), for any interval t, we have $Dur(t) > 0$.

In what follows, without confusion, we shall simply write Dur_T as *Dur*, and $Meets_T$ as *Meets*.

A time t is called *right* (*left*) *closed* if there is a prime time p and a time t' such that $t = t' \oplus p$ ($t = p \oplus t'$) and $Dur(p) = 0$; otherwise t is called *right* (*left*) *open*. In this thesis, we introduce a function, *REnd*, from **T** to set {*open*, *closed*}, which specifies the right end status of a time. For a given time t, if it is right open, then $REnd(t) = open$; if it is right closed, then $REnd(t) = closed$. These definitions are useful when the notion of *duration type* is introduced (see next chapter).

As shown in [MaK94], in terms of the single relation *Meets*, there are in total 30 temporal relations among time intervals/points may be defined. These temporal relations can be classified into the following four groups:

- Temporal relations relating intervals to intervals:

  {*Equals, Before, After, Meets, Met_by, Overlaps, Overlapped_by, Starts, Started_by, During, Contains, Finishes, Finished_by*}

- Temporal relations relating points to points:

  {*Equals, Before, After*}

- Temporal relations relating points to intervals:

  {*Before, After, Meets, Met_by, Starts, During, Finishes*}

- Temporal relations relating intervals to points:

  {*Before, After, Meets, Met_by, Started_by, Contains, Finished_by*}

Definitions of these derived relations are:

$$Equal(t_1, t_2) \Leftrightarrow t_1 = t_2$$

$Before(t_1, t_2) \Leftrightarrow \exists t(Meets(t_1, t) \wedge Meets(t, t_2))$

$Overlaps(t_1, t_2) \Leftrightarrow \exists t, t', t'' \ (t_1 = t' \oplus t \wedge t_2 = t \oplus t'')$

$Starts(t_1, t_2) \Leftrightarrow \exists t(t_2 = t_1 \oplus t)$

$During(t_1, t_2) \Leftrightarrow \exists t', t''(t_2 = t' \oplus t_1 \oplus t'')$

$Finishes(t_1, t_2) \Leftrightarrow \exists t(t_2 = t \oplus t_1)$

$Met\text{-}by(t_1, t_2) \Leftrightarrow Meets(t_2, t_1)$

$After(t_1, t_2) \Leftrightarrow Before(t_2, t_1)$

$Overlapped(t_1, t_2) \Leftrightarrow Overlaps(t_2, t_1)$

$Started\text{-}by(t_1, t_2) \Leftrightarrow Starts(t_2, t_1)$

$Contains(t_1, t_2) \Leftrightarrow During(t_2, t_1)$

$Finished\text{-}by(t_1, t_2) \Leftrightarrow Finishes(t_2, t_1)$

In addition, for the convenience of expression, we may also define that

$In(t_1, t_2) \Leftrightarrow Starts(t_1, t_2) \vee During(t_1, t_2) \vee Finishes(t_1, t_2)$

$Sub(t_1, t_2) \Leftrightarrow Equal(t_1, t_2) \vee Starts(t_1, t_2) \vee During(t_1, t_2) \vee Finishes(t_1, t_2)$

where $In(t_1,t_2)$ denotes that time t_1 is a proper part of time t_2, and $Sub(t_1,t_2)$ states that time t_1 is either a proper part of time t_2 or is t_2 itself. Also we will take use of the following property:

(Pro5.1) $Meets(t, t_1) \wedge Meets(t, t_2) \Rightarrow Starts(t_1, t_2) \vee Starts(t_2, t_1) \vee Equal(t_1, t_2)$

That is, for any time t, if it meets two times, t_1 and t_2, then either t_1 starts t_2 or t_2 starts t_1 or t_1 equals to t_2. This property can be deduced directly from the linear property of the time structure.

For the purpose of reasoning about actions and change, in this chapter, a prime time structure is introduced first, then a general time structure is proposed by means of extending the prime time structure. Such a time model is based on both time points and time intervals, and supports the concept of non-decomposable time moments with positive temporal duration. This temporal structure will be used in the formalism to be presented in next chapter in order to enrich the temporal ontology and help to deal with the *Intermingling Problem* and *Dividing Instant Problem* [Van83], which beset most interval-based temporal systems.

CHAPTER 6

TEMPORAL STATE TRANSITION CALCULUS

Representing and reasoning about the dynamic aspects of the world -- primarily about actions and change -- is a problem of interest to many different disciplines. In the research field of artificial intelligence, we are interested in such a problem for a number of reasons, in particular to model the reasoning of intelligent agents as they plan to act, and to reason about causal effects in the real word. Temporal information plays an important role in the description of the dynamic aspects of the world. As discussed previously, there are mainly two classes of approaches for this purpose, namely constructive models (state-transition-based) and non-constructive models (temporal-logic-based) [AlF94]. Each class has its own advantages. In this chapter, a discrete model for representing and reasoning about actions and change, which is called *Temporal State Transition Calculus* (TSTC), is proposed in attempting to combine most of the benefits from both of these two classes.

The formalism will be described in terms of a many-sorted reified logic with equality, including sorts **P**, **F**, **S** and **A** for *prime times*, *propositional fluents*, *states* and *actions* respectively. Variables are denoted by lower case letters (with or without subscripts), and constants are denoted by upper case letters (with or without subscripts). Unless otherwise stated, letters $\{p, p_1, p_2, \ldots (P, P_1, P_2, \ldots)\}$, $\{f, f_1, f_2, \ldots (F, F_1, F_2, \ldots)\}$, $\{s, s_1, s_2, \ldots (S, S_1, S_2, \ldots)\}$, and $\{a, a_1, a_2, \ldots (A, A_1, A_2, \ldots)\}$ are used for variables (constants) of sorts **P**, **F**, **S** and **A,** respectively. Also, we adopt the conventional theories of *reals* and *integers*. Based on these primitive sorts, some supersorts (which are formed by combining some sorts) will be added when some new entities are introduced. These supersorts will be explained when they appear at first time.

6.1 Fluents Revisited

A fluent is a proposition whose truth value is dependent on times. It may take different values over different times. In order to associate fluents with times, following the common practice in Artifitial Intelligence [DAA95], a meta-predicate, *True*, over **F**×**P** is introduced so that the formula *True*(f, p) represents that fluent f holds true with respect to prime time p.

Corresponding to the extension from **P** to its closure **T**, under operator ⊕, a new sort, general times **T**, (which is a supersort of prime times) is added to the language. Variables (constants) of sort **T** are denoted by lower case letters (with or without subscripts) t, t_1, t_2 ..., and constants are denoted by upper case letters (with or without subscripts) T, T_1, T_2 .... Also, we extend predicate *True* to **F**×**T** and hence the formula *True*(f, t), for each pair of fluent f in **F** and time t in **T**.

Unlike the prime times that are all non-decomposable, an interval can be decomposed into a sequence of sub-intervals/moments/points. However, when intervals are allowed to be arguments of the predicate *True*, we will face the possibility that a fluent f might neither hold true nor hold false <u>throughout</u> some interval t. That is, it may be the case that fluent f holds true with respect to some sub-intervals/moments/points of t but holds false with respect to some other sub-intervals/moments/points of t. As pointed out by Shoham [Sho87b], Bacchus *et al.* [BTK91] and Allen and Ferguson [AlF94], there are two ways we might interpret the negative sentence $\neg$*True*(f, t). In the strong interpretation of negation, $\neg$*True*(f, t) is true if and only if f holds false <u>throughout</u> t, so neither *True*(f, t) nor $\neg$*True*(f, t) would be true in the case that fluent f holds true with respect to some sub-intervals/moments/points of t and also holds false with respect to some other sub-intervals/moments/points of t. Therefore, such a strong interpretation of negation does not preserve *True* as a two-valued predicate any more. In the weak interpretation, $\neg$*True*(f, t) is true if and only if it is not the case that f holds true <u>throughout</u> t, and hence $\neg$*True*(f, t) is true if f changes its truth-value over time t.

In our formalism, we take the weak interpretation of negation as the constraint imposed on the *True* predicate with respect to decomposable intervals, since it seems to be the appropriate interpretation for the standard definition of implication and preserves a simple two-valued logic [AlF94]. The following axiom shows the relation between the truth of a fluent over an interval and its truth over parts of that interval:

(6.1.1) $\exists t_1, t_2(t = t_1 \oplus t_2 \Rightarrow (True(f, t) \Leftrightarrow \forall t'(In(t', t) \Rightarrow True(f, t'))))$

That is, a fluent f holds true with respect to interval t if and only if it holds true with respect to any sub-interval/moment/point of t, that is f holds true <u>throughout</u> t.

It is important to note that, (6.1.1) associates sub-interval/moments/points proposition just to decomposable intervals, rather than non-decomposable moments or points. This is different from Allen's approach [All84], which may lead to some dubious inferences. If one does not limit time t in axiom (6.1.1) to a decomposable interval, one may face the trouble that for any fluent f, it can be shown to hold over any time moment or point unconditionally from this axiom, since there is not any proper sub-time within any given time moment or point, (e.g. any proposition holds true with respect to any moment) [MKP94].

Theorem 6.1.1 If a fluent f holds true with respect to two adjacent times, t_1 and t_2, respectively, then f holds true with respect to the ordered union time, $t_1 \oplus t_2$. That is

$$True(f, t_1) \wedge True(f, t_2) \wedge Meets(t_1, t_2) \Rightarrow True(f, t_1 \oplus t_2).$$

Proof: Suppose it is not. This is we have

$$True(f, t_1) \wedge True(f, t_2) \wedge Meets(t_1, t_2)$$

but

$$\neg True(f, t_1 \oplus t_2).$$

By axiom (6.1.1), we have

$$\exists t'(In(t', t_1 \oplus t_2) \Rightarrow \neg True(f, t')).$$

From the structure of time, there exists a prime time p that satisfies

$$Sub(p, t') \wedge (Sub(p, t_1) \vee Sub(p, t_2)) \wedge \neg True(f, p).$$

This contradicts the fact that $True(f, t_1) \wedge True(f, t_2)$. □

This theorem preserves the extended persistence of a proposition if this proposition holds true over two adjacent times.

Following Galton's suggestion [Gal90], we shall use not(f) to represent the negation of fluent f, to be kept distinct from ordinary sentence-negation, symbolised by "$\neg$". Since we take the weak interpretation of the sentence-negation, we also take the following axioms:

(6.1.2) $True(\text{not}(f), t) \Leftrightarrow \forall t'(Sub(t', t) \Rightarrow \neg True(f, t'))$

(6.1.3) $\neg True(f, t) \Leftrightarrow \exists t'(Sub(t', t) \wedge True(\text{not}(f), t'))$.

Axiom (6.1.2) says that not(f) holds true throughout time t if and only if f does not hold true throughout any sub-interval/internal-point of t. In fact, axiom (6.1.2) is given by Hamblin [Ham71] as the definition of negations. Axiom (6.1.3) means that f does not hold true throughout time t if and only if there exists a sub-element of t throughout which not(f) holds true. Axiom (6.1.3) coincides with the weak interpretation of the sentence-negation.

Intuitively, with respect to any time t, any fluent and its negation can not both hold true. In other words, they are in conflict with each other over any time. However, it is important to note that, for a given fluent, say f, it may be the case that its negation not(f) is not the only fluent that conflicts with it. That is, there may be some fluents other than not(f) that can not be true together with fluent f.

For the purpose of representing conflicts among fluents, $Conf(f_1, f_2)$ is introduced to denote that fluent f_1 is in conflict with fluent f_2:

(6.1.4) $Conf(f_1, f_2) \Leftrightarrow \forall t(\neg True(f_1, t) \vee \neg True(f_2, t))$.

Axiom (6.1.4) says that two fluents, f_1 and f_2 conflict with each other if and only if with respect to any time either f_1 does not hold true or f_2 does not hold true. The "if" part of this axiom is quite clear. To understand the "only if" part, think about this condition: with respect to any time either f_1 does not hold true or f_2 does not hold true. That is, they conflict with each other. This means that there does not exist any time with respect to which both fluents f_1 and

f_2 hold true.

Especially, by axiom (6.1.4) together with the axioms (6.1.2) and (6.1.3), it is straightforward to obtain the following property: for any fluent f

$$Conf(f, \text{not}(f)).$$

This property simply tells that any fluent and its negation are in conflict with each other. Also, axiom (6.1.4) can be generalised to cope with cases involving more than two fluents:

$$Conf(f_1, f_2, \ldots, f_n) \Leftrightarrow \forall t(\neg(True(f_1, t) \wedge True(f_2, t) \wedge \ldots \wedge True(f_n, t))).$$

This says that a set of fluents, $f_1, f_2, \ldots, f_n$, for any n, is in conflict if and only if with respect to any time at least one of the set of fluents does not hold true. The notion of conflict will be used to determine the consistency of states in next section.

It is interesting to note that, in most interval-based temporal theories, there exist two major problematic questions, i.e., the so-called *dividing instant problem* [Van83, Vil94] and the *intermingling problem* [Ham71, Gal96]. The dividing instant problem involves the question of whether time intervals should include their ending-points or not, which can lead to some difficulties in determining the truth values of fluents at the ending-points. The intermingling problem arises when a fluent may change its truth value infinitely often in an interval with a finite duration. This will lead to some difficulties in characterising the relationships between the negation of fluents and the negation of involved sentences.

However, in the time structure described in last chapter, each time is simply defined as an ordered union of prime points/moments that are non-decomposable, there is no definition of ending points for intervals. Therefore, the so-called *dividing instant problem* doesn't exist. In addition, since the time structure, **T**, is the closure of the ordered set of prime times, **P**, by axiom (5.2.5) in chapter 5, we know that **T** satisfies the so-called *finite decomposition* property [Gal96]:

For every fluent f and every time t, there is a decomposition

$$t = p_1 \oplus p_2 \ldots \oplus p_n$$

where $n \geq 1$, such that for $i = 1, 2, \ldots, n$,

$$True(f, t_i) \vee True(not(f), t_i).$$

Then by the result in [Gal96], the time structure, **T**, bypasses the *intermingling problem*.

6.2 States and Situations

Having defined the notion of fluents, the next task is to examine the relationships among fluents and the character of a collection of fluents, and therefore, to study the behavior of the real world. In this section, we are going to introduce the concepts of states and situations. Also we will discuss some properties regarding to these two terms and the relationship between them.

Generally speaking, it is usually impossible to collect all the information for completely describing a world. This means one never knows a world -- instead, one only knows some facts about a world. In fact, we only need to deal with the information in which we are interested. Following Lin and Shoham's idea [LiS94], let **F** be a fixed set of fluent constants, which includes all the fluents in which we are interested, a state can be defined as follows.

Definition 6.2.1 A set S of fluents is a *state* with respect to **F** if there is a subset S' of **F** such that

$$S = \{f \mid f \in S'\} \cup \{\text{not}(f) \mid f \in \mathbf{F} - S'\}$$

Therefore, if S is a state, then for any fluent $f \in$ **F**, either $f \in S$ or not(f)$\in S$. Intuitively, states defined as above completely characterise the belongingness of all fluents in a given set of fluents, such as **F**. The fixed set of fluents, **F**, plays a role similar to Lifschitz's *Frame* predicate [Lif90]. For instance, in a theory of moving and painting blocks, the locations and colors of blocks are the fleunts which we may be interested in, and hence comprise the corresponding fixed set. This definition is motivated by the ideas from the work both in [LiS95] and in [Sha95]. From Lin and Shoham's work, we adopt the idea that considering the interested fluents only by means of a fixed set of fluents. Similar to the treatment of that in [Sha95], we define a state as a set of fluents, keeping it at high level. The main difference between this definition and that given by Lin and Shoham is as follows. Fluents are associated with times to comprise states in [LiS95], whereas we define states at high level, without any association with times. Comparing with that in [Sha95], we use a fixed set of fluent to make the expression of state complete, whereas states are only sets of fluents by Shanahan. The benefits of our interpretation will become clear when a temporal term, *situation*, is introduced later. We shall denote the set of all possible states, as **S**. Elements of **S** will be denoted by (possibly indexed) s. To express the relationship of a given fluent and a given state, a predicate, *Belongs* over **F**×**S** is introduced. Formula *Belongs*(f, s) denotes that fluent f belongs to the state s.

In what follows, without explicit specification of the fixed set **F**, we assume that all the fluents that we will consider must be within a fixed set of fluents, which includes all the fluents in which we may be interested. This assumption is called the *closed-world* assumption. Also, in what follows, for simplicity, we may only use the first part of S in the definition given above to express a state. By default, the negations of the fluents which remain in the closed world should belong to this state. For instance, in the Two Blocks World example in chapter 2, there are four fluents in which we are interested: *OnTable*(A), *OnTable*(B), *On*(A, B), *On*(B, A), comprising the corresponding fixed set of fluents. S_1 = {*OnTable*(A), *OnTable*(B)} can be considered as a state. By default, it implies that the negations of *On*(A, B) and *On*(B, A) belong to this state.

From the definition of states given above, one can not see clearly the relationships among the fluents in a state. For instance, "Block A is on the table" and "Block A is on the top of block B" are two fluents in a block world. By the definition, there may exist a state in which both of these two fluents hold true. However, intuitively it is impossible. Similar to the discussion about the truth values of fluents over a given time in the previous section, some constraints on the states are necessary.

Definition 6.2.2 A state s is called *consistent* if and only if for any fluents, $f_1, f_2, \ldots, f_n$,

$$Belongs(f_1, s) \wedge Belongs(f_2, s) \wedge \ldots \wedge Belongs(f_n, s) \Rightarrow \neg Conf(f_1, f_2, \ldots, f_n)$$

otherwise, s is called an *inconsistent* state.

In what follows, all the states that will be considered are assumed to be consistent. This assumption is quite intuitive. In fact, it is no use talking about a state that is impossible to exist in the real world. Specially, for any fluent, its negation and itself could not both belong to a consistent state. This is also implied from the definition of states. In general, for any consistent state, we impose the following axioms in addition to the definition of predicate *Belongs*.

(6.2.1) $s_1 = s_2 \Leftrightarrow \forall f(Belongs(f, s_1) \Leftrightarrow Belongs(f, s_2))$

Axiom (6.2.1) indicates that two states are equal if and only if they are comprised by the same set of fluents.

Also, from the definition (6.2.2), together with the fact that f and its negative not(f) are in conflict with each other, we have

$$Belongs(f, s) \Leftrightarrow \neg Belongs(\text{not}(f), s),$$

which says that a fluent and its negation can not belong to the same state.

States are terms at high level, independent from time. They are just sets of fluents, with respect to a fixed set of fluents. To associate states with times, a predicate, *S_True* over **S**×**T** is introduced. Formula *S_True*(s, t) represents that s is the state (of the closed world) with respect to time t, imposing:

(6.2.2) $S\_True(s, t) \Leftrightarrow \forall f\,(Belongs(f, s) \Rightarrow True(f, t) \wedge \neg Belongs(f, s) \Rightarrow True(\text{not}(f), t))$

that is, s is the state with respect to time t if and only if every fluent belonging to s holds true with respect to time t, and for every fluent not belonging to s, its negation holds true with respect to time t.

Following the discussion of the interpretation of negation in previous section, the weak interpretation is also used for the predicate *S_True*. This is $\neg S\_True(s, t)$ is true if and only if it is not the case that s holds true throughout t, and hence $\neg S\_True(s, t)$ is true if each of the flunets that belongs to s changes its truth-value over time t. From the definition of predicate *Belongs* and axiom (6.2.2), we have

Theorem 6.2.1 A state s holds true with respect to interval t if and only if it holds true with respect to any sub-interval/moment/point of t, that is each of the fluents that belongs to s holds true throughout t. That is, for any state s and interval t, we have

$$S\_True(s, t) \Leftrightarrow \forall t'(In(t', t) \Rightarrow S\_True(s, t')).$$

Proof: By axiom (6.2.2), we have

$$S\_True(s, t) \Leftrightarrow \forall f\,(Belongs(f, s) \Rightarrow True(f, t) \wedge \neg Belongs(f, s) \Rightarrow True(\text{not}(f), t)).$$

Then by the definition of predicate *Belongs*, the above sentence is equivalent to

$$\forall f\,(Belongs(f, s) \Rightarrow \forall t'(In(t', t) \Rightarrow True(f, t')) \wedge \neg Belongs(f, s) \Rightarrow \forall t'(In(t', t) \Rightarrow True(\text{not}(f), t'))).$$

This is logically equivalent to

$$\forall t'(In(t', t) \Rightarrow \forall f\,(Belongs(f, s) \Rightarrow True(f, t') \wedge \neg Belongs(f, s) \Rightarrow True(\text{not}(f), t'))).$$

Again, by axiom (6.2.2), it is equivalent to

$$\forall t'(In(t', t) \Rightarrow S\_True(s, t')).$$

Thus we have proved that

$$S\_True(s, t) \Leftrightarrow \forall t'(In(t', t) \Rightarrow S\_True(s, t')). \qquad \Box$$

Similarly to the theorem 6.1.1 in the previous section, we have the following theorem, preserving the extended persistence of a state if this state holds true over two adjacent times.

Theorem 6.2.2 If a state s holds true with respect to two adjacent times, t_1 and t_2, respectively, then s holds true with respect to the ordered union time, $t_1 \oplus t_2$. That is

$$S\_True(s, t_1) \wedge S\_True(s, t_2) \wedge Meets(t_1, t_2) \Rightarrow S\_True(s, t_1 \oplus t_2).$$

Proof: Suppose it is not. This is we have

$$S\_True(s, t_1) \wedge True(s, t_2) \wedge Meets(t_1, t_2)$$

but

$$\neg S\_True(s, t_1 \oplus t_2).$$

By theorem 6.2.1, we have

$$\exists t'(In(t', t_1 \oplus t_2) \Rightarrow \neg S\_True(s, t')).$$

From the structure of time, there exists a prime time p that satisfies

$$Sub(p, t') \wedge (Sub(p, t_1) \vee Sub(p, t_2)) \wedge \neg\, S\_True(s, p).$$

This contradicts the fact that $S\_True(s, t_1) \wedge S\_True(s, t_2)$. $\Box$

Although states and times are two different entities, to represent the dynamic world, states should be associated with times. For example, two fluents, "John is in his office", and "John is working" may comprise one state. "John is at home" and "John is sleeping" may comprise another state. The former state describes the fact that John is in his office and working, while the latter tells that John is at home and sleeping. Without additional temporal knowledge, one can not tell anything more than that. However, if these two states are associated with times, for instance, the former state is associated with time T_1, and the latter with time T_2, then one can conclude that John is in his office and working over time T_1 and at home and sleeping over time T_2. Of course, he could not be at home over time T_1, nor at his office over time T_2, because fluents "John is in his office" and "John is at home" are in conflict with each other. Intuitively, with respect to a given time, there must be only one state that holds true. The following theorem states this assertion.

Theorem 6.2.3 For any two states, if their truth values hold with respect to the same time, they must be equal to each other. That is

$$S\_True(s_1, t) \wedge S\_True(s_2, t) \Rightarrow s_1 = s_2.$$

Proof: Let S_1 and S_2 be two states. For a given time T, they satisfy:

$$(*) \qquad S\_True(S_1, T) \wedge S\_True(S_2, T)$$

but

$$S_1 \neq S_2.$$

We are going to prove that $S_1 \neq S_2$ will lead to a contradiction. By axiom (6.2.2), we have

$$\exists f((Belongs(f, S_1) \wedge \neg Belongs(f, S_2)) \vee (\neg Belongs(f, S_1) \wedge Belongs(f, S_2))).$$

Without loss of generality, suppose there exists a fluent F, such that

$$Belongs(F, S_1) \wedge \neg Belongs(F, S_2).$$

Since formula (*) is supposed to be true, by axiom (6.2.2), we have

$True(F, T)$

and at the same time

$True(\text{not}(F), T)$.

This is a contradiction. Therefore we proved that state S_1 must be equal to S_2. □

Theorem 6.2.3 claims that, with respect to a given time, the state of the world is unique. However, there is nothing to stop a temporally contiguous time having the same state. In fact, a state may hold true with respect to different times.

In order to capture the relationships between states and times, we introduce the concept of situations as follows.

Definition 6.2.3 A situation is a pair of a state and a time, representing the state of the world associated with a particular time over which the world holds in that state, denoted as $<s, t>$. All the situations form a collection of pairs of states and times, denoted as **Sit**.

The new set of situations is an additional sort (which is a supersort of sorts, states and times) to the language. Variables for situations are denoted by *sit*, sit_1, sit_2, ..., and constants by *Sit*, Sit_1, Sit_2 .... The following axioms specify the basic characters of situations.

(6.2.3) $\forall sit \exists s \exists t(sit = <s, t> \wedge S\_True(s, t))$

(6.2.4) $sit = <s_1, t_1> \wedge sit = <s_2, t_2> \Rightarrow s_1 = s_2 \wedge t_1 = t_2$

Axiom (6.2.3) says that each situation is formed by a pair of a state and a time, and the state must hold true with respect to the time. Axiom (6.2.4) guarantees that the representation of any situation is unique.

Since a situation is formed by a pair of a state and a time, to represent its arguments individually, the following notions are introduced:

Definition 6.2.4 For any situation $sit = <s, t>$, s is called the *reference state* of situation *sit* and t the *reference time* of situation *sit*, denoted as *State*(*sit*) and *Time*(*sit*), respectively.

In fact, *State* and *Time* may be seen as two functions from the set of situations to the set of states, and the set of times, respectively. The following theorem tells the relationship between these two functions.

Theorem 6.2.4 For any two situations, sit_1 and sit_2, if their reference times are the same, their reference states must be the same as well. That is

$$Time(sit_1) = Time(sit_2) \Rightarrow State(sit_1) = State(sit_2).$$

Proof: By Theorem 6.2.3, we only need to show that

$$S\_True(State(sit_1), Time(sit_1)) \wedge S\_True(State(sit_2), Time(sit_2)).$$

This comes directly from axiom (6.2.3). $\square$

However, the inverse of the implication in this theory does not hold. Thus is if there are two situations with the same reference state, one can not reach the conclusion that these two situations must have the same reference time. In fact, the same state may hold true with respect to various times and therefore form different situations. For instance, suppose that state "raining in London" together with a time "the first of May in 1998" form a situation, denoted by Sit_1 and the same state "raining in London" together with another time "the first of June in 1998" constitute another situation, denoted by Sit_2. It is easy to see that

$$State(Sit_1) = \{Raining\_In\_London\} = State(Sit_2)$$

but

$$Time(Sit_1) = 1/5/98 \neq 1/6/98 = Time(Sit_2).$$

As reflected on the discussion above, one can see the advantage of the strategy we have taken here: defining a state as a set of fluents at high level and making its expression complete, then associating it with a time to form a temporal entity, a situation at low level. This strategy makes the representation of states and situations more expressive and more flexible. A state may holds true with respect to more than one different time, while a situation describes a status of a real (closed) world by way of specifying the particular pair of a state and a time. Based on the fact that within a situation the corresponding state must remain unchanged, directly from the above theorem we have the following corollary:

Corollary 6.2.1 For any two situations, if there exists a time that is a sub-time of both these two situations' reference times, their reference states must be the same. That is:

$$\exists t(Sub(t, Time(sit_1)) \wedge Sub(t, Time(sit_2)) \Rightarrow State(sit_1) = State(sit_2)).$$

This corollary states that if there is a common time in any two situations' reference times, then these two situations must have the same state. To express the relationship between a fluent and a situation, a predicate *Holds* over **F×Sit** is introduced. Formula *Holds*(*f*, *sit*) denotes that fluent *f* holds true in situation *sit*, providing that:

(6.2.5) $Holds(f, sit) \Leftrightarrow True(f, Time(sit)))$

That is a fluent holds true in a situation if and only if it holds true with respect to its reference time.

As stated by the definition of predicate *Belongs*, a fluent and its negation can not belong to the same state. Therefore, they can not both hold in a given situation. The following theorem characterise this intuition.

Theorem 6.2.5 For any fluent *f* and any situation *sit*, the truth value of the negation of *f* holds in situation *sit* if and only if fluent *f* does not hold in situation *sit*. That is:

$$Holds(\text{not}(f), sit) \Leftrightarrow \neg Holds(f, sit).$$

Proof: Suppose *Holds*(not(*f*), *sit*). By axiom (6.2.5), we have

$True(\text{not}(f), Time(sit))$.

Then, it is easy to see from axiom (6.1.2):

$\neg True(f, Time(sit))$.

Again by axiom (6.2.5), we reach the result

$\neg Holds(f, sit)$.

To prove the other side, suppose $\neg Holds(f, sit)$. By axiom (6.2.5), we have

$\neg True(f, Time(sit))$.

By axiom (6.1.3)

$\exists t'(Sub(t', Time(sit)) \wedge True(\text{not}(f), t'))$.

Then by the definition of situations and the corollary (6.2.1), we reach

$Holds(\text{not}(f), sit)$.

Therefore the theorem is proved. □

N.B. For the convenience of expression, in what follows, we shall call situation *sit* a prime situation if its reference time is a prime one.

6.3 Actions and Events

Situations are used to describe the states of the world in a given time. It has little to do with the change of the world. Intuitively, a state of the world will be supposed to keep unchanged until something happens to cause it to change. The things that may cause a situation to change are usually called actions or events. As discussed in the introduction, in this dissertation, actions and events will be treated as different entities. *Actions* are simply some names denoting certain useful and relevant activities. Generally, these actions are relative to some temporal duration. For instance, "*Heating* a tank of water for 10 minutes". Here "*Heating*" is an activity. However, if we only talk about the "*Heating*", it is not clear what will exactly happen as the consequence of this activity. Therefore the consideration of a relative temporal duration for such kind of activities is necessary. That is when we talk about the activity "*Heating*", the temporal duration such as how long the activity will last for is needed to consider. For instance, "*Heating* a tank of water for 10 minutes" may cause the water in the tank boiling.

This discussion leads us to consider that actions are not expressive enough for the formalism. An action together with a temporal duration, named an *action type* is defined as follows.

Definition 6.3.1 An *action type* is a pair of an action and a non-negative real number, denoting a temporal duration, representing the scheme of performing action a with respect to a temporal duration d (d may be zero), denoted as $< a, d>$.

An *action type* is a time independent entity at high level. Its arguments a and d have little to do with any specified time. For instance, at high level, the expression <*Heating*, *10*,> denotes performing action *Heating* for 10 minutes. At low level, it is not specified that where this temporal duration, 10 minutes, should be located. In order to map an action type to the real world, a binary predicate, *Performs* over $\mathbf{A}\times\mathbf{T}$ is introduced. Formula *Performs*(a, t) states that action a performs over time t. Here, time t plays a similar role to the temporal duration in an action type. An action type may perform once, more than once over different times, or may not even perform at all. Each individual performance of an action type over the corresponding time constitutes a temporal entity at low level. This temporal entity is a mirror of an action type at low level. We call such an entity an event. It can be defined formally as follows.

Definition 6.3.2 An *event* is a pair of an action and a time entity, $<a, t>$, meaning that action a performs over time t. The space of events will be denoted by **E**.

The new sort of events (which is a supersort of sorts, actions and times) is also needed for the language. Variables for events are denoted by e, e_1, e_2, ..., and constants by E, E_1, E_2 .... Events are low level entities. They present the occurrences of action types over actual times. For example, the performance of action type, "Striking for 24 hours" over the whole day of the 16th of September 1994 constitutes an event, denoted as E = <Striking, 16/09/94>. Performing an action type over different times composes different events. For example, the performance of action type, "Striking for 24 hours" over times: the whole day of the 16th of September 1994 and the whole day of the 15th of November 1994 constitute two different events, e_1 = <Striking, 16/09/94> and e_2 = <Striking, 15/11/94>. This example also shows that an action type may perform more than once. However, some action types may not perform at all. Consider the sentence "A man raising up a very heavy stone (e.g. 1000 kg) over his head using 10 minutes". The action type can be expressed as <Raising_the_stone,10>. Of course, by the common sense knowledge, this action type can never perform successfully.

As most researchers have described [McD82, All84, KoS86 etc.], the events defined above satisfies the nature of anti- homogeneity: If an event occurs over an interval i, then it does not occur over any subinterval of i, as it would not yet be completed. The following axiom preserves that if an action performs over two adjacent times respectively, then it performs over the ordered union of the two times:

(6.3.1) $Performs(a, t_1) \wedge Performs(a, t_2) \wedge Meets(t_1, t_2) \Rightarrow Performs(a, t_1 \oplus t_2)$

and on the other hand, if an action performs over a decomposable interval, then it performs over any proper subtimes of it:

(6.3.2) $\exists t_1,t_2(t = t_1 \oplus t_2 \Rightarrow (Performs(a, t) \Rightarrow \forall t'(In(t', t) \Rightarrow Performs(a, t'))))$.

These two axioms are important for the discussion about concurrent events (We are not going to discuss this issue in detail in this work). It is worth noting that these axioms do not conflict with the above nature of anti-homogeneity, because for any action, with respect to different time entities, the different events are defined. For instance, suppose action *Heating* was performed over time T_1 that represents the time from 12:00 to 12:04 on 16/8/96 and T_2

represents the time from 12:04 to 12:10 on 16/8/96 respectively. Then we can write:

$Performs(Heating, T_1) \wedge Performs(Heating, T_2)$.

Also

$Performs(Heating, T_1 \oplus T_2)$,

Since we have

$Meets(T_1, T_2)$.

In fact, there were three different events happened. First, heating the tank of water for 4 minutes, $E_1 = <Heating, T_1>$, the second is heating the tank of water for 6 minutes, $E_2 = <Heating, T_2>$, and the last is heating the tank of water for 10 minutes, $E_3 = <Heating, T_1 \oplus T_2>$. As the consequence of event E_3, the water in the tank should be boiling, while as the consequence of the event E_1, the temperature of the water in the tank may reach 40 degree.

The uniqueness of the representation of an event is guaranteed by:

(6.3.3) $\forall e \exists a \exists t(e = <a, t> \wedge Performs(a, t))$

(6.3.4) $e = <a_1, t_1> \wedge e = <a_2, t_2> \Rightarrow a_1 = a_2 \wedge t_1 = t_2$

These axioms also show that an event is a derived structure from an action and a time, denoting the actual performance of a given action over a certain time at low level. In literature, the two terms, "action" and "event" are used in many different senses by various researchers; sometimes they are used interchangeably.

Analogous to the definitions of the reference state and the reference time of a given situation, the following definition is given.

Definition 6.3.3 For event $e = <a, t>$, we shall call a and t the *reference action* and the *reference time* of e, denoted as $a = Action(e)$ and $t = Time(e)$, respectively.

Axiom (6.3.3) also states that for any event, its reference action must perform over its reference time. That is if we know that event e exists, then we can conclude that formula *Performs*(*Action*(e), *Time*(e)) is true. For instance, in the striking example above, i.e., event e_1 = <*Striking*, 16/09/94>, the corresponding reference action is *Striking* and the reference time is the 16th of September 1994.

6.4 Pre-conditions

In some cases, the execution of a given action may be constrained by some pre-conditions that must hold. For example, in the Blocks World, to move block A onto block B, both blocks A and B must be clear. Sometimes for the same action, if the action types are different, their pre-conditions may be different as well. For instance, "Driving for two hours" and "Driving for six hours" are two different action types, which have the same action name "Driving". However, the pre-conditions of these two action types may be different. For example, to drive for two hours (about 70 miles/hour), the car should have at least 10 liters of petrol, while driving for six hours, the car must have at least 30 liters of petrol.

For general treatment, the pre-condition of an action, a, denoted by $PreC(a)$, is a set (possibly empty) of fluents. A binary predicate, *PreCon*, over $\mathbf{A}\times\mathbf{S}$ is introduced to represent whether the precondition of an action holds in a state. Formula $PreCon(a, s)$ states that the set of fluents, $PreC(a)$ must satisfy in state s, i.e. all fluents in set $PreC(a)$ belong to state s, for action a to be effective, imposing

(6.4.1) $PreCon(a, s) \Leftrightarrow \forall f(f \in PreC(a) \Rightarrow Belongs(f, s))$

Set $PreC(a)$ is the pre-condition of performing action a. If $PreC(a)$ is little to do with the duration of the performance, then the expression described above is useful. However, in our formalism, since the temporal duration of performing an action plays an important role, it is necessary to modify the corresponding expression as follows. Similarly, for action type <a, d>, the pre-condition is denoted by $PreCT(a, d)$. It is also a set (possibly empty) of fluents. A predicate, *PreConT*, over $\mathbf{A}\times\mathbf{R}\times\mathbf{S}$ is introduced to represent whether the pre-condition of an action type holds in a state. Formula $PreConT(a, d, s)$ states that the set of fluents,

$PreCT(a,d)$ must satisfy in state s, i.e. all fluents in set $PreCT(a, d)$ belong to state s, for action type $<a, d>$ to be effective, imposing:

(6.4.2) $PreConT(a, d, s) \Leftrightarrow \forall f(f \in PreCT(a, d) \Rightarrow Belongs(f, s))$.

Pre-conditions are the premise of executions of actions. However, if an action is known to execute in a situation, the corresponding pre-conditions must hold in that situation. That is

(6.4.3) $\forall a \forall t(Performs(a, t) \Rightarrow \exists t' \exists s \exists d(S\_True(s, t') \wedge Dur(t) = d$
$\wedge PreConT(a, d, s) \wedge Meets(t', t)))$

this says that for any action a and any time t if action a occurs over time t, then there must exist a time t' and a state s such that state s holds over time t', pre-condition $PreCT(a, d)$ holds true in state s and time t' meets time t.

The expression of pre-conditions of actions/action types is not essential for the results of performing actions/action types. However, for the qualification problem, which specifies every possible relevant condition, it is important. The representation of pre-conditions proposed in this section will enable us to deal with qualification problem by way of minimising the *PreConT* predicate, which will guarantee that the known preconditions of each action are the only preconditions. However, as mentioned in chapter 4, since the qualification problem has little to do with the temporal domain, we are not going to look the qualification problem in any more detail here. In what follows, without confusion, if there is no explicit specification for the pre-conditions of the execution of actions/action types, by default it is assumed that the pre-conditions of the execution of actions/action types are satisfied.

6.5 Causality

Causation plays an important role in reasoning about actions and change. As mentioned in the introduction, when we reason about change over time, causation provides an implicit preference: in view of common-sense we prefer sequences of world states in which one world state leads causally in terms of the occurrence of some certain event to the next, rather than sequences in which one world state follows another at random and without causal connections. For example, in the car-stolen problem, we prefer sequences in which the thieve steal the car to sequences in which the car disappears without explanation, although not necessarily to sequences in which the car remains untouched. In this section, we are going to show how to use the formalism proposed here to represent common-sense causality at both high level and low level.

As argued by Gooday and Galton [Gal96], many action and change problems encountered in AI may be formulated without any low level temporal reference. In particular, most common-sense causal relationships are actually expressed at high level, making no reference to any specific times. Although these are relationships involving temporal information, they are time independent and therefore hold for all times. For example, we might know that if a kettle is filled with cold water and switched on, it will start to boil after five minutes. This causal knowledge applies whenever the kettle is switched on, and specifies that the performance of the action type, i.e. $<Switch\_On, 0>$, causes the world to change from state $\{Water\_Is\_Cold\}$ to state $\{Water\_Is\_Boiling\}$ 5 minutes later, where *Switch_On* represents the action name and the temporal duration of performing this action is supposed to be zero. Of course, a common-sense statement such as this does not really mean that the water will always be boiling after being switched on, regardless of other possible actions. Rather it means that it will be boiling in the absence of other events occurring during the 5 minutes, such as countermanding switching or electrical power failures, etc.

To capture this common sense meaning, a predicate *Causes* is introduced as follows:

(6.5.1) $Causes(s_1, <a, d_a>, d, s_2) \Leftrightarrow$

$$\forall t_1,t_a,t(\ S\_True(s_1, t_1) \wedge Dur(t_a) = d_a \wedge Performs(a, t_a) \wedge Dur(t) = d$$

$$\wedge\ Meets(t_1, t_a) \wedge Meets(t_1, t)$$

$$\wedge\ \forall t'\forall a'(Sub(t',t) \wedge a' \neq a \Rightarrow \neg Performs(a',t')))$$

$$\Rightarrow \exists t_2(Meets(t, t_2) \wedge S\_True(s_2, t_2)))$$

Predicate *Causes* relates an initial state s_1, an action type $<a, d_a>$, a temporal duration d and the result state s_2. It states that starting from state s_1, after a time duration d, the performance of the action type $<a, d_a>$ will change the world into state s_2 as long as no other disturbing action has occurred. Argument d in the *Causes* predicate plays an important role. For the same initial state and the same action type, if the argument d takes a different value, the resulting state may be different as well. For instance, in the kettle example, the corresponding causal axiom can be written as:

(Ke1) *Causes*({*Water_Is_Cold*}, <*Switch_On*, *0*>, *5*, {*Water_Is_Boiling*}).

However, if the argument d takes a less value, for instance 2 minutes, then the resulting state may not be {*Water_Is_Boiling*}. Hence, the causal predicate depends not only on the initial state and the action type, but also on the temporal duration, d. Here, a question may arise with respect to this argument, d, i.e., if the duration d is greater than zero, it may correspond to four possible time entities at low level: t, $p \oplus t$, $t \oplus p$ and $p_1 \oplus t \oplus p_2$, where $Dur(t) = d$. Therefore, although t and $p_1 \oplus t \oplus p_2$ denote an open interval and a closed one respectively, and $p \oplus t$ and $t \oplus p$ represent two intervals that are left and right closed respectively and hence represent four different times, they all have the same temporal duration.

For the purpose of reasoning, since what we are interested in is the consequence of performing an action rather than the starting time of performing the action, we shall only consider two of the four possible cases, i.e., t and $t \oplus p$, where $Dur(t) = d$. However, if it is necessary, the extension of the following discussion to the all four cases is straightforward. In general, the alternative of t or $t \oplus p$ with respect to duration d will depend on the information provided. For the expressive purpose, a notion of *duration type* is introduced.

Definition 6.5.1 A *duration type* is a pair of a temporal duration and an element *co* that belongs to the set {*close*, *open*}, denoted as <*d*, *co*>. Notation <*d*, *close*> represents that the duration type can only be mapped to a time entity that is right closed, while <*d*, *open*> represents that the duration type can only be mapped to a time entity that is right open.

To see the motivation of this treatment, let's consider the following two examples. First, consider the kettle example. As it is known, axiom (Ke1) can be seen as a causal law, that is starting from a state in which the water in the kettle is cold, switch on the kettle, then 5 minutes later it will result in a state in which the water is boiling. In this case, the duration type is not important. Whether the duration type <*5*, *close*> or <*5*, *open*> being used is not crucial, since in fact, it will not make any difference for the result. However, in some cases, the duration type is crucial. To see this, consider the earlier example of throwing a ball into the air: The motion may be described qualitatively by the use of two intervals, interval i (suppose its duration is 8 seconds) where the ball is going up, and interval j where the ball is going down. According to classical physics, there is a point p at which the ball is stationary. Suppose the interval i is right open. Here the temporal duration $d = Dur(i) = Dur(i \oplus p)$ relates two time entities: interval i and interval $i \oplus p$. However, these two time entities have different properties: over time interval i the proposition *Ball_Going_Up* holds true, but over interval $i \oplus p$ *Ball_Going_Up* does not holds true. Now, how to express the (delayed) effects of the action (type) of throwing the ball?

In [GLR91], Gelfond *et al* proposed an approach of using an action called *Wait* to deal with the delay between an action and its effect, where the duration of *Wait* equals the time delay. For instance, in their approach, one may use $Result(S_0, Throw+Wait)$ to represent the situation 8 seconds after throwing the ball. However, in this result situation, is the velocity zero or not? The answer is not unique. In fact, there are two situations, one is the situation where the ball is at the stationary point, and another is the situation immediately after the stationary point. Both of them satisfy that the *Wait* action lasts for 8 seconds. Therefore, Gelfond *et al*'s approach seems unable to distinguish these two different delayed effects. Ma *et al* also examined the complicated temporal relations between actions and their effects [KMP97, MKP97 etc]. For example, in [MKP97], they developed another approach of using a predicate $Changes(sit_1, e, t, sit_2)$ to denote the proposition that, immediately after time t, event e changes situation sit_1 into situation sit_2. In their approach, they do not provide any scheme to deal with the point-sensitive cases, like the throwing ball example. In this case, we have to use duration type $<d, open>$ to represent the temporal information of the knowledge that "the ball is going up", which specifies the information at the right end of the period of time. In order to represent this kind of causal knowledge, instead of using axiom (6.5.1), the causal axiom should be rewritten as:

(6.5.1)' $Causes(s_1, <a, d_a>, <d, co>, s_2) \Leftrightarrow$

$$\forall t_1,t_a,t(\ S\_True(s_1, t_1) \wedge Dur(t_a) = d_a \wedge Performs(a, t_a)$$
$$\wedge\ Dur(t) = d \wedge REnd(t) = co \wedge Meets(t_1, t_a) \wedge Meets(t_1, t)$$
$$\wedge \forall t' \forall a'(Sub(t',t) \wedge a' \neq a \Rightarrow \neg Performs(a',t')))$$

$$\Rightarrow \exists t_2(Meets(t, t_2) \wedge S\_True(s_2, t_2))).$$

Then one may easily express the causal knowledge in throwing a ball example as:

$Causes(s_1, <Throw, 0>, <8, open>, s_2)$

$Causes(s_1, <Throw, 0>, <8, close>, s_2')$

where in s_2 the ball's velocity is zero, and in s_2' it is not.

Axioms (6.5.1) and (6.5.1)' are similar. The only difference between these two axioms is that the corresponding duration type is employed in axiom (6.5.1)', whereas it not in axiom (6.5.1). Taking into account of the fact that, in most cases, it is not necessary to specify the value of *co* in a duration type, since the difference of the value of *co* in a duration type normally does not make any difference in the knowledge representation. In what follows, if there is no explicit specification for the value of *co* in the expression of a duration type, by default it is assumed that the value of *co* is closed. In axiom (6.5.1), the duration type $<d, co>$ (simple written *d*) follows this default assumption. In most cases, this assumption is explicit enough, although for some cases, such as the throwing a ball example, in which the type of change is point-sensitive, i.e., the change happens at a time point, this assumption may cause problems. This kind of change may be called *point-sensitive* change (the corresponding effects are called *point-sensitive effects*). In what follows, axiom (6.5.1) is adopted for interpreting the predicate *Causes*. If it is necessary, axiom (6.5.1)' can be used in stead of axiom (6.5.1).

Predicate *Causes* is independent of time. It expresses high level knowledge in the similar sense of Gooday and Galton [GoG96]. In [GoG96], a transition schema is defined as an ordered pair $<<S_1, S_2>>$, where $S_1$and S_2 are states. The schema is intended to represent the transition from S_1 to S_2, i.e., an event such that S_1 holds true immediately before it and S_2 holds true immediately after it, with no state holding in between. Using the predicate *Causes*, as an example, we may express the knowledge about the kettle boiling as:

(Ke1) $Causes(\{Water\_Is\_Cold\}, <Switch\_On, 0>, 5, \{Water\_Is\_Boiling\})$

This high level knowledge makes no reference to any specific times. However, it denotes a high level causality that allows us to deduce relationships at low-level, where actual time instances are specified. For instance, given temporal information about times related as in figure 6.1, where a time point is represented by a double-lined arrow:

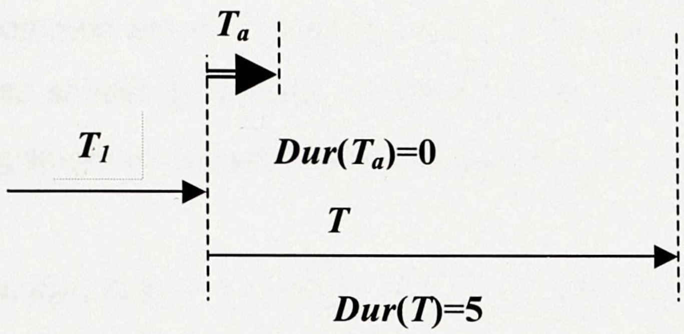

Figure 6.1

That is:

(Ke2) $Meets(T_1, T_a) \wedge Meets(T_1, T) \wedge Dur(T_a) = 0 \wedge Dur(T) = 5$

In this case, if we know that the initial state $\{Water\_Is\_Cold\}$ holds over time T_1:

(Ke3) $S\_True(\{Water\_Is\_Cold\}, T_1)$

and action *Switch_On* performs over time T_a:

(Ke4) $Performs(Switch\_On, T_a)$,

and we know that no other action happens during time T:

(Ke5) $Sub(t', T) \wedge a' \neq Switch\_On \Rightarrow \neg Performs(a', t')$

then by (6.5.1), we can deduce the "effect" that there exists a time T_2 such that:

(Ke6) $Meets(T, T_2) \wedge S\_True(\{Water\_Is\_Boiling\}, T_2)$

Intuitively, according to the predicate *Causes*, the causal result should be unique with respect to the initial state, action type and the temporal duration. For instance, again, consider the kettle example, in common sense, 5 minutes after performing the action type $<Switch\_On, 0>$, the resulting state should be $\{Water\_Is\_Boiling\}$. There are no other possibilities. We impose the following axiom to characterize this statement:

(6.5.2) $Causes(s_1, <a, d_a>, d, s_2) \wedge Causes(s_1, <a, d_a>, d, s_2') \Rightarrow s_2 = s_2'$

N.B. The specification of duration types, for instance, here by default, d = $<d, closed>$, guarantees the unique result. Without the specification of the duration types, with respect to the same causal expression, as shown above, one can not use predicate *Causes* to represent the relative causal knowledge.

Formula $Causes(s_1, <a, d_a>, d, s_2)$ characterised above denotes a high level causal law that demonstrates the temporal duration and order in which, the performance of an action causes the world to change from one state to another. It makes no reference to any actual times at low level. For convenience of expression at low level, a predicate *E_Causes* is introduced, which expresses an instance of the causal law. The formula $E\_Causes(sit_1, e, t, sit_2)$ has the meaning that the world changes from situation sit_1 into situation sit_2 after time t, and the change is caused by the occurrence of event e, where no other disturbing events occur.

$$
\begin{aligned}
&(6.5.3)\ E\_Causes(sit_1, e, t, sit_2)\\
&\quad \Leftrightarrow \quad Meets(Time(sit_1), Time(e)) \wedge Meets(Time(sit_1), t) \wedge Meets(t, Time(sit_2))\\
&\qquad\qquad \wedge \forall t'\forall a'(Sub(t', t) \wedge a' \neq Action(e) \Rightarrow \neg Performs(a', t'))
\end{aligned}
$$

The function *REnd* can be used to specify the right end status of time t and relate it to the corresponding duration type. By default, in axiom (6.5.3) the value for $REnd(t)$ is *closed.* To represent the causal change in the throw-ball example at low level, the expression of function *REnd* is necessary. Suppose Sit_0 is the original situation in which the ball is at hand. E_{Throw}

denotes the event that performs action *Throw* at time T_{Throw}, where $Dur(T_{Throw}) = 1$. T represents the time after the beginning of the event and $Dur(T) = 8$. Sit_1 is the result situation. Then we have

$E\_Causes(Sit_0, E_{Throw}, T, Sit_1)$

where $REnd(T) = open \wedge Holds(Ball\_Stationary, Sit_1)$

If $REnd(T) = close$, then fluent *Ball_Stationary* will not hold in the result situation. In next chapter, the full representation for this example will be provided.

Similarly, in what follows, if there is no explicit specification for the value of function $REnd(t)$, that will mean the value of $REnd(t)$ is not important and by default it is *closed*. Axioms (6.5.1) and (6.5.3) together characterise an important causal relation at distinct levels. Since the causal result of an event will only come true in the absence of other unknown disturbing events, as in other approaches an assumption must be made: *unexpected events do not occur*. This assumption is specified axiomatically rather than being built into the semantic model. It guarantees the mapping from the high level causal knowledge to the low level causality. In fact, in the case where the corresponding knowledge is available, one can use the following procedure to deduce low level changes from high level causal laws:

Theorem 6.5.1 Given causal axiom:

(CA) $Causes(s_1, <a, d_a>, d, s_2)$

and the knowledge that there exist situation sit_1, event e and time t such that:

(TC) $Meets(Time(sit_1), Time(e)) \wedge Meets(Time(sit_1), t)$

together with observation:

(OB1) $State(sit_1) = s_1$

(OB2) $Action(e) = a$

(OB3) $Dur(Time(e)) = d_a$

(OB4) $Dur(t) = d$

(OB5) $Performs(a, Time(e))$

and constraint on disturbing actions:

(ND) $Sub(t', t) \wedge a' \neq a \Rightarrow \neg Performs(a', t')$.

Then we can deduce that there exists a situation sit_2, such that:

$$E\_Causes(sit_1, e, t, sit_2) \wedge State(sit_2) = s_2.$$

This procedure can be obtained directly from axioms (6.5.1) and (6.5.3).

Since situations do not have to be primes, the result situations caused by the performance of an event may not be unique. However, the relative conclusion about the result reference states can be proved:

Theorem 6.5.2 For any situation sit_1, if there exist an event, e and a time t such that

$$Meets(Time(sit_1), Time(e)) \wedge Meets(Time(sit_1), t),$$

then the occurrence of event e in situation sit_1 after time t will lead to a unique state. That is

$$E\_Causes(sit_1, e, t, sit_2) \wedge E\_Causes(sit_1, e, t, sit_2') \Rightarrow State(sit_2) = State(sit_2').$$

Proof: Suppose that the condition

$$E\_Causes(sit_1, e, t, sit_2) \wedge E\_Causes(sit_1, e, t, sit_2')$$

is satisfied. By axiom (6.5.3), we have

$$Meets(Time(sit_1), Time(e)) \wedge Meets(Time(sit_1), t) \wedge Meets(t, Time(sit_2))$$

and

$Meets(t, Time(sit_2'))$.

Then, by property (Pro5.1) in chapter 5:

(Pro5.1) $Meets(t, t_1) \wedge Meets(t, t_2) \Rightarrow Starts(t_1, t_2) \vee States(t_2, t_1) \vee Equal(t_1, t_2)$

we have

$$Starts(Time(sit_2), Time(sit_2')) \vee Starts(Time(sit_2'), Time(sit_2)) \vee Equal(Time(sit_2), Time(sit_2')).$$

Therefore, by Corollary 6.2.1, we reach the conclusion that $State(sit_2) = State(sit_2')$. □

In most existing versions of the situation calculus or event calculus, e.g., [McH69], [Sch90], [LiS92,94], [PiR93,95], [MiS94], [Sha95] and [KoS97], the effect of an event is represented by the effect immediately after the occurrence of the event (i.e., Case (B) in figure 6.2). Some other cases such as delayed effect (Case (E) in figure 6.2) and synchronous effects (Case (A, C, D) in figure 6.2) are neglected. Axiom (6.5.3) presented above allows other cases besides that of an effect immediately following the cause and provides an explicit causal relation between the effect and its causal event. In fact, the temporal constraint imposed in (6.5.3) is consistent with the so-called (most) *general temporal constraint* (*GTC*) (see [McD82], [All84], [Sho88], [TeT95] and [KPM98]), which guarantees the common-sense assertion that "*the beginning of the effect cannot precede the beginning of the cause*". There are in total 5 possible qualitative temporal relationships between $Time(sit_1)$, $Time(e)$, t and $Time(sit_2)$ that satisfy (6.5.3). These are illustrated in Figure 6.2, including:

Case (A) where the end of event e coincides with the end of the effect sit_2. In the extreme circumstance, when time t is a point with duration of zero, the duration of event e equals to the duration of effect sit_2; that is, the effect becomes true immediately after the beginning of its causal event, and only holds true while the event is in progress;

Case (B) where the effect becomes true immediately after the end of the event and remains true for some time (i.e., $Time(sit_2)$) after the event;

Case (C) where the effect becomes true during the progress of the event and remains true for some time after the event;

Case (D) where the effect only hold over some time during the progress of the event (Similar to case (A), cases (C) and (D) include the extreme circumstance where the effect becomes true immediately after the beginning of the event, i.e., time t is a point with zero duration);

Case (E) where there is a time delay, i.e., t_d, between the event and its effect.

For the notational convenience, in what follows we will use *Immediate Sequential Effects* to represent the temporal relationship described in case (B), *Delayed Sequential Effects* to denote that in case (E) and *Coincident Effects* to express that in the other cases.

Theorem 6.5.1 provides a procedure to deduce a conclusion from given knowledge and observations. Since $E\_Causes(sit_1, e, t, sit_2)$ only tells the change from situation sit_1 to sit_2. It does not guarantee the persistence of fluents over time t. The deduction about the information over time t has to be done by means of reasoning by default. Therefore, in this case, one has to handle the frame problem. Since the causal axioms (6.5.1), (6.5.1)' and (6.5.3) are crucial in this formalism, it is natural to think about using causal minimization technique to tackle the frame problem. Also, it can be proved that, by extending the standard circumscription policy to include time, the corresponding frame problem with respect to the formalism proposed here can be dealt with by means of the state-based minimization technique. This issue will be discussed in details in next section.

6.6 Treatment of the frame problem

Any approach to common sense temporal reasoning must address the frame problem: how best to specify everything that does not change as a result of an action. This problem has the extra dimension of time in the formalism presented here. Not only do we have to specify the values of fluents that are completely unaffected by an action, but also we have to specify the values of fluents with respect to time elements where they are unaffected. That is we have to specify times over which fluents change their truth values as the result of performing an action, as well as time periods over which they persist

Based on the situation calculus, there are mainly two ways for solving this problem: monotonic and nonmonotonic. Schubert [Sch90] and Reiter [Rei91] propose monotonic approaches to this problem based on the idea of "explanation closure". For instance, Reiter provides a solution to the frame problem, using successor state axioms. Each such axiom provides a complete characterization of a fluent's truth value in the next state *Result*(a, s). Also, a number of solutions based on the use of nonmonotonic formalisms have been proposed. Those techniques include *Chronological Minimization* [Kau86, Sho86, Lif86], *Causal Minimization* [Hau87, Lif87] and *State-Based Minimization* [McC86, Bak91, Sha97]. For instance, Baker provides a nonmonotonic solution to the frame problem, using situation calculus and circumscription (*State-Based Minimization*). As mentioned in chapter 4, these techniques have been fully discussed and extended by Shanahan [Sha97]. In what follows, based on the techniques of the causal minimization and the state-based minimization, two extended circumscription theories will be developed for the treatment of the corresponding frame problem in TSTC. For simplicity, the two theories are still named as state-based minimization and causal minimization respectively.

6.6.1 Causal Minimization

In TSTC, since predicates *Causes* and *E_Causes* are employed to represent the causal change, the causal minimization technique seems to be the best choice to deal with the frame problem. Causal minimization [Hau87, Lif87] is a nonmonotonic approach to the frame problem by means of circumscribing *Causes*. Circumscribing *Causes* means that we prefer sequences of world states in which actions have fewer effects. In causal minimization, every change has to be caused by actions, and all changes that can not be explained in terms of the effects of known actions are minimised. As discussed in chapter 4, to deal with the frame problem more efficiently, several versions of causal minimization have been developed. [StM94, Sha97, ect.]. However, none of them can be adopted here straightforwardly, because a richer temporal ontology has been imbedded in the new formalism. In this section, we will borrow and extend the main idea and technique of the causal minimization to deal with the frame problem.

Since the predicate *Causes* is at high level, on one hand, one formula of *Causes* may map to more than one corresponding low level formula of *E_Causes*; on the other hand, one formula *E_Causes* can only trace back to one high level causal law. Also the low level expression of the causation is an instance of the high level causal law. Considering the fact that the high level causal laws are provided by predicate *Causes* in the domain description, based on the principle of separation for the causal minimization technique, the minimization should focus on the causal predicate *Causes*. To deal with the frame problem, how to express and use frame axioms is an important issue. Normally, the frame axioms should specify the fluents that persist their truth value. In the conventional causal minimization formalism [Lif87], the frame axiom can be expressed as:

(FA) $\quad \neg\exists v(Causes(f, a, v) \Rightarrow (Holds(f, sit) \Leftrightarrow Holds(f, Results(a, sit)))$.

This states that if there does not exist truth value v (true or false) such that performing action a causes fluent f to take on truth value v, then the truth value for fluent f will persist from situation *sit* to *Results*(a, *sit*).

However, in the proposed temporal state transition calculus, TSTC, since there isn't any fluent as its argument in predicate *Causes*, it is not possible to represent a corresponding frame axiom like (FA) directly. Following Shanahan's expression [Sha97], the corresponding frame axiom can be written as:

(6.6.1) $\neg Affects(f, s_1, <a, d_a>, d, s_2) \Rightarrow Belongs(f, s_1) \Leftrightarrow Belongs(f, s_2)$.

This axiom states that if fluent f is not affected, after time duration d, by the occurrence of action type $<a, d_a>$, then fluent f will persist its truth value from state s_1 to state s_2. It insists that all change is caused by actions/events. *Affects* is not a predicate symbol of the language, but is abbreviation defined as follows.

$$\begin{aligned} Affects(f, s_1, <a, d_a>, d, s_2) \equiv_{\text{def}} \; & Causes(s_1, <a, d_a>, d, s_2) \wedge PreconT(a, d_a, s_1) \\ & \wedge (Belongs(f, s_1) \wedge \neg Belongs(f, s_2) \\ & \vee \neg Belongs(f, s_1) \wedge Belongs(f, s_2)). \end{aligned}$$

The circumscription policy to overcome the frame problem is to minimize predicate *Causes*, allowing predicate *Holds* to vary. In this way, one can obtain the expected minimised model, and get rid of all unexpected models.

Let D be the conjunction of all domain constraint axioms, O the conjunction of all observation facts, CA the conjunction of all the causal axioms, and FA the frame axiom (axiom (6.6.1)). For the purpose of convenience, let Σ be the conjunction of D, O, CA and FA. Then the general circumscription theory is

$$\text{CIRC}(\Sigma; Causes; \; Holds).$$

This theory will provide a solution to the corresponding frame problem. Applications of this circumscription theory will be shown in next chapter. Also, the treatment of relative ramification problem will be discussed by way of applying this theory to some classical examples.

6.6.2 State-based Minimization

In the Temporal State Transition Calculus, $Causes(s_1, <a, d_a>, d, s_2)$ is introduced to represent high level causation (changing), where time duration d_a and d play important roles. In other words, from a given state, the result state does not only depend on what an action applies, but also depend on how long the action performs and how long since the action has performed. Comparing with the conventional Situation Calculus, in which the result situation is only dependent on the action and the initial situation, the formula introduced here is more expressive, and, of course, more complicated.

Differing from the case in previous section, since the low level causal predicate *E_Causes* is an instance of high level causal law, in this case, the minimization should focus on the low level causal predicate *E_Causes*. Since predicate *E_Causes* has got some extra arguments compared to the *Result* function as used in conventional Situation Calculus, the predicate *Ab* is needed to be re-defined for the sake of applying the circumscription method. Conventional predicate $Ab(f, a, sit)$ simply states that, as the result of performing action a starting from situation sit, fluent f will change its truth value. In TSTC, $Ab(f, sit_1, e, t, sit_2)$ shall be used instead of $Ab(f, a, sit)$. Formula $Ab(f, sit_1, e, t, sit_2)$ states that fluent f is abnormal with respect to a tuple (sit_1, e, t, sit_2). Since there is an extra argument t in predicate *Ab*, the standard *common-sense law of inertia*:

$$\neg Ab(f, a, sit) \Rightarrow Holds(f, sit) \Leftrightarrow Holds(f, Results(a, sit))$$

is not applicable to the formalism proposed here. It needs to be revised in order to deal with the corresponding frame problem correctly.

In the situation calculus, if we say that fluent f is abnormal, it means fluent f changes its truth value from the original situation to the corresponding result situation when an action is performed in the original situation. Also, since in the situation calculus, there are no situations defined during the time over which an action is executing, one can not assert anything about the world during this time. Hence, there is a unique interpretation for the persistent expression:

$$Holds(f, sit) \Leftrightarrow Holds(f, Results(a, sit)).$$

That is the truth value of fluent f persists from situation sit to the next situation $Results(a, sit)$. However, in TSTC, there is a time t between the original situation sit_1 and the corresponding result situation sit_2 with respect to event e. Therefore, even if fluent f holds the same truth value in both sit_1 and sit_2, we still can not claim that fluent f persists its truth value from situation sit_1 to situation sit_2. In fact, there are two cases: fluent f does not change its truth value over time t, and fluent f does change its truth value over time t. This suggests two options to revise the standard *common-sense law of inertia*:

(Option1) $\neg Ab(f, sit_1, e, t, sit_2) \Rightarrow Holds(f, sit_1) \Leftrightarrow Holds(f, sit_2)$;

(Option2) $\neg Ab(f, sit_1, e, t, sit_2) \Rightarrow Holds(f, sit_1) \wedge True(f, t) \Leftrightarrow Holds(f, sit_2)$.

According to the first option, fluent f is not abnormal with respect to the tuple (sit_1, e, t, sit_2) implies that fluent f holds true in situation sit_1 if and only if it holds true in situation sit_2. It says nothing about the truth value of fluent f over time t. The second option tells that fluent f is not abnormal with respect to the tuple (sit_1, e, t, sit_2) implies that fluent f holds true in situation sit_1 and over time t if and only if it holds true in situation sit_2. This implicitly states that the truth value of fluent f persists from situation sit_1 to situation sit_2. This explanation seems closer to the standard common-sense law of inertia. In situation calculus, it is expressed as axiom (CLISC), which says that the value of a fluent persists from one situation to the next situation unless something is abnormal. However, as mentioned above, one of the main differences between the situation calculus and the TSTC is that TSTC allows some situations existing between a starting situation and the resulting situation, whereas the situation calculus does not. So although in situation calculus, there is no problem to define the common sense law of inertia as axiom (CLISC), in TSTC it does have some trouble. In fact, if the second option is adopted, some fluents that change their truth values between the starting situation sit_1 and the resulting situation sit_2 and then change back to their original truth value in sit_2 may be treated as abnormal. This seems not intuitive. In common sense, we say a fluent is not abnormal with respect to two situations, sit_1 and sit_2. It means that the truth value of that fluent in sit_1 is as same as in sit_2. It is not necessary to put any constraint between sit_1 and sit_2. Therefore the first option is the better choice for the revision of the standard common sense law of inertia in TSTC.

In what follows, the first option is adopted as the revised *common sense law of inertia*:

(6.6.2) $\neg Ab(f, sit_1, e, t, sit_2) \Rightarrow Holds(f, sit_1) \Leftrightarrow Holds(f, sit_2)$.

Since sit_1 and sit_2 in formula $Ab(f, sit_1, e, t, sit_2)$ are not necessarily prime ones, with respect to f, sit_1, e and t, one may get different result situations. Following Baker's notation [Bak91], $<f, sit_1, e, t, sit_2>$ is used to denote the abnormality, which means that f is abnormal with respect to sit_1, e, t and sit_2. To distinguish the abnormalities, the following axiom is imposed:

(6.6.3) $<f_1, sit_1, e_1, t_1, sit_2> = <f_2, sit_3, e_2, t_2, sit_4> \Leftrightarrow$

$$f_1 = f_2 \wedge State(sit_1) = State(sit_3) \wedge e_1 = e_2 \wedge t_1 = t_2 \wedge State(sit_2) = State(sit_4).$$

The intuitive meaning of axiom (6.6.2) is that fluent f has the same truth value in both situation sit_1 and situation sit_2 unless something is abnormal. It is important to note that, fluent f is not abnormal with respect to sit_1, e, t and sit_2 just ensures that f has the same truth value over $Time(sit_1)$ and $Time(sit_2)$, but does not preserve that the truth value of fluent f persists throughout time t. Hence, the following two questions still remain:

- Over time t, does the truth value of fluent f persist?

- Over time t, when does fluent f start to change its truth value?

Since formula $E\_Causes(sit_1, e, t, sit_2)$ tells us not just what a state will hold in the result situation, but also the difference among the result situations as the variation of argument t, the circumscription policy should be circumscribing Ab with both sit_2 and t vary. The main idea of this policy is, if time argument t and situation sit_2 are allowed vary, the result situations will range over all possible situations as the variation of argument t and sit_2.

Similarly to the treatment of the previous section, let D be the conjunction of all domain constraint axioms, which includes axiom (6.6.3), O the conjunction of all observation facts, CA the conjunction of all the causal axioms, and C the *common-sense law of inertia* (axiom (6.6.2)). For the purpose of convenient, let Σ be the conjunction of D, O, CA and C. Then the general circumscription theory is

$$\mathrm{CIRC}(\Sigma; Ab; t, sit_2).$$

This circumscription theory will provide a solution to the corresponding frame problem. Here

sit_2 denotes the result situation. Since this circumscription theory follows the idea of Baker's approach, one of the important things, which is needed to be mentioned, is the existence-of-situations axiom should be included in the collection D.

N.B. In the trivial case (case (B) in Figure 6.2), since the argument t in predicate E_Causes is constrained to be equal to the reference time of event e, the circumscription theory is equivalent to

$$\mathrm{CIRC}(\Sigma; Ab; sit_2).$$

The application of this theory will be shown in next chapter by applying it to some classical examples.

CHAPTER 7
APPLICATIONS

The temporal relationship between an event and its effect is complex. The TSTC combines the advantages of temporal logic by the way of allowing explicit time representation and that of state-based approaches such as the situation calculus which can easily represent the worlds that remain static except when the agent acts, and where nothing important happens while actions are being executed.

By means of illustrating the application of the logic and the techniques for dealing with the corresponding frame problem proposed here, we now present formalizations of some of the standard problems from literature in reasoning about action and change gathered in [San94] and some examples relative to the possible qualitative temporal relationships between *Time*(sit_1), *Time*(e), t and *Time*(sit_2) that have been discussed in previous chapter. The classification for this sort of relationships will follow the discussion in section 6.5.

7.1 Immediate Sequential Effects

In this section, we are going to formalise a class of problems which are mostly variants of the Yale Shooting Problem [HaM87], a basic AI test case for reasoning about action and change. Each scenario involves a very simple domain where a single problem is presented and thus it is not the best vehicle for demonstrating the generality of our approach. However, it does establish a certain baseline and allows us to support our claims as the expressiveness and naturalness of the temporal state transition calculus. This section will begin with the basic Yale Shooting Problem.

7.1.1 Yale Shooting Problem (YSP)

Consider the following story:

Originally Fred is alive and the gun is loaded. If the loaded gun is fired at Fred, it will make him dead. Question: from the original situation, after waiting for some time the gun is fired to Fred, is Fred still alive?

Intuitively, the issue is that while the load of the gun ought to persist through the waiting (and then the shooting succeeds in killing Fred), Fred's "aliveness" ought not to persist after the shooting (indeed, cannot, without risk of inconsistency) although it should persist through the waiting. However, there might be "non-standard" models where the gun somehow becomes unloaded, in which case the shooting would fail to kill Fred. Our aim is to formalise this problem in the logical system presented here which will derive the expected conclusion that Fred is dead.

In this scenario, there are two fluents, *Loaded* and *Alive*, and two actions, *Wait* and *Shoot*. To express this example at high level, the following two action types are used:

<*Wait*, 60>: waiting for 60 seconds;

<*Shoot*, 1>: firing the gun which, without loss of generality, can be assumed to take 1 second.

If we have the knowledge that during the action *Wait*, there are no disturbing actions and there is no change, then we can assume the following axiom (YCA1), which tells us that from a initial state performing action *Wait* for 60 seconds will not cause any change. Axiom (YCA2) is a causal axiom: Fred dies after a *Shoot* action so long as the *Shoot* action is successful, i.e., the gun is loaded at the shooting time. In this case, the frame problem does not arise, since we have complete knowledge about the scenario. Suppose fluents *Loaded* and *Alive* belong to the initial state s_0. We have

(YCA1) $Causes(s_0, <Wait, 60>, 60, s_0)$

(YCA2) $Causes(s_0, <Shoot, 1>, 1, s_1) \Rightarrow \neg Belongs(Alive, s_1)$

At this level, the fact that performing action *Wait* for 60 seconds will not change anything is expressed. Also if fluents *Loaded* and *Alive* belong to state s_0, then from this state, performing action *Shoot* for 1 second will cause the world to change to state s_1, in which fluent *Alive* dose not hold if no other disturbing events occur. There are no references to any specified times. In what follows, we will show how to use Theorem (6.5.1) to deduce the intuitive conclusion that Fred is dead after shooting.

To express the scenario at low level, two events are defined as:

E_{Wait}: waiting over time T_{Wait}, where $Dur(T_{Wait}) = 60$;

E_{Shoot}: firing to Fred at time T_{Shoot}, where $Dur(T_{Shoot}) = 1$.

In what follows, Sit_0 is used to denote the original situation, where two facts are observed: Fred is alive and the gun is loaded, Sit_1 the result situation of performing the first action type, e.g. <*Wait*, 60>, and Sit_2 the result situation of performing the second action type, e.g. <*Shoot*, 1>.

(YOB1) $Holds(Loaded, Sit_0)$

(YOB2) $Holds(Alive, Sit_0)$.

Also, one event is observed starting from Sit_0:

(YOB3) $Performs(Wait, T_{Wait})$

together with the knowledge about the temporal constraint

(YTC1) $Meets(Time(Sit_0), Time(E_{Wait})) \wedge Time(E_{Wait}) = T_{Wait}$

and the constraint on disturbing actions:

(YND1) $Sub(t, T_{Wait}) \wedge a \neq Wait \Rightarrow \neg Performs(a, t)$

Then, by Theorem 6.5.1 from axioms (YCA1), (YOB1) - (YOB3), (YTC1) and (YND1), we can deduce that there exists a situation, Sit_1, such that:

(YEC1) $E\_Causes(Sit_0, E_{Wait}, T_{Wait}, Sit_1)$

$\wedge\ State(Sit_1) = State(Sit_0)$

$\wedge\ Meets(T_{Wait}, Time(Sit_1))$

This is a pseudo change. In fact, E_{Wait} causes nothing to change. It is a special event. In general, if an event occurs, it should cause some change. (YEC1) guarantees the assumption that we have the complete knowledge about the scenario. Without this assumption, some ambiguity will arise, which will be discussed later. From situation Sit_1, since we have (YEC1), together with the knowledge:

(YOB4) $Performs(Shoot, T_{Shoot})$

(YTC2) $Meets(Time(Sit_1), Time(E_{Shoot})) \wedge Time(E_{Shoot}) = T_{Shoot}$

(YND2) $Sub(t, T_{Shoot}) \wedge a \neq Shoot \Rightarrow \neg Performs(a, t)$

again by Theorem 6.5.1, from axioms (YCA2), (YEC1), (YOB1) - (YOB4), (YTC2) and (YND2), we can deduce the low level causal change, that is there exists a situation, Sit_2, such that:

(YEC2) $E\_Causes(Sit_1, E_{Shoot}, T_{Shoot}, Sit_2)$

$\wedge\ \neg Holds(Alive, Sit_2)$

$\wedge\ Meets(T_{Shoot}, Time(Sit_2))$

Therefore, the expected result has been deduced. That is Fred is dead after shooting. At this low level, the scenario is expressed by assigning each state and each action type a specified time. Also the temporal relationships amongst situations Sit_0, Sit_1 and Sit_2, events E_{Wait} and E_{Shoot} have been explicitly expressed as shown in figure 7.1.

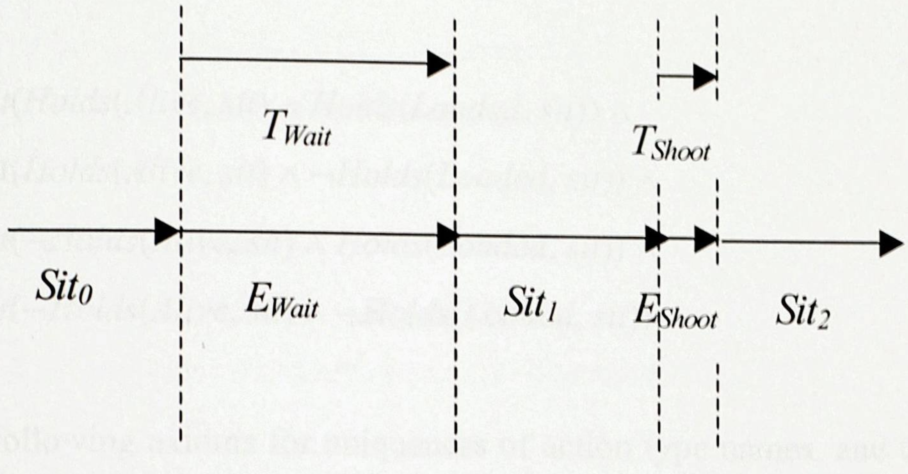

Figure 7.1

In normal cases, as one expects, the answer should be that Fred is dead in Sit_2. However, to obtain the conclusion by the former method, one needs the knowledge that there are no effects of performing action *Wait*. This is an important assumption for the deduction. The reason will be clear later. Also it will be clear what will happen if this sort of information is not available. However, this kind of knowledge may not be in the knowledge base. This means that some ambiguity about the action *Wait* may arise during the period of performing action *Wait*, which would lead to an abnormal case: in the resulting situation of performing action *Shooting* for one second, Sit_2, Fred is still alive. This is an unexpected result. The reason may be the lack of knowledge about the action *Waiting*. There is no guarantee that no change should be caused by action *Waiting*. For instance, the gun might be unloaded while waiting. In this case, the frame problem may arise. In what follows, we will deal with the corresponding frame problem by way of applying the two techniques: the state-based minimization and causal minimization respectively.

State-based Minimization Since this example is the trivial case in our category for representing temporal relationships between events and their effects, that is the effect of an event becomes true immediately after the end of the event and remains true for some time (i.e., *Time*(sit_2)) after the event, in the circumscription policy, the result situation is allowed to vary when the predicate *Ab* is circumscribed.

In addition, a key component of the state-based minimisation technique is to take the existence-of-situations axiom as a domain constraint axiom [Sha97]. The situation existence axiom for the addressed question is:

(YS1) $\exists sit(Holds(Alive, sit) \wedge Holds(Loaded, sit)) \wedge$

$\exists sit(Holds(Alive, sit) \wedge \neg Holds(Loaded, sit)) \wedge$

$\exists sit(\neg Holds(Alive, sit) \wedge Holds(Loaded, sit)) \wedge$

$\exists sit(\neg Holds(Alive, sit) \wedge \neg Holds(Loaded, sit))$

Also, the following axioms for uniqueness of action type names, and the domain closure for fluents are needed:

(YS2) $\forall a,d(<a, d> = <Wait, 60> \vee <a, d> = <Shoot, 1>)$

(YS3) $\forall f(f = Alive \vee f = Loaded)$

Proposition 7.1.1 Let Σ be the conjunction of (YS1) - (YS3), (YOB1) - (YOB4), (YTC1), (YTC2), (YND1), (YND2), axiom (YCA2) and axioms (6.6.2) - (6.6.3) then

$$CIRC(\Sigma; Ab; sit_2) \models (YEC2)$$

Proof: Let M be any model of CIRC(Σ; Ab; sit_2). Such a model consists of various domains: the domain of times $|M|_t$, the domain of situations $|M|_{sit}$, the domain of action types $|M|_{<a, d>}$, the domain of events $|M|_e$ and the domain of fluents $|M|_f$, as well as for the constants:

$M[[T_{Wait}]] \in |M|_t$, $M[[T_{Shoot}]] \in |M|_t$,

$M[[Sit_0]] \in |M|_{sit}$,

$M[[<Wait, 60>]] \in |M|_{<a, d>}$, $M[[<Shoot, 1>]] \in |M|_{<a, d>}$,

$M[[<Wait, Time(E_{Wait})>]] \in |M|_e$, $M[[<Shoot, Time(E_{Shoot})>]] \in |M|_e$,

$M[[Alive]] \in |M|_f$, $M[[Loaded]] \in |M|_f$,

interpretations for the relations:

$M[[Holds]] \subseteq |M|_f \times |M|_{sit}$,

$$M[[Ab]] \subseteq |M|_f \times |M|_{sit} \times |M|_e \times |M|_t \times |M|_{sit},$$

$$M[[E\_Causes]] \subseteq |M|_{sit} \times |M|_e \times |M|_t \times |M|_{sit}.$$

To show that M must satisfy (YEC2), axioms about uniqueness of names for fluents, events (i.e., (YS2), (YS3)) are needed. Without loss of generality, we can take

$$|M|_f = \{Alive, Loaded\}$$

$$|M|_{<a,\,d>} = \{<Wait, 60>, <Shoot, 1>\}$$

with *Alive*, *Loaded* and *Wait*, *Shoot* interpreted as themselves.

In this case the argument *t* in predicate *E_Causes* has been constrained to be equal to *Time*(*e*). Therefore the result situation has been limited. There are eight cases to consider, corresponding to the four possible combinations of the two fluents, *Alive* and *Loaded*, with the two events E_{Wait} and E_{Shoot}. For each of these cases, we ask which fluent persists and which does not. We shall consider just one of these cases, since the rest are analogous.

Let $Sit' \in |M|_{sit}$ be any situation in which fluents *Alive* and *Loaded* hold:

$$\langle Alive, Sit'\rangle \in M[[Holds]],$$

$$\langle Loaded, Sit'\rangle \in M[[Holds]],$$

and let *Sit''* be the result of performing action *Shoot* for one second starting from *Sit'*. This event can be denoted as E_{Shoot}. The temporal relationships among *Sit'*, E_{Shoot} and *Sit''* are described as:

$$Meets(Time(Sit'), Time(E_{Shoot})) \wedge Meets(Time(E_{Shoot}), Time(Sit'')).$$

Then from axioms (YCA2) and (YND2), by Theorem 6.5.1 we have

$$\neg Holds(Alive, Sit'').$$

However, we do not know whether *Loaded* holds in *Sit''*. Suppose it does not. That is we have the abnormality:

$$\langle Loaded, Sit', E_{Shoot}, Time(E_{Shoot}), Sit''\rangle \in M[[Ab]].$$

Now let M' be a model that is identical to M except that M' satisfies:

$$M'[[Ab]] = M[[Ab]] - \langle Loaded, Sit', E_{Shoot}, Time(E_{Shoot}), Sit''\rangle.$$

It is clear that model M' exists because

- Σ does not constrain what happens to fluent *Loaded* when action *Shoot* performs from situation *Sit'*, and
- The existence-of-situation axiom (YS1) tells us that in every model there exists a situation in which fluent *Loaded* holds true and *Alive* does not hold true.

This would eliminate abnormality $\langle Loaded, Sit', E_{Shoot}, Time(E_{Shoot}), Sit''\rangle$ without introducing any new abnormalities (since there are only two fluents). Therefore, M' is smaller than M with respect to the circumscription policy. This contradicts the assumption that M is the model of the circumscription.

As already mentioned, the other seven cases are analogous. Therefore the proposition is proved. $\square$

As pointed out by Baker [Bak91], there are two axioms that play crucial roles in state-based minimization. One is the domain closure axiom, like (YS3) in this example. Without this kind of domain constraint, some mysterious nameless fluents may bother us. The other one is the existence-of-situations axiom, like (YS1). This axiom "violates the spirit of the nonmonotonic enterprise" [Bak91]. If it forced us to write out by hand an axiom like (YS1) for each domain we were interested in, state-based-minimization would hardly constitute a very satisfactory solution to the frame problem. To overcome this problem, a general method for generating such an axiom has been developed by Baker, and then extended by Shanahan. See [Sha97] for the detailed discussion.

Causal Minimization To see the causal minimization technique works in TSTC as well, this approach is also applied to treat the Yale shooting problem.

For the purpose of causal minimisation, the Yale shooting scenario is represented as follows, where the constants are interpreted as before.

(YCM1) $Causes(s_0, \langle Shoot, 1\rangle, 1, s_1) \Rightarrow \neg Belongs(Alive, s_1)$

(YCM2) $Holds(Loaded, Sit_0)$

(YCM3) $Holds(Alive, Sit_0)$

(YCM4) $Performs(Wait, T_{Wait})$

(YCM5) $Performs(Shoot, T_{Shoot})$

(YCM6) $Meets(Time(Sit_0), Time(E_{Wait})) \wedge Meets(Time(E_{Wait}), Time(Sit_1))$
$\wedge\ Meets(Time(Sit_1), Time(E_{Shoot})) \wedge Meets(Time(E_{Shoot}), Time(Sit_2))$
$\wedge\ Time(E_{Wait}) = T_{Wait} \wedge Time(E_{Shoot}) = T_{Shoot}$

(YCM7) $Sub(t, T_{Wait}) \wedge a \neq Wait \Rightarrow \neg Performs(a, t)$

(YCM8) $Sub(t, T_{Shoot}) \wedge a \neq Shoot \Rightarrow \neg Performs(a, t)$

(YCM9) $\forall a,d(\langle a, d\rangle = \langle Wait, 60\rangle \vee \langle a, d\rangle = \langle Shoot, 1\rangle)$

(YCM10) $\forall f(f = Alive \vee f = Loaded)$

Axiom (YCM1) is the only causal axiom. Axioms (YCM2) and (YCM5) are observation facts. Axiom (YCM6) is the temporal constraint among the original situation, Sit_0, the event E_{Wait}, its result situation Sit_1, the event E_{Shoot} and the resulting situation of E_{Shoot}, Sit_2. Axioms (YCM7) and (YCM8) are constraints on disturbing actions. Axioms (YCM9) and (YCM10) are axioms for uniqueness of event names, and the domain closure for fluents. The following proposition states that the causal minimisation can deal correctly with the Yale shooting problem.

Proposition 7.1.2 Let Σ be the conjunction of (YCM1) - (YCM10) and axiom (6.6.1) then

$$\text{CIRC}(\Sigma; \textit{Cause}; \textit{Holds}) \quad \models \quad \text{(YEC2)}$$

Proof: First, we show that all models of the circumscription theory satisfy the following sentence.

(7.1.1) $Causes(s_1, \langle a, d_a \rangle, d, s_2) \Leftrightarrow \ Belongs(Alive, s_1) \wedge Belongs(Loaded, s_1)$
$\wedge\, a = Shoot \wedge d_a = 1 \wedge d = 1 \wedge \neg Belongs(Alive, s_2)$.

From (YCM1) - (YCM10), by theorem (6.5.1) we see that all models have to satisfy the if half of the sentence. It remains to prove the only-if half. Suppose M is a model of the circumscription that does not satisfy the only-if half of the sentence. Now consider any model M' of Σ that meets the following criteria:

- M' agrees with M on the interpretation of everything except *Causes* and *Holds*
- M' $\models$ (7.1.1)
- M' $\models Holds(Alive, Sit_0)$
- M' $\models Holds(Loaded, Sit_0)$
- M' $\models Holds(f, sit_2)$ if and only if,
    - M' $\models Holds(f, sit_1)$ and M' $\models \neg Affects(f, State(sit_1), \langle a, d_a \rangle, d, State(sit_2))$, or
    - M' $\models \neg Holds(f, sit_1)$ and M' $\models Affects(f, State(sit_1), \langle a, d_a \rangle, d, State(sit_2))$

It is clear that such an M' exists, and that the extension of *Causes* in M' is a strict subset of that in M. So M' is as small as M with respect to the circumscription policy, but M is not as small as M'. Then M is not a model of the circumscription. This leads to a contradiction. Therefore, the only-if half of (7.1.1) must be satisfied by all models of the circumscription. This agrees that every change has an cause and this kind of causes is minimized.

Given sentence (7.1.1), by (6.6.1) we can obtain,

$Holds(Alive, Sit_1)$,

$Holds(Loaded, Sit_1)$,

and therefore,

$\neg Holds(Alive, Sit_2)$.

This proves the proposition. $\square$

In the remainder of this section, we use the notations and axioms above for several of the core problems of the Sandewall test suite [San94]. Each of these problems deals with prediction from a given situation, sometimes predicting previous facts (retrodicting). For each problem, we provide an axiomatization of the problem description. The role of Fred is taken by a turkey in Sandewall collection, hence we make the corresponding change in terminology.

7.1.2 Stanford Murder Mystery (SMM)

In this variant, the turkey is initially alive, and after the actions *Shoot* and then *Wait* are performed in succession (the opposite of the Yale shooting order), the turkey is dead. Suppose Sit_0 is used to denote the original situation, where one fact is observed: Fred is alive, Sit_1 the result situation of performing the first action type, e.g. <*Shoot*, 1>, and Sit_2 the result situation of performing the second action type, e.g. <*Wait*, 60>. We are going to determine when the turkey died, and whether or not the gun was originally loaded. This is a temporal explanation problem, i.e. one of the methods to deal with such a problem is to predict aspects of the situation backwards in time.

The following axioms are employed to describe the SMM scenario:

(SMM1) $Holds(Alive, Sit_0)$

(SMM2) $\neg Holds(Alive, Sit_2)$

(SMM3) $Performs(Shoot, T_{Shoot})$

(SMM4) $Performs(Wait, T_{Wait})$

(SMM5) $Meets(Time(Sit_0), Time(E_{Shoot})) \wedge Meets(Time(E_{Shoot}), Time(Sit_1))$
$\wedge Meets(Time(Sit_1), Time(E_{Wait})) \wedge Meets(Time(E_{Wait}), Time(Sit_2))$
$\wedge Time(E_{Wait}) = T_{Wait} \wedge Time(E_{Shoot}) = T_{Shoot}$

(SMM6) $Causes(s_0, <Shoot, 1>, 1, s_1) \Rightarrow \neg Belongs(Alive, s_1)$

(YCM7) $Sub(t, T_{Shoot}) \wedge a \neq Shoot \Rightarrow \neg Performs(a, t)$

(YCM8) $Sub(t, T_{Wait}) \wedge a \neq Wait \Rightarrow \neg Performs(a, t)$

(YCM9) $\forall a,d(<a, d> = <Shoot, 1> \vee <a, d> = <Wait, 60>)$

(SMM10) $\forall f(f = Alive \vee f = Loaded)$

The desired explanation is simply that the gun was loaded in the original situation Sit_0, and the turkey died as a result of the shooting. Causal minimization works in this case because the explanation doesn't demand the introduction of any new effects of actions.

Proposition 7.1.3 Let Σ be the conjunction of (SMM1) to (SMM10) with axioms (6.6.1) then

$$\text{CIRC}(\Sigma;\ Causes;\ Holds) \models Holds(Loaded, Sit_0) \wedge \neg Holds(Alive, Sit_1)$$

Proof: It is similar to the proof of **Proposition 7.1.2**. $\square$

The state-based minimization also can give the right answer. As pointed out by Baker [Bak91] in addition to the result situation, the situation constants Sit_0 and Sit_2 also must be allowed to vary during the circumscription of *Ab*. Then the corresponding circumscription policy for this problem should be circumscribing *Ab* with all three situations Sit_0, Sit_1 and Sit_2 vary.

7.1.3 Russian Turkey Shoot (RTS)

This variant is slightly different from the basic Yale shooting scenario. Initially the gun is loaded and the turkey is alive. Instead of a waiting action, the gun's chamber is spun first, then shooting action is performed. Following the spin action, the value of the *Loaded* fluent is unknown. Then it is not expected to be able to conclude whether or not the turkey is alive after the action sequence *Spin* and then *Shoot*. The issue of this variant is the ability of the representation to deal with uncertainty in the effects of actions. Since there are no predicates or functions to cope with uncertainty in TSTC, following Kartha and Lifschitz [KaL94], a new predicate *Releases* is introduced. Formula $Releases(f, s, \langle a, d_a \rangle, d, s')$ states that it is not known whether or not the fluent f holds in the result state s' after a time duration d, the action type $\langle a, d_a \rangle$ is performed from the state s as long as no other disturbing action has occurred. Similarly, for convenience of expression at low level, a predicate *E_Releases* is needed. The formula $E\_Releasess(f, sit_1, e, t, sit_2)$ has the meaning that the world changes from situation sit_1 into situation sit_2 after time t, where no other disturbing events occur, however, it is unknown whether or not fluent f holds in situation sit_2.

Following the expression of the basic Yale shooting scenario, in stead of action type $\langle Wait, 60 \rangle$ and event E_{Wait}, we have

<*Spin*, 2>: spinning the gun for 2 seconds,

and

E_{Spin}: spinning the gun over time T_{Spin}, where $Dur(T_{Spin}) = 2$.

Then the Russian turkey shooting scenario can be represented as follows.

The temporal constraints among the relative situations and events can be described as

(RTS1) $$Meets(Time(Sit_0), Time(E_{Spin})) \wedge Meets(Time(E_{Spin}), Time(Sit_1)) \wedge Meets(Time(Sit_1), Time(E_{Shoot})) \wedge Meets(Time(E_{Shoot}), Time(Sit_2)) \wedge Time(E_{Spin}) = T_{Spin} \wedge Time(E_{Shoot}) = T_{Shoot}$$

Originally, the gun is loaded and the turkey is alive in situation Sit_0:

(RTS2) $Holds(Alive, Sit_0)$

(RTS3) $Holds(Loaded, Sit_0)$

From this situation, event E_{Spin} occurs, that is action *Spin* is performed for 2 seconds,

(RTS4) $Performs(Spin, T_{Spin})$

It is known that following the spin action, the value of the *Loaded* fluent is unknown.

(RTS5) $E\_Releases(Loaded, Sit_0, E_{Spin}, T_{Spin}, Sit_1)$

Together with the knowledge that there are no disturbing events:

(RTS6) $Sub(t, T_{Spin}) \wedge a \neq Spin \Rightarrow \neg Performs(a, t)$

(RTS7) $Sub(t, T_{Shoot}) \wedge a \neq Shoot \Rightarrow \neg Performs(a, t)$

(RTS8) $\forall a,d(<a, d> = <Spin, 2> \vee <a, d> = <Shoot, 1>)$

One can conclude that it is not known whether or not the gun is loaded in situation Sit_1. This is the main issue of this scenario, which says that the action *Spin* is a possible cause of the gun becoming unloaded. Therefore, from situation Sit_1 we can not conclude that after shooting the turkey is not alive. The only explanation is:

If the gun is loaded in situation Sit_1, by the causal law:

(RTS9) $Causes(s_0, <Shoot, 1>, 1, s_1) \Rightarrow \neg Belongs(Alive, s_1)$

we can conclude that the turkey is not alive in situation Sit_2:

$E\_Causes(Sit_1, E_{Shoot}, T_{Shoot}, Sit_2)$

where $\neg Holds(Alive, Sit_2)$.

Otherwise, if the gun is unloaded in situation Sit_1, by the axiom (6.6.1), we obtain that the truth value of fluent *Alive* will persist, that is

$Holds(Alive, Sit_2)$.

We see that the TSTC can handle the uncertainty in the effects of events, together with the introduction of a new predicate *Releases*.

7.2 Coincident Effects

In the previous section, some benchmark examples in reasoning about action and change are represented using the formalism provided in chapter 6. All the examples have one common character that is the effect of an event becomes true immediately after the end of the event and remains true for some time (i.e., $Time(sit_2)$) after the event. However, the temporal relationship between an event and it effects is not as simple as this. It is in fact quite complex and interesting. To see the expressive power of TSTC, in the rest of this chapter we will demonstrate some other cases, which have been categorized in last chapter. First, let us examine an example in which the end of event *e* coincides with the end of the effect sit_2.

Consider the following simple example:

> *There is a flashlight with a button for flashing it. The light is on only when the button is being pressed down and is still being pressed down. Suppose that from a situation where the flashlight is off, the button is pressed and lasts for 5 seconds, then is released.*

In this example, there are only one fluent *Flashlight_On* and two actions *Press_Button* and *Release_Button* involved. The expected result should be that the flashlight is on for the 5 seconds while the button is pressed and then off after releasing the button.

In addition to the names of fluent and action, the following notations are employed for representing this scenario:

<*Press_Button*, *d*>:	action type, press the button for duration *d*,
<*Press_Button*, 5>:	action type, press the button for 5 seconds,
<*Release_Button*, 0>:	action type, release the button at a time point,
E_{Press_Button}:	event, press the button over time T_{Press_Button}, where $Dur(T_{Press\_Button}) = 5$,
$E_{Release\_Button}$:	event, release the button over time $T_{Release\_Button}$, where $Dur(T_{Release\_Button}) = 0$.

This is the extreme circumstance of case (A) in the category for representing temporal relationship between an event and it effects given in chapter 6. The time *t* is a point with duration of zero, the duration of event *e* equals to the duration of effect sit_2; that is, the effect becomes true immediately after the beginning of its causal event, and only holds true while the event is in progress. Another issue of this example is to specify the end point of a time entity with duration. It will be clear through the expressing of this example why it is necessary to involve this issue. The high level causation for this example can be expressed as:

(FB1) $Causes(s_0, <Press\_Button, d>, 0, s_1) \wedge Dur(d) > 0$
$\Rightarrow Belongs(Light\_On, s_1)$

(FB2) $Causes(s_0, <Release\_Button, 0>, 0, s_1)$
$\Rightarrow \neg Belongs(Light\_On, s_1)$

Here the duration between the beginning of performing action *Press_Button* and the beginning of the result situation is zero. It is a trivial case to our general definition of duration type in chapter 6. Since its relative time entity is a time point, it must be right closed. Therefore, the reference time of the corresponding result situation must be left open in order to satisfy the temporal constraints.

To express this knowledge at low level, suppose that in the original situation, Sit_0, fluent *Light_On* does not hold, and from Sit_0 action *Press_Button* is performed over time T_{Press_Button}, leading to the result situation Sit_1 after time point T. Then action *Release_Button* is performed over time $T_{Release\_Button}$, leading to the result situation Sit_2 after time T_1. The temporal relationships among Sit_0, T, T_{Press_Button}, Sit_1, $T_{Release\_Button}$, T_1 and Sit_2 are interpreted in figure 7.2. Then we have

(FB3) $\neg Holds(Light\_On, Sit_0)$

(FB4) $Performs(Press\_Button, T_{Press\_Button})$

(FB5) $Meets(Time(Sit_0), T) \wedge Meets(T, Time(Sit_1)) \wedge Meets(Time(Sit_0), T_{Press\_Button})$

(FB6) $Dur(T) = 0 \wedge Dur(T_{Press\_Button}) = 5$

(FB7) $Performs(Release\_Button, T_{Release\_Button})$

(FB8) $Meets(Time(Sit_1), T_1) \wedge Meets(T_1, Time(Sit_2)) \wedge Meets(Time(Sit_1), T_{Release\_Button})$

$\wedge\ Meets(T_{Press\_Button}, T_1) \wedge Meets(T_{Release\_Button}, Time(Sit_2))$

(FB9) $Dur(T_1) = 0 \wedge Dur(T_{Release\_Button}) = 0$

(FB10) $Sub(t, T_{Press\_Button}) \wedge a \neq Press\_Button \Rightarrow \neg Performs(a, t)$

(FB11) $Sub(t, T_{Release\_Button}) \wedge a \neq Release\_Button \Rightarrow \neg Performs(a, t)$

(FB12) $\forall a,d(<a, d> = <Press\_Button, d> \wedge (d < 5 \vee d = 5) \vee <a, d> = <Release\_Button, 0>)$

(FB13) $\forall f(f = Light\_On)$

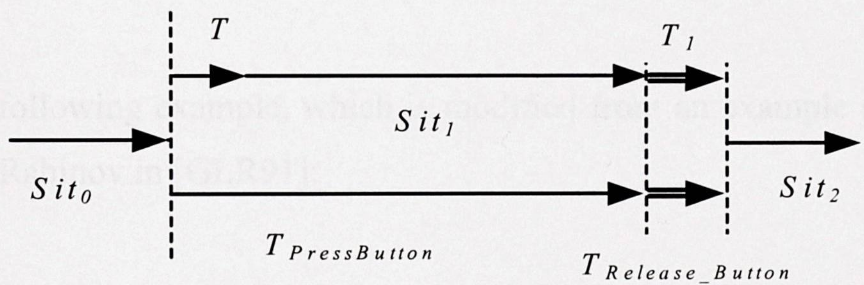

Figure 7. 2

The desired explanation is simply that the light is on in situation Sit_1 as the consequence of performing action *Press_Button* over time T_{Press_Button}, and then the light becomes off in situation Sit_2 as the consequence of performing action *Release_Button* over time $T_{Release\_Button}$. That is

(RFB1) $E\_Causes(Sit_0, E_{Press\_Button}, T, Sit_1) \wedge$

$Holds(Light\_On, Sit_1) \wedge Dur(Time(Sit_1)) = 5$

(RFB2) $E\_Causes(Sit_1, E_{Release\_Button}, T_1, Sit_2) \wedge$

$\neg Holds(Light\_On, Sit_2)$

Proposition 7.2 Let Σ be the conjunction of (FB1) to (FB13) with axioms (6.6.1) then

CIRC(Σ; *Causes*; *Holds*) $\models$ (RFB1) ∧ (RFB2).

Proof: A proof in the same style as that of Proposition 7.1.2 is easily constructed. □

By now, we correctly represent the fact that the effect becomes true immediately after the beginning of the event, and explicitly express the temporal constraints of the relative temporal entities.

7.3 Delayed Sequential Effects

One of the important abilities for formalisms to reasoning about actions and change is to represent the phenomenum that there is a time delay between an event and its effects. In this section, we will show how to deal with this issue using TSTC.

7.3.1 The Expression of the Scenario

Consider the following example, which is modified from an example presented by Gelfond, Lifschitz and Rabinov in [GLR91]:

> *25 seconds after a pedestrian starts pressing the button at the crosswalk, the pedestrian crossing light turns to yellow from red, and after another 5 seconds it turns to green.*

Three fluents to describe the state of the pedestrian crossing light are used:

RedOn: the light at the crosswalk is red;
YellowOn: the light at the crosswalk is yellow;
GreenOn: the light at the crosswalk is green.

Also, one action type is employed:

<*PressButton*, 0>: pressing the button punctually that takes no duration;

and one event:

$E_{PressButton}$: pressing the button at time point T_{Press}.

Again, the issue of specifying the end point of a time entity with duration is involved. We assume that in any situation there is exactly one of the three fluents that holds true:

(PCL1) $Holds(RedOn, sit) \wedge \neg Holds(YellowOn, sit) \wedge \neg Holds(GreenOn, sit)$
$\vee \neg Holds(RedOn, sit) \wedge Holds(YellowOn, sit) \wedge \neg Holds(GreenOn, sit)$
$\vee \neg Holds(RedOn, sit) \wedge \neg Holds(YellowOn, sit) \wedge Holds(GreenOn, sit)$

The high level causal axioms in this case are as below:

(PCL2) $Belongs(RedOn, s_0) \wedge Causes(s_0, \langle Pressbutton, 0\rangle, 25, s_1)$
$\Rightarrow Belongs(YellowOn, s_1)$

(PCL3) $Belongs(RedOn, s_0) \wedge Causes(s_0, \langle Pressbutton, 0\rangle, 30, s_1)$
$\Rightarrow Belongs(GreenOn, s_1)$

Let Sit_0 denote a situation in which the red light is on, and the yellow and green lights are off. By axiom (PCL1), this knowledge can be expressed as:

(PCL4) $Holds(RedOn, Sit_0)$

Assuming in situation Sit_0 event $E_{PressButton}$ occurs, and let Sit_1 and Sit_2 denote situations, 25 seconds and 30 seconds after the beginning of event $E_{PressButton}$, respectively, then according to (PCL1)-(PCL4) together with the knowledge that for time T_2 (see figure 7.3)

(PCL5) $Sub(t, T_2) \wedge a \neq PressButton \Rightarrow \neg Performs(a, t)$

by Theorem 6.5.1, we can deduce:

(7.3.1) $E\_Causes(Sit_0, E_{PressButton}, T_1, Sit_1)$

where $Dur(T_1) = 25 \wedge Holds(YellowOn, Sit_1) \wedge Dur(Time(Sit_1)) \leq 5$

and

(7.3.2) $E\_Causes(Sit_0, E_{PressButton}, T_2, Sit_2)$

where $Dur(T_2) = 30 \wedge Holds(GreenOn, Sit_2)$

Here we successfully express the fact that there is a delayed time, say T_{D1}, standing between the reference time of event $E_{PressButton}$ and the reference time of the situation Sit_1, that is:

$$Meets(Time(E_{PressButton}), T_{D1}) \wedge Meets(T_{D1}, Time(Sit_1))$$

Similarly, there is a delayed time, say T_{D2}, standing between $Time(E_{PressButton})$ and $Time(Sit_2)$, that is:

$$Meets(Time(E_{PressButton}), T_{D2}) \wedge Meets(T_{D2}, Time(Sit_2))$$

The above knowledge can be graphically presented as figure 7.3:

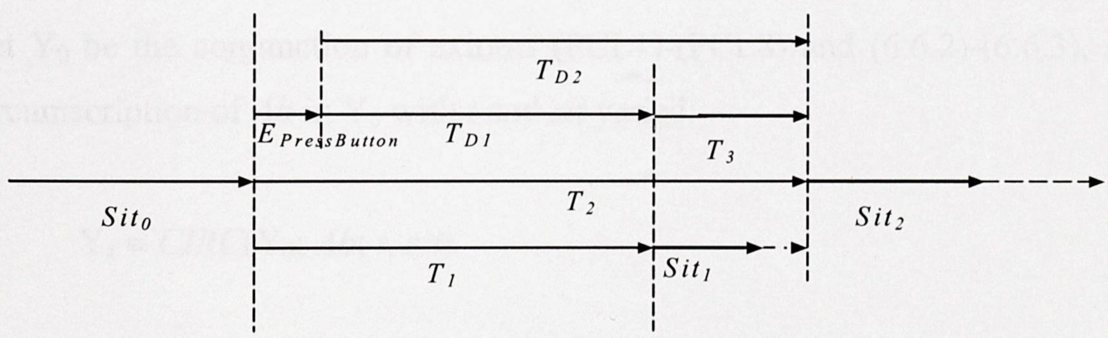

Figure 7.3

Similar to the case in the Yale Shooting scenario, with respect to this expression the frame problem arises. That is, during the delayed time T_{D1}: do all the fluents persist their truth value? In next section we shall propose a nonmonotonic solution to this problem.

7.3.2 The Pedestrian Crossing Light Problem

In our crossing light example, the issue is how to represent the persistence of all the fluents' truth values over the delayed time, i.e., if starting from the original situation Sit_0, action *PressBotton* is performed, does fluent *RedOn* persist over the next 25 seconds?

To deal with this issue by means of using the State-Based Minimisation technique, the existence-of-situations axiom is needed as a domain constraint axiom.

(PCL6) $\exists sit(Holds(RedOn, sit) \wedge \neg Holds(YellowOn, sit) \wedge \neg Holds(GreenOn, sit)) \wedge$
$\exists sit(\neg Holds(RedOn, sit) \wedge Holds(YellowOn, sit) \wedge \neg Holds(GreenOn, sit)) \wedge$
$\exists sit(\neg Holds(RedOn, sit) \wedge \neg Holds(YellowOn, sit) \wedge Holds(GreenOn, sit))$

In addition, the following axioms for uniqueness of event names, and the domain closure for fluents are needed as well:

(PCL7) $\forall a,d(<a, d> = <PressButton, 0>)$

(PCL8) $\forall f(f = RedOn \vee f = YellowOn \vee f = GreenOn)$

Let Y_0 be the conjunction of axioms (PCL1)-(PCL8) and (6.6.2)-(6.6.3), and let Y_1 be the circumscription of *Ab* in Y_0 with *t* and *sit* varied:

$$Y_1 \equiv CIRC(Y_0; Ab; t, sit)$$

We expect that Y_1 ensures the following conclusions

(7.3.3) $E\_Causes(Sit_0, E_{PressButton}, T_1, Sit_1)$
$\wedge Holds(YellowOn, Sit_1) \wedge Holds(RedOn, T_1)$

(7.3.4) $E\_Causes(Sit_0, E_{PressButton}, T_1, Sit_1) \wedge E\_Causes(Sit_0, E_{PressButton}, T_2, Sit_2)$
$\wedge Holds(YellowOn, T_3) \wedge Holds(GreenOn, Sit_2)$

Where Sit_0, Sit_1, Sit_2, T_1 and T_2 are interpreted as in figure 7.3, and T_3 is the time connecting T_1 and $Time(Sit_2)$.

Proposition 7.3 $Y_1 \models$ (7.3.3) and (7.3.4)

Proof: Let M be any model of Y_1. Such a model consists of various domains: the domain of times $|M|_t$, the domain of situations $|M|_{sit}$, the domain of action types $|M|_{<a,\, d>}$, the domain of events $|M|_e$, and the domain of fluents $|M|_f$, as well as for the constants:

$$M[[Sit_0]] \in |M|_{sit},$$

$$M[[<PressButton, d>]] \in |M|_{<a,\, d>},$$

$$M[[<PressButton, Time(E_{PressButton})>]] \in |M|_e,$$

$$M[[RedOn]] \in |M|_f,\ M[[YellowOn]] \in |M|_f,\ M[[GreenOn]] \in |M|_f,$$

interpretations for the relations:

$$M[[Holds]] \subseteq |M|_f \times |M|_{sit},$$

$$M[[Ab]] \subseteq |M|_f \times |M|_{sit} \times |M|_e \times |M|_t \times |M|_{sit},$$

$$M[[E\_Causes]] \subseteq |M|_{sit} \times |M|_e \times |M|_t \times |M|_{sit}\ .$$

To show that M must satisfy (7.3.3) and (7.3.4), axioms about uniqueness of names for fluents, events are needed. Without loss of generality, we can take

$$|M|_f = \{RedOn, YellowOn, GreenOn\}$$

$$|M|_{<a,\, d>} = \{<PressButton, 0>\}$$

with *RedOn*, *YellowOn*, *GreenOn* and $E_{PressButton}$ interpreted as themselves.

For a time T, suppose $E\_Causes(Sit_0, E_{PressButton}, T, Sit_1')$ is true. If time $T = T_1$, by Theorem (6.5.2), we have $State(Sit_1') = State(Sit_1)$. Then by axioms (PCL1) and (7.3.1) we have

(7.3.5) $\neg Holds(RedOn, Sit_1')$.

The question left is to prove that it is impossible for time T to be a proper part of T_1, starting together with T_1, such that (7.3.5) is satisfied. To prove this, by reduction to absurdity, suppose this is true, that is

$$E\_Causes(Sit_0, E_{PressButton}, T, Sit_1') \wedge Starts(T, T_1) \wedge \neg Holds(RedOn, Sit_1')$$

Therefore we have,

(7.3.6) $\langle RedOn, Sit_0, E_{PressButton}, T, Sit_1' \rangle \in M[[Ab]]$

From (PCL4), (7.3.1) and (6.6.3), we know there is another abnormality in model M:

$$\langle RedOn, Sit_0, E_{PressButton}, T_1, Sit_1 \rangle \in M[[Ab]]$$

The existence-of-situations axiom (PCL6) and the domain constraint axiom (PCL1) guarantee the existence of some situation Sit_1'', in which *RedOn* holds, but *YellowOn* and *GreenOn* do not. Then we would be able to further minimise the extension of abnormality by making Sit_1'' as the result situation with respect to Sit_0, $E_{PressButton}$ and T, i.e., we could define another model M' of Y_1 which is exactly like M except that:

$$E\_Causes(Sit_0, E_{PressButon}, T, Sit_1'') \wedge Holds(RedOn, Sit_1'')$$

$$M'[[Ab]] = M[[Ab]] - \langle RedOn, Sit_0, E_{PressButton}, T, Sit_1' \rangle.$$

This would eliminate abnormality (7.3.6) in model M' without introducing any new abnormalities. There may be some other abnormalities of predicate *Ab*. For instance, suppose after time T_1, $State(Sit_1)$ may persist over a non-prime time, say T_1', where $Meets(T_1, T_1')$. Since by axiom (5.2.5) in chapter 5, the time structure employed here is discrete and each time can be represented in the form of adjacent union of a sequence of prime times, we can represent T_1' as $p_1 \oplus p_2 \oplus \ldots \oplus p_n$ where p_i (i=1,2,...,n) are prime times. Then, by axiom (6.6.2) and (6.6.3), we know $\langle RedOn, Sit_0, E_{PressButton}, T_1 \oplus p_1, Sit_p \rangle$, $\langle RedOn, Sit_0, E_{PressButton}, T_1 \oplus p_1 \oplus p_2, Sit_p' \rangle$, etc., are different from $\langle RedOn, Sit_0, E_{PressButton}, T_1, Sit_1 \rangle$ and all belong to M'[[*Ab*]]. But this will not bother our proof since in this case we are not interested in those abnormalities, i.e., they belong to M[[*Ab*]] as well. Therefore we obtain that M' is smaller than M with respect to the circumscription.

Hence M could not have been a minimal model of Y_0, contradicting the assumption that it was a model of Y_1. Therefore we have proved (7.3.3). Similarly we can prove (7.3.4). □

It is worth noting the important role played by the time structure proposed in chapter 5 in this proof. Otherwise, the description of the distinction of abnormalities would not be as easier as that in this proof. The other issue is in the circumscription policy: while circumscribing predicate *Ab*, time *t* is allowed to vary in parallel with the result situation sit_2. This guarantees the situations over time *t* can be checked.

Also, the causal minimisation works correctly for this example. The formalisation and the proof of the relative proposition are similar to that in section 7.1. We will not formalise it in detail here.

7.4 Point-Sensitive Effects

As mentioned in the previous chapter, how to express duration type is an interesting issue in temporal reasoning about actions and change. In some cases, it is necessary to explicitly specify the value of *co* in the expression of a duration type in order to obtain a unique expression for the relative time entity. Also, the specification of the value of *co* helps to deal with the point-sensitive effects of actions. In this section, we will use the throwing ball example to show how to handle this issue in TSTC.

Consider the following scenario:

> *Throwing a ball into the air: after throwing, the ball will go up for 8 seconds and then immediately go down*

In the previous chapter, we have already discussed this example. According to classical physics, while the ball is going up (for 8 seconds), the velocity is not zero (and again, not zero when the ball is going down). Only at the apex (the stationary point) where the ball is neither going up nor going down, the velocity becomes zero (the ball is stationary). There are four fluents involved:

Ball_In_Hand: the ball is in one's hand;
Ball_Going_Up: the ball is going up;
Ball_Going_Down: the ball is going down;
Ball_Stationary: the ball is stationary,

and one action type:

<*Throw*, 1>: throwing the ball, suppose using 1 second,

At low level, one event can be described as:

E_{Throw}: throwing the ball over time T_{Throw}, where $Dur(T_{Throw}) = 1$.

This example falls in case (F) in the category for representing temporal relationship between an event and its effects in the previous chapter. Another issue of discussing this example is to show the necessity of specifying the status of the end point of a time entity. The high level causality can be expressed as:

(TB1) $Causes(s_0, <Throw, 1>, 1, s_1) \Rightarrow Belongs(Ball\_Going\_Up, s_1)$

(TB2) $Causes(s_0, <Throw, 1>, <8, open>, s_1) \Rightarrow Belongs(Ball\_Stationary, s_1)$

(TB3) $Causes(s_0, <Throw, 1>, <8, closed>, s_1) \Rightarrow Belongs(Ball\_Going\_Down, s_1)$

To express this knowledge at low level, suppose Sit_0 is the original situation in which the Ball is held in one's hand. The expected result of the occurrence of event E_{Throw} is: immediately after the event the fluent *Ball_Going_Up* holds true for 8 seconds, then fluent *Ball_Stationary* holds true at a time point, and fluent *Ball_Going_Down* holds true successively. This knowledge can be formalised as following:

In the original situation, fluent *Ball_In_Hand* holds true:

(TB4) $Holds(Ball\_In\_Hand, Sit_0)$

Also, the action *Throw* is performed from this original situation over time T_{Throw}

(TB5) $Holds(Performs, T_{Throw})$

Suppose there are two times T_1 and T_2 that are met by the reference time of situation Sit_0, and Sit_1, Sit_2 and Sit_3 are three situations immediately follow the times T_{Throw}, T_1 and T_2 respectively. The status of the right end of T_1 and T_2 is shown as

(TB6) $Dur(T_1) = 8 \wedge REnd(T_1) = open \wedge Dur(T_2) = 8 \wedge REnd(T_2) = closed$

The temporal constraints among the relevant times can be expressed as:

(TB7) $Meets(Sit_0, T_{Throw}) \wedge Meets(Sit_0, T_1) \wedge Meets(Sit_0, T_2)$
$\wedge Meets(T_{Throw}, Sit_1) \wedge Meets(T_1, Sit_2) \wedge Meets(T_2, Sit_3)$

The constraint on disturbing actions can be written as:

(TB8) $Sub(t, T_2) \wedge a \neq Throw \Rightarrow \neg Performs(a, t)$

The axioms for uniqueness of action type names, and the domain closure for fluents:

(TB9) $\forall a,d(<a, d> = <Throw, 1>)$

(TB10) $\forall f(\; f = Ball\_In\_Hand \vee f = Ball\_Going\_Up$
$\vee f = Ball\_Goning\_Down \vee f = Ball\_Stationary)$

Let Y_0 be the conjunction of axioms (TB1)-(TB10) and axiom (6.6.1), and let Y_1 be the circumscription of *Causes* in Y_0 with *Holds* varied:

$\Upsilon_1 \equiv CIRC(\Upsilon_0; Causes; Holds)$

Then we expect that Υ_1 ensure the following conclusions

(7.4.1) $E\_Causes(Sit_0, E_{Throw}, T_{Throw}, Sit_1)$

$Holds(Ball\_Going\_Up, Sit_1) \wedge True(Ball\_Going\_Up, T_3)$

(7.4.2) $E\_Causes(Sit_0, E_{Throw}, T_1, Sit_2)$

$Holds(Ball\_Stationay, Sit_2)$

(7.4.3) $E\_Causes(Sit_0, E_{Throw}, T_2, Sit_3)$

$Holds(Ball\_Going\_Down, Sit_3)$

Where Sit_0, Sit_1, Sit_2, Sit_3 T_1, T_2 and T_3 are interpreted as in figure 7.4. T_3 is the time extension of the situation Sit_1, i.e. state $State(Sit_1)$ holds true with respect to time T_3. In fact, there exists a situation, whose reference state is $State(Sit_1)$ and reference time is $Time(Sit_1) \oplus T_3$. T_{D1} is a delay time standing between the reference time of event E_{Throw} and the reference time of situation Sit_2. T_{D2} is another delay time standing between the reference time of event E_{Throw} and the reference time of situation Sit_3. It is worth noting that the duration of time T_{D1} and the duration of T_{D2} are the same. However, the value of $REnd(T_{D1})$ is *open*, while that of $REnd(T_{D2})$ is *closed*. The fluent *Ball_Stationay* holding true in situation Sit_2 is a punctual-change type of effects, since the reference time of situation Sit_2 is a time point. The following proposition is the expected conclusion for the example.

Proposition 7.4 $\Upsilon_1 \models$ (7.4.1), (7.4.2) and (7.4.3).

This proposition can be proved following the same style of the proof as in Proposition 7.1.2.

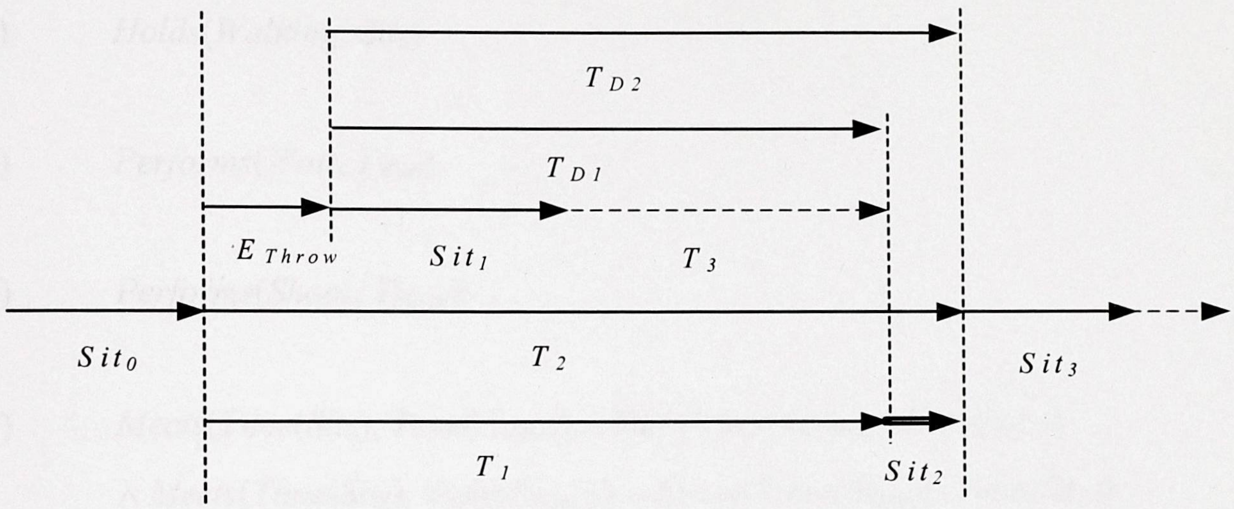

Figure 7. 4

7.5 Ramification

Often, it is impractical to list explicitly all the consequences of an action/event. Rather, some of these consequences will be ramifications; that is, they will be implied by domain constraints [GiS87a]. In this section, we are going to demonstrate that some of the examples for the ramification problem can be handled by the TSTC.

Consider the so-called "walking turkey shoot" scenario [Bak91], a variation of the Yale shooting problem, in which a new fluent *Walking* is added and originally the turkey is walking. Also the shooting action, which directly causes the turkey to be not alive, also indirectly stops the walking. This indirect effect can be represented as a domain constraint stating that in order to be walking, the turkey must be alive:

(WTS1) $Holds(Walking, sit) \Rightarrow Holds(Alive, sit)$

In addition, the walking turkey shoot scenario has the following axioms:

(WTS2) $Causes(s_0, \langle Shoot, 1\rangle, 1, s_1) \Rightarrow \neg Belongs(Alive, s_1)$

(WTS3) $Holds(Loaded, Sit_0)$

(WTS4) $Holds(Walking, Sit_0)$

(WTS5) $Performs(Wait, T_{Wait})$

(WTS6) $Performs(Shoot, T_{Shoot})$

(WTS7) $Meets(Time(Sit_0), Time(E_{Wait})) \wedge Meets(Time(E_{Wait}), Time(Sit_1))$
$\wedge\, Meets(Time(Sit_1), Time(E_{Shoot})) \wedge Meets(Time(E_{Shoot}), Time(Sit_2))$

(WTS8) $Sub(t, T_{Wait}) \wedge a \neq Wait \Rightarrow \neg Performs(a, t)$

(WTS9) $Sub(t, T_{Shoot}) \wedge a \neq Shoot \Rightarrow \neg Performs(a, t)$

(WTS10) $\forall a,d(<a, d> = <Wait, 60> \vee <a, d> = <Shoot, 1>)$

(WTS11) $\forall f(f = Alive \vee f = Loaded \vee f = Walking)$

(WTS12) $\exists sit(Holds(Alive, sit) \wedge Holds(Loaded, sit) \wedge Holds(Walking, sit)) \wedge$
$\exists sit(Holds(Alive, sit) \wedge Holds(Loaded, sit) \wedge \neg Holds(Walking, sit)) \wedge$
$\exists sit(Holds(Alive, sit) \wedge \neg Holds(Loaded, sit) \wedge Holds(Walking, sit)) \wedge$
$\exists sit(Holds(Alive, sit) \wedge \neg Holds(Loaded, sit) \wedge \neg Holds(Walking, sit)) \wedge$
$\exists sit(\neg Holds(Alive, sit) \wedge Holds(Loaded, sit) \wedge \neg Holds(Walking, sit)) \wedge$
$\exists sit(\neg Holds(Alive, sit) \wedge \neg Holds(Loaded, sit) \wedge \neg Holds(Walking, sit))$

These axioms are similar to those for the basic Yale shooting scenario, except that instead of stating that the turkey is alive in the original situation, (WTS4) states that the turkey is walking in situation Sit_0. Also, since a new fluent *Walking* is added, the domain closure for fluents is modified as axiom (WTS11). (WTS12) is the existence-of-situations axiom. This axiom comes from the fact that *not*(*Alive*) and *Walking* are in conflict with each other. Therefore *not*(*Alive*) and *Walking* could not both hold in a consistent state. The expected conclusion is neither fluent *Alive* nor fluent *Walking* holds in situation Sit_2.

Proposition 7.5 Let Σ is the conjunction of (WTS1) - (WTS12), and axioms (6.6.2) - (6.6.3) then

$$\mathrm{CIRC}(\Sigma; Ab; sit_2) \models \neg Holds(Walking, Sit_2).$$

Proof: By (WTS4) and (WTS1), it is easy to see that

$$Holds(Alive, Sit_0).$$

Then, following straightforwardly the same style of proof as in that of Proposition 7.1.1, we can obtain

$$\neg Holds(Alive, Sit_2).$$

Finally, by domain constraint (WTS1), we reach the conclusion that fluent *Walking* does not hold in situation Sit_2. □

Baker [Bak91] asserted that the conventional causal minimization cannot handle the ramification problem correctly. Since then, several proposals [Lin95, Lin96, Sha97 etc.] have been put forward for extending conventional causal minimization to deal correctly with ramification problem. We believe it is not difficult to extend further the causal minimization that has been extended in TSTC to treat the relative ramification problem. In this dissertation, we are not going to address this issue in detail.

7.6 Discussion

In this chapter, a number of applications are demonstrated in order to show the expressive power of TSTC. Through the examples presented above, several issues have been addressed. The first one is how to explicitly represent the causal relationship between the effect and its causal event. This issue has been largely neglected by the researchers in the community of reasoning about action and change. Examples that demonstrate the case where the effect becomes true during the progress (section 7.2) and the case where there is a delay between the effect and its causal event (section 7.3) are successfully expressed.

The second issue is how to distinguish the difference between a temporal duration and its corresponding time entity, which causes problems for some approaches, such as that in [GLR91]. This issue is correctly addressed by representing the throwing a ball example in section 7.4. Our approach is based on the introduction of the notion of duration types. In this example, the ball will reach a stationary status at a time point, and in both sides of this time point the ball will be in different status (*Going_Up* and *Going_Down*). Without employing the duration type and specifying the end status of the corresponding time entity, one may face the difficulty of obtaining a unique causal result.

The third one is how to reason formally about what changes and what doesn't change as the result of some events, i.e., the frame problem. In the light of nonmonotonic reasoning, there are two popular techniques for the frame problem: Causal Minimization and State-based Minimization. Rather than generating new methods, these two approaches have been extended in TSTC and applied to the examples above.

The fourth issue is to represent the uncertain effects. By way of introducing a predicate, *Releases*, the Russian Turkey Shoot example has been represented. Although, we did not discuss this issue further in detail, the successful expression of RTS scenario in TSTC shows the ability to cope with this issue in the formalism proposed here.

Finally, by representing the Walking Turkey Shoot scenario, the ramification problem has been touched upon, although we do not discuss the ramification problem in detail.

CHAPTER 8
CONCLUSIONS

An important aspect of common sense reasoning is the ability to reason about action and change. The possible of change makes the passage of time crucial. This thesis has investigated issues in formal reasoning about action and change, especially focused on the relative temporal representation. In this chapter, the summary of this work is given first. Then a number of ways in which the work in this thesis could be extended are suggested.

8.1 Summary

There are mainly two folds of contributions of this thesis. One of them is the examination of the conceptual issues that arise in formalising dynamic domains. The other is the development of a general formalism for reasoning about action and change over time in which temporal ontology and causal notions are together represented more explicitly than they typically have been in the past.

In this work, some of the fundamental concepts of temporal knowledge representation and the relative issues have been examined. Among them, time plays a central role. The time structure proposed in this thesis allows us to address some of the temporal issues in temporal knowledge representation such as the so-called *intermingling problem* and the *dividing instant problem* correctly. Also, it allows us to distinguish the negation of a given fluent and the negation of involved sentences.

To represent and reason about action and change, some key terms, such as states, situations, actions and events are examined. States are treated as time independent - the state at a given time does not necessarily have to be different from the state at another time, while situations are treated as time dependent - a situation is an association of a given state of the world with a particular time over which the world maintains in that state. Hence, each situation must be taken as unique. In other words, two situations that have different reference times must be treated as different, no matter whether the states at these two times are the same or not. Additionally, since each situation is associated with a particular time, there should be, on intuitive grounds, a constraint that excludes the case that an action may change a situation to an earlier situation. The causal schema taken in this work agrees to this intuitive constraint.

The distinction between states and situations is formally made by way of defining a situation as a pair of a state and a time over which the world holds in the state. In an analogous way, a formal distinction between actions, action-types and events is proposed, which allows the expression of common-sense causal laws at high level. It is shown how these laws can be used to deduce state change over time at low level, when events occur under certain preconditions.

One major advantage of the Temporal State Transition Calculus proposed in this thesis is its approach that allows expression of high level common-sense knowledge, and also supports an explicit representation of time and occurrence of events at a low level. At a high level, the common-sense knowledge can be expressed without any temporal reference, making no reference to any specific times. Although these are relationships involving relative time, they are time independent in that they hold for all time. At low level, knowledge makes references to some given specific time. The Temporal State Transition Calculus combines the knowledge at both high level and low level to represent and reason about action and change by way of employing the high level causal relations as common-sense causal laws to deduce corresponding conclusions at low level.

Another important issue that has been addressed is to do with temporal durations. For interval-based temporal knowledge representation, the issue of how to deal with the relationship between a time duration and its relative time entity is quite interesting and important. A realistic characterisation of most of the examples in chapter 7 would require such a capability. In this thesis, the notion of duration types has been introduced. This helps us to correctly manage the relationship between a temporal duration and the corresponding time entity. Using this method, together with the special treatment of time allow us to express the temporal relationships between actions and their effects, e.g., time-delayed effects, point-sensitive effects and coincident effects of actions, which involve this issue and suffer some of the existing systems, successfully and more efficiently.

Rather than providing any novel technique, the conventional nonmonotonic reasoning techniques, such as causal minimisation and state-based minimisation are correctly extended, to deal with the frame problem, one of the most important and difficult problems in reasoning about action and change. As side effects, the other two related problems, ramification problem and qualification problem, are also briefly addressed.

As the objective of the Temporal State Transition Calculus, the formalism combines many of the existing techniques into a unified, formal framework by way of enriching temporal ontology to the state based formalism and separating the high level and low level change in a natural way. In general, to reason about action and change, there are mainly four questions that need to be answered: one is whether a proposition is true or not; the second is when it becomes true, i.e., the temporal relation between actions and their effects; the third is how long it may persist; and the last is what causes it to be true. Most of the existing formalisms only deal with some of these questions. For instance, the situation calculus and its extensions answer the first, the second and the fourth questions, while the event calculus and its extensions answer the first three questions, although there are no explicit expression available for the complicated temporal relations between actions and their effects. The formalisms proposed here can deal with all the four questions. It allows more flexible temporal causal relationships than do other formalisms for reasoning about causal change, such as the situation calculus or the event calculus. It includes effects that start, during, immediately after, or some time after their causes, and which end before, simultaneously with, or after their causes. The causal axioms guarantee the common-sense assertion that "*the beginning of the effect cannot precede the beginning of the cause*".

8.2 Future Work

The work described in this thesis can be extended in a number of different ways. We now discuss four specific directions for future work.

Reasoning about continuous change. In this thesis a discrete time structure has been employed as the foundation of TSTC. While this time model has proved to be useful to overcome/bypass some problematic issues, it is clear that for many purposes, viewing time in this way is not expressive enough. In particular, when reasoning about the physical world, the natural view of time seems to be that of a continuous model. For example, to represent continuously changing quantities, such as the height of a falling object, the position of a rotating wheel or the level of liquid in a filling tank. Since in TSTC, durations play an important role in the causal change, it seems reasonable to extend the fundamental time structure to a dense one in order to express continuous change. However, in this case, the intermingling problem may arise and hence needs some careful treatments.

Reasoning about complex events and concurrency. The approach proposed in this thesis suppose that all actions are primitive. However, in the real world, occurrences of many actions may overlap in time, which complicates temporal prediction and explanation in AI. For definite goals some actions may be planned to be carried out at the same time in order to save time, or to decrease production cost, or for other context-dependent purposes. Dealing with this problem is also essential for setting in which there are multiple agents, each of which may be performing complex action types. In this work, this issue has not been addressed. However, we believe that based on the formalism proposed here, following the methods proposed by Gelfond *et al* [GLR91] , Lin and Shoham [LiS95] and Miller and Shanahan [MiS94], and some techniques using in programming language community [Fis94 etc.], it is not difficult to extend the formalism in this thesis to represent concurrency of complex events.

Reasoning about Uncertainties. The introduction of the predicate *Release* enables us to represent some sorts of uncertainties, such as the Russian Turkey Shoot example discussed in chapter 7. However, the majority of issues regarding the representation of uncertainties have been left untouched in this work. In [PMK96], Peng *et al* attempted to develop a formalism to deal with uncertainties within situation calculus. Although the work is preliminary, the idea can be extended. In general, when agents devise plans for execution in the real world, they mainly face two forms of uncertainties: they can never have complete knowledge about the situation of the real world, and they do not have complete control, as the effects of actions may be uncertain. The following two questions may be discussed by way of introducing probabilistic representation in TSTC, such as the probability of the effects of an event, the probability of the occurrence of an event, the probability of the potential persistence of a fluent, etc.

- How long will a fluent persist according to the knowledge base?
- As a result of performing an event, what will happen? Further more, at any time (point t), what is the probability of *True*(f, t)?

REFERENCES

[AlF94] Allen, J. F. and Ferguson, G. Actions and Events in Interval Temporal Logic, *Journal of Logic and Computation - Special Issue on Actions and Processes*. Vol.4, No.5, pp.531-579, 1994.

[AlH89] Allen, J. F. and Hayes, P. J. Moments and Points in an Interval-based Temporal-based Logic, *Comput. Intell.* (Canada), (Nov. 1989), Vol.5, no.4, pp.225-238.

[All81] Allen, J. F. An interval-based representation of temporal knowledge, *Proc. 7th Int. Joint Conf. on AI*, 1981, pp.221-226.

[All83] Allen, J. F. Maintaining Knowledge about Temporal Intervals, *communication of ACM.*, Nov. 1983, Vol.26, pp.123-154.

[All84] Allen, J. F. Towards a General Theory of Action and Time, *Artificial Intelligence*, Vol.23, pp.123-154, 1984.

[Bak91] Baker, A. Nonmonotonic Reasoning in the Framework of the Situation Calculus. *Artificial Intelligence*, **49**, pp5-23, 1991.

[Bel89] Bell, J. Nonmonotonic Reasoning, Nonmonotonic Logic & Reasoning about Change. *Technical Report CSM-126*, Essex University, 1989.

[Bel91] Bell, J. Extended Causal Theories. *Artificial Intelligence*, **48**, pp211-224, 1991.

[BFG89] H. Barringer, M. Fisher, D. Gabbay, G. Gough and R. Owens, METATEM: A Framework for Programming in Temporal Logic, in *Proceedings of Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness, LNCS* vol. 430, 1989.

[BKP86] H. Barringer, R. Kuiper and A. Pnueli, A Really Abstract Concurrent Model and Its Temporal Logic, in *Proceedings of the 13th ACM Symp: Principles of Programming Languages*, pp.173 - 183, 1986.

[Bru72] Bruce, B.C.: "A Model for Temporal References and Application in a Question Answering Program", *Artificial Intelligence*, 3(1972), pp.1-25.

[BTK91] Bacchus, F., Tenenberg, J. and Koomen, J. A Non-reified Temporal Logic, *Artificial Intelligence*, **52**, pp.87-108, 1991.

[CaZ97] A. Cau and H. Zedan, Refining Interval Temporal Logic Specifications, In *Proceedings of the Fourth AMAST Workshop on Real-Time Systems, Concurrent, and Distributed Software*, pp. 79-94, Mallorca, Spain, 1997.

[DAA95] Dean, T., Allen, J. and Aloimonos, Y. *Artificial Intelligence: Theory and Practice*, Benjamin Cummings, 1995.

[FiB91] M. Fisher and H. Barringer, Concurrent METATEM Processes - A Language for Distributed AI, in *Proc. of the European Simulation Multiconference*, Copenhagen, 1991.

[Fin87] Finger, J. J *Exploiting Constraints in Design Synthesis*. PhD thesis, Stanford University, Stanford, CA, 1987.

[FiO92] M. Fisher and R. Owens. From the Past to the Future: Executing Temporal Logic Programs, in *Proceedings of Logic Programming and Automated Reasoning (LPAR), St. Petersberg, LNCS*, vol 624, 1992.

[Fis93] M. Fisher. Concurrent METATEM - A Language for Modelling Reactive Systems, in *Parallel Architectures and Languages, Europe* (*PARLE*), Munich, Germany, 1993.

[Fis94] M. Fisher. A Survey of Concurrent METATEM -- The Language and its Applications, in *LNCS* vol.827, 1994

[Fis96] M. Fisher. An Introduction to Executable Temporal Logics, *Knowledge Engineering Review*, 11(1), March 1996.

[Gab91] D. Gabbay, Modal and Temporal Logic II (A Temporal Prolog Machine), in T. Dodd, R. Owens, and S. Torrance Editors, *Logic Programming - Expanding the Horizon*. Intellect Books Ltd, 1991.

[Gab96] D. Gabbay. Declarative Past and Imperative Future. in H. Barringer, M. Fisher, D. Gabbay, R. Owens, and M. Reynolds, editors, *The Imperative Future: Principles of Executable Temporal Logic*, Research Studies Press, 1996.

[Gal90] Galton, A. A Critical Examination of Allen's Theory of Action and Time, *Artificial Intelligence*, **42**, pp.159-188, 1990.

[Gal96] Galton, A. An Investigation of Non-intermingling Principles in Temporal Logic, *Journal of Logic and Computation*, **6**, pp.267-290, 1996.

[Gef98] Geffner, H. *Electronic News Journal on Reasoning about Actions and Change*, Vol. 2, p.26, 1998.

[GeN87] Genesereth, M. and Nilsson, N. *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann. 1987.

[GiS87a] Ginsberg, M.L. and Smith, D.E. Reasoning about action I: A possible worlds approach, in F. Brown (ed.), *The Frame Problem in Artificial Intelligence*, Morgan Kaufmann, pp. 233-258, 1987.

[GiS87b] Ginsberg, M.L. and Smith, D.E. Reasoning about action II: A qualification problem, in F. Brown (ed.), *The Frame Problem in Artificial Intelligence*, Morgan Kaufmann, pp. 259-288, 1987.

[GLR91] Gelfond, M., Lifschitz, V. and Rabinov, A. What are the Limitations of the Situation Calculus? in *Working Notes of AAAI Spring Symposium Series. Symposium: Logical Formalization of Commonsense Reasoning*, pp.59-69, 1991.

[GoG96] Gooday, J. and Galton, A. The Transition Calculus: a High-level Formalism for Reasoning about Action and Change, *Journal of Experimental and Theoretical Artificial Intelligence*, **9**, pp.67-95, 1996.

[Haa87] Haas, A. W. The Case for Domain-specific Frame Axioms, in *Proceedings of the 1987 Workshop: The Frame Problem in Artificial Intelligence*, F. M. Brown ed. pp.343-348, Morgan Kaufmann, April 1987.

[Haj96] Hajnicz, E. Time Structures: Formal Description and Algorithmic Representation, *Lecture Notes in Artificial Intelligence* (1047), Springer, 1996.

[Ham71] Hamblin, C. L. Instants and Intervals, *Studium Generale* **24**, pp127-134, 1971.

[HaM86] Hanks, S. and McDermott, D. Default Reasoning, Nonmonotonic Logics and the Frame Problem, in *Proc. AAAI-86*, pp.328-333, 1986.

[HaM87] Hanks, S. and McDermott, D. Nonmonotonic Logic and Temporal Projection, *Artificial Intelligence*, Vol.33, No.3, pp.379-412, 1987.

[Hau87] Haugh, B. A. Simple Causal Minimizations for Temporal Persistence and Projection, *Proc. AAAI-87*, pp.218-223, 1987.

[Hay77] Hayes, P. In Defence of Logic, *Proc. IJCAI-77*, pp559-569, 1977.

[Hay78] Hayes, P. The Naive Physics manifesto, *Expert systems in the microelectronic age*, Michie D. ed., Edinburgh, 1978.

[Hay85] Hayes, P. Naive Physics 1: Ontology for Liquids, in *Formal Theories of the Commonsense World*, ed. Hobbs J.R. and Moor R.C., Ablex pp71-107, 1985.

[KaL94] Kartha, G.N. and Lifschitz, V. Actions with indirect effects (preliminary report), in *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, pp341-350, Bonn, 1994.

[KaM97] Kakas, A. and Miller, R. Reasoning about actions, narratives, and ramifications. *Linkoping Electronic Articles in Computer and Information Science*, 2(no. 012), 1997.

[Kau86] Kautz, H.A. The Logic of Persistence, in *Proceedings of AAAI-86*, pp.401-405, 1986.

[KeF00] A. Kellett and M. Fisher, Coordinating Heterogeneous Components Using Executable Temporal Logic, in Meyer and Treur (eds), *Agents, Reasoning and Dynamics*. Vol. 6 in *Series of Handbooks in Defeasible Reasoning and Uncertainty Management Systems*. Kluwer Academic Publishers. (To appear)

[KMP94] Y. Kesten, Z. Manna and A. Pnueli, Temporal Verification of Simulation and Refinement, *LNCS*, vol 803, 1994

[KMP97] B. Knight, J. Ma and T. Peng, A Discrete Formalism for Reasoning about Action and Change, *Proc. of The twelfth International Conference on Artificial Intelligence in Engineering*, Capri (Naples), Italy, July, 1997.

[KPM98] Brian Knight, Taoxin Peng and Jixin Ma: Reasoning about Change over Time: Actions, Events, and their Effects, *Proc. of Formalization of Commonsense Reasoning*, pp.183-197, London, UK, 1998.

[KnM92] Knight, B. and Ma, J. A General Temporal Model Supporting Duration Reasoning, *AICOM* Vol.5, No.2 pp.75-84, 1992

[KnM93] Knight, B. and Ma, J. An Extended Temporal System Based on Points and Intervals, *Information System*, Vol.18, 2(1993), pp.111-120.

[KnM94] Knight, B. and Ma, J. Time Representation: A Taxonomy of Temporal Models, *AI Review*, 7, pp.401-409, 1994.

[KoS86] Kowalski, R. and Sergot, M. A Logic-based Calculus of Events, *New Generation Computing*, 4 pp67-95, 1986.

[KoS97] Kowalski, R. and Sadri, F. Reconciling the Event Calculus with the Situation Calculus, A Logic-based Calculus of Events, *Journal of Logic Programming: Special Issue: reasoning about action and change*, **31**(1-3), pp.39-58, 1997.

[Kow92] Kowalski, R. Database Updates in the Event Calculus, *Journal of Logic Programming*, 12 pp121-146, 1992.

[Lif86] Lifschitz, V., Pointwise Circumscription: Preliminary Report, in *Proceedings of AAAI-86*, pp. 406-411, 1986

[Lif87] Lifschitz, V. Formal theories of action, In *Proceedings of the workshop on the Frame Problem*, pp.35-57, 1987.

[Lif90] Lifschitz, V., Frames in the Space of Situations, *Artificial Intelligence*, Vol.46, pp.365-376, 1990.

[Lif94] Lifschitz, V., Circumscription, In Gabbay D.M., Hogger C.J. and Robinson J.A. editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol.3, pp.297-352, Oxford University Press Inc., New York, 1994

[Lin96] Lin, F. Embracing Causality in Specifying the Indeterminate Effects of Actions, in *Proceedings of AAAI-96*, 1996.

[Lip64] Lipschutz, S. *Theory and Problems of Set Theory and Related Topics*, Schaum Publishing Co., 257 Park Avenue South, New York, 1964.

[LiR94] Lin, F. and Reiter, R. State Constrains revisited, *Journal of Logic and Computation*, 4(5) pp655-678, 1994.

[LiS92] Lin, F. and Shoham, Y. Concurrent Actions in the Situation Calculus, in *Proceedings of AAAI-92*, pp.590-595, 1992.

[LiS95] Lin, F. and Shoham, Y. Provably Correct Theories of Action, *Journal of ACM*, **42**(2), pp.293-320, 1995.

[LLR99] Lesperance, Y. Levesque, H.J. and Reiter, R. A Situation Calculus Approach to Modeling and Programming Agents, To appear in A. Rao and M. Wooldridge ed., *Foundations and Theories of Rational Agency*, Kluwer, 1999.

[Lon89] Long, D. A Review of Temporal Logics, *The Knowledge Engineering Review*, Vol.4, No.2, pp.141-162, 1989.

[LRL97] Levesque, H.J., Reiter, R., Lesperance, Y., Lin, F. and Scherl, R. GOLOG: A Logic Programming Language for Dynamic Domains, *Journal of Logic Programming*, **31**, pp.59-84, 1997.

[MaK94] Ma, J. and Knight, B. A General Temporal Theory, *The Computer Journal*, Vol.37, No.2, pp.114-123, 1994.

[MaK96] Ma, J. and Knight, B. A Reified Temporal Logic, *The Computer Journal*, Vol.39, No.9, pp.800-807, 1996.

[McC59] McCarthy, J. Programs with Common Sense, *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, London, pp.75-91, 1959.

[McC63] McCarthy, J. Situations, Actions and Causal Laws, *Stanford Artificial Intelligence Project: Memo 2*, 1963.

[McC77] McCarthy, J. Epistemological Programs of Artificial Intelligence, *Proceedings of IJCAI-77*, pp.1038-1044, 1977.

[McC80] McCarthy, J. Circumscription - a Form of Non-monotonic Reasoning, *Artificial Intelligence*, **13**,pp.27-39, 171-172, 1980.

[McC89] McCarthy, J. Artificial Intelligence, Logic and Formalising Common Sense, In *Philosophical Logic and Artificial Inteligence*, edited by R. Thomason, Dordrecht, Kluwer Academic 1989.

[McC98] McCarthy, J. What is Artificial Intelligence? in http://www-formal.stanford.edu/jmc/, 1998.

[McD82] McDermott, D.V. A Temporal Logic for Reasoning about Processes and Plans, *Cognitive Science*, Vol.6, pp.101-155, 1982.

[McH69] McCarthy, J. and Hayes, P. J. Some Philosophical Problems from the Standpoint of Artificial Intelligence, in *Machine Intelligence*, Vol.4, Eds. Meltzer B. and Michie D., Edinburgh University Press, pp.463-502, 1969.

[McT95] McCain, N. and Turner, H. A causal theory of ramifications and qualifications, in *Proceedings of the International Joint Conference on Artificial Intelligence* (*IJCAI*), pp.1978-1984, Montreal, Canada, Morgan Kaufmann, 1995.

[Mil96] Miller, R. A Case Study in Reasoning about Actions and Continuous Change, *Proceedings of ECAI-96*, pp.624-628, 1996.

[MiS94] Miller, R. and Shanahan, M. Narratives in the Situation Calculus, *Journal of Logic and Computation - Special Issue on Actions and Processes*. Vol 4, No.5, pp.513-510, 1994.

[MKP94] Ma, J., Knight, B and Petridis, M. A Revised Theory of Action and Time based on Intervals and Points, *The Computer Journal*, **37**(10), pp.847-857, 1994.

[MKP97] J. Ma, B. Knight and T. Peng, Representing Temporal Relationships between Events and Their Effects, *Proc. of the 4th Workshop on Temporal Representation and Reasoning*, Daytona, FL, USA, IEEE Computer Society Press, pp.148-152, May, 1997.

[MoM84] B. Moszkowski and Z. Manna, Reasoning in Interval Temporal Logic, In *Proceedings of the ACM/NSF/ONR Workshop on Logics of Programs, LNCS*, Vol 164, Springer Verlag, pp.371-383, 1984.

[Moo82] Moore, R.C. The Role of Logic in Knowledge Representation and Commonsense Reasoning, In *Proc. AAAI-82*, pp.428-433, 1982.

[Mos85] B. Moszkowski, A Temporal Logic for Multilevel Reasoning about Hardware, *IEEE Computer* 18, pp. 10-19, 1985.

[Mos86] B. Moszkowski, Executing Temporal Logic Programs, Cambridge University Press, 1986.

[MoS88] Morgenstern L. and Stein L. A. Why Things Go Wrong: a Formal Theory of Causal Reasoning, in *Readings in Planning*, ed. J. Allen, J. Hendler and A. Tate, pp.641-646. Morgan Kaufmann, 1990.

[Pin94] Pinto, J. *Temporal Reasoning in the Situation Calculus*, PhD Thesis, Computer Science Department, University of Toronto, 1994.

[PiR93] Pinto, J. and Reiter, R. Temporal Reasoning in logic Programming: A Case for the Situation Calculus, in *Proceedings of 10th Int. Conf. on Logic Programming*, Budapest, Hungary, June, pp.21-24, 1993.

[PiR95] Pinto, J. and Reiter, R. Reasoning about Time in the Situation Calculus, *Annals of Mathematics and Artificial Intelligence*, Vol:14, Iss:2-4, pp251-268, 1995.

[PMK96] T. Peng, J. Ma and B. Knight, Probabilistic Temporal Reasoning in Situation Calculus, *Proc. of ICIE-96*, Buenos Aires, Argentina, April, 1996.

[Pnu92] A. Pnueli, System Specification and Refinement in Temporal Logic, in *Proceedings of Foundations of Software Technology and Theoretical Computer Science, LNCS* vol 652, pp.1-38, 1992

[Rei80] Reiter, R. A Logic for Default Reasoning, *Artificial Intelligenc*, Vol.13 pp.41-72, 1980.

[Rei91] Reiter, R. The Frame Problem in the Situation Calculus: a Simple Solution (sometimes) and a Completeness Result for Goal Regression, in V. Lifschitz, ed., *Artificial Intelligence and Mathematical Theory of Computation* (Papers in Honor of John McCarthy), pp.359-380, 1991.

[Rei96] Reiter, R. Natural Actions, Concurrent and Continuous Time in the Situation Calculus, *Proceedings of KR-96*, pp.2-13, 1996.

[Sad87] Sadri, F. Three Recent Approaches to Temporal Reasoning, in A.P. Galton ed., *Temporal Logics and their Applications*, Academic Press, pp.121-168, 1987.

[SaK95] Sadri, F. and Kowalski, R. Variant of the Event Calculus, *Proceedings of ICLP*, pp.67-82, 1995.

[San92] Sandewall, E. *Features and Fluents*. Technical Report LiTH-IDA_R-92-03, Department of Computer and Information Science, Linkoping UniversitySweden.

[San93] Sandewall, E. The Range of Applicability of Nonmonotonic Logics for the Intertia Problem, *Proceedings of IJCAI-93*, pp.738-743, 1993.

[San94] Sandewall, E. *Features and Fluents*: *The Representation of Knowledge about Dynamical Systems*, Vol.1, Oxford University Press, 1994.

[San95] Sandewall, E. Reasoning about Actions and Change with Ramification, *Computer Science Today* Vol.1000 of LNCS, Springer, 1995.

[San96] Sandewall, E. Assessments of Ramification Methods that Use Static Domain Constraints, in *Proceedings of KR-96*, pp.99-110, 1996.

[Sch90] Schubert, L.K. Monotonic Solution of the Frame Problem in the Situation Calculus: an Efficient Method for Worlds with Fully Specified Actions. in H.E.Kyberg, R.P.Loui, and G.N.Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pp23-67. Kluwer Academic Press, Boston, Mass., 1990.

[Sha89] Shanahan, M. Prediction Is Deduction but Explanation Is Abduction, *Proceedings of IJCAI-89*, pp1055-1060, 1989.

[Sha90] Shanahan, M. Representing Continuous Change in the Event Calculus, *Proceedings of ECAI-90*, pp598-603, 1990.

[Sha95] Shanahan, M. A Circumscriptive Calculus of Events, *Artificial Intelligence*, Vol. 77, pp29-384, 1995.

[Sha97] Shanahan, M. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*, The MIT Press, 1997.

[Sho86] Shoham, Y. Chronological Ignorance: Time, Nonmonotonicity, Necessity, and Causal Theories, in *Proceedings of AAAI-86*, pp. 389-393, 1986.

[Sho87a] Shoham, Y. Temporal Logics in AI: Semantical and Ontological Considerations, *Artificial Intelligence*, Vol.33, pp.89-104, 1987.

[Sho87b] Shoham, Y. Reified Temporal Logics: Semantical and Ontological Considerations, *Advances in Artificial Intelligence - II*, B. Du Boulay, D. Hogg and L. Steels (Editors), Elsevier Science Publishers B. V. (North-Holland), 183-190, 1987.

[Sho88] Shoham, Y. *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*, MIT Press, Cambridge, MA,1988.

[Sho90a] Shoham, Y. Nonmonotonic Reasoning and Causation, *Cognitive Science*, **14**, pp. 213-252, 1990.

[Sho90b] y. Shoham, Agent Oriented Programming. Technical Report STAN-CS-1335-90, Department of Computer Science, Stanford University, California, USA, 1990.

[StM94] Stein, L.A. and Morgenstern, L. Motivated Action Theory, *Artificial Intelligence*, **77**, pp.1-42, 1994.

[Sup70] Suppes, P. *A Probabilistic Theory of Causation*, North Holland, 1970.

[TeT95] Terenziani, P. and Torasso, P. Time, Action-Types, and Causation: an Integrated Analysis, *Computational Intelligence*, Vol.11, No. 3, pp.529-552, 1995.

[Thi96] Thielscher, M. Causality and the Qualification problem, in *Proc. of KR-96*, pp.51-62, Morgan Kaufmann, 1996.

[Thi97] Thielscher, M. Ramifications and Causality, *Artificial Intelligence*, Vol. 89, pp317-364, 1997.

[Van83] Van Benthem, J. *The Logicof Time*, D. Reidel Publishing Company, Dordrecht, Holland, 1983.

[VDD95] Van Belleghem, K., Denecher, M. and De Schreye D. Combining Situation Calculus and Event Calculus, in *Proc. of ICLP,* pp.83-97, 1995.

[Vik86] Vilain, M.B. and Kautz, H. Constraint Propagation Algorithms for Temporal Reasoning, in *Proceedings of AAAI-86*, pp. 377-382, 1986.

[Vil82] Vilain, M.B. A System for Reasoning about Time, in *Proceedings of AAAI-82*, pp.179-201, 1982.

[Vil94] Vila, L. A Survey on Temporal Reasoning in Artificial Intelligence, *AICOM* Vol7, No.1, pp4-28, 1994.

[ZCM00] H. Zedan, A. Cau and B. Moszkowski, Compositional Modelling: The Formal Perspective, In *Proceedings of the Workshop on Systems Modelling for Business Process Improvement*, pp. 333-354, 2000.