

# The Ramification and Qualification Problems in Temporal Databases

Nick Papadakis and Dimitris Plexousakis

Department of Computer Science, University of Crete and  
Institute of Computer Science, FORTH  
P.O. Box 2208, GR-71409, Heraklion, Crete, Greece  
Tel: +30 (81) 393528, Fax: +30 (81) 393501  
{npapadak, dp}@csd.uch.gr

**Abstract.** The ramification and qualification problems are two infamous, hard and ever present problems in databases and, more generally, in systems exhibiting a dynamic behavior. The ramification problem refers to determining the indirect effects of actions, whereas the qualification problem refers to determining the preconditions which must hold prior to the execution of an action. A solution to these problems in database systems permits reasoning about the dynamics of databases and allows proving consistency properties. These two problems become increasingly complex in temporal databases and no satisfactory solution has been proposed as of yet. In this paper, we describe these two problems in the context of temporal databases and we propose a solution of polynomial complexity based on the language of the Situation Calculus. This solution extends previous proposals for the solution of these problems in conventional (non-temporal) databases.

## 1 Introduction

Reasoning about action and change has been one of the main research themes of the knowledge representation and planning communities of the last two decades. Action theories providing an axiomatic basis for managing change are applicable to a wide area of disciplines including software engineering [1], (cognitive) robotics [20] and data/knowledge base systems [16]. In this paper we consider the case of database systems. Databases are dynamical systems whose contents change as the result of database transactions. An atomic database transaction can be regarded as an action and hence, we can say that the changes in a database occur as the result of actions. Changes to a database may affect its consistency. Appropriate mechanisms must be employed in order to guarantee that a database will never reach an inconsistent state. To enforce this requirement one must be able to prescribe - in a parsimonious fashion - the exact changes (direct or indirect) that are effected by the execution of an action, and consequently determine which actions should be allowed to execute. These interrelated problems have been known as the *ramification* and *qualification* problems and were initially introduced by McCarthy and Hayes in [13].

We describe these problems by means of an example. Suppose we are interested in maintaining a database describing the contents of a room as part of a robot's perception of its environment. Suppose that the contents of the database are represented as propositions describing the location of each item in the room, as shown below:

$$\begin{aligned} on(bookcase, x_1) \quad on(table, x_2) \quad on(book, x_1) \\ on(bottle, x_2) \quad on(chair, x_3). \end{aligned}$$

Two objects cannot occupy the same room location unless one is stacked on top of the other. As we can observe, the book and the bookcase have the same position. This happens because of the presence of a constraint requiring that books must be on the bookcase (respectively for the bottle and table). The execution of the action  $move(chair, x_4)$  has the effect of the *chair* changing position from  $x_3$  to  $x_4$ . This action has as its only direct effect the change of the position of the *chair*. However, actions may have indirect effects as well. The action  $move(bookcase, x_5)$  has both direct and indirect effects. The direct effect is to change the position of the *bookcase* whereas its indirect effect is to change the position of the *book*, because the *book* is in the *bookcase* and so it moves together with the *bookcase*. Notice that the indirect effect is caused by the presence of the constraint that the book must be on the bookcase.

Whenever an action takes place it is necessary to be able to understand all the direct and indirect effects of this action. Otherwise the contents of database may not satisfy the constraints that describe the consistent states of the database, and thus the database will be inconsistent. In the above example, after the execution of the action  $move(bookcase, x_5)$ , if the position of the *book* does not change, then the contents of database violate the aforementioned constraint.

Such indirect effects are caused by the presence of constraints. The **ramification problem** [3,4] refers to the concise description of the indirect effects of an action in the presence of constraints.

As far as the actions themselves are concerned, not all actions are allowed to take place in any given situation. For each action there are some preconditions which when true, they permit the action's execution. In the previous example, the action  $move(bookcase, x_2)$  is not allowed to execute because a *table* occupies the target position. The action  $move(bookcase, x)$  can be executed only if the position  $x$  is clear. So the precondition of action  $move(p, x)$  is  $clear(x)$ .

The problem of determining the context in which an action is allowed to execute is the **qualification problem** [22]. As we observe, both problems appear in the context of our example and in the context of any changing world, giving rise to the **qualified ramification problem** [24]. To give a brief description of this problem consider that in above example the table and the chair are somehow connected. When the robot moves the table to a new the location, the chair will be moved too. Now the action  $move(table, x_3)$  can be executed because the indirect effect of the action  $move(table, x_3)$  is to change location of the *chair*. Hence, the preconditions  $clear(x_3)$  holds. Before the execution of the action  $move clear(x_3)$  was false. In cases, like this a solution must be able to

take into account the fact that the indirect effects of actions may make action preconditions true.

The rest of paper is organized as follows: in section 2 we review the most prevalent solutions which have been proposed for addressing the ramification and the qualification problems in the context of conventional (non-temporal) databases. We also briefly examine the qualified ramification problem. The ramification and qualification problems in temporal databases are examined at section 3, and a solution is presented at section 4. The paper concludes with a summary and directions for further research.

## 2 Action Theories in Conventional Databases

### 2.1 The Ramification Problem

For the ramification problem many solutions have been suggested. The majority of them are based on the Situation Calculus [13]. The situation calculus is a second-order language that represents the changes which occur in a domain of interest, as results of actions. One possible evolution of the world is a sequence of actions and is represented by a first-order term, called a situation. The initial situation  $S_0$  is a distinguished term representing the situation in which no action has occurred yet. A binary function,  $do(a, s)$  yields the situation resulting from the execution of an action  $a$  while in situation  $s$ . Predicates, called *fluents*, may change truth value from one situation to another and a situation term is used as one of their arguments. Similarly, one can represent functions whose values are dependent on the situations on which they are evaluated (*functional fluents*). Solutions to the ramification problem aim at providing a parsimonious representation of what changes from one situation to the next, when an action takes place.

Among the simplest solutions proposed are those based on the *minimal change* approach [3,25]. These solutions suggest that, when an action occurs in a situation  $S$ , one needs to find the consistent situation  $S'$  which has the fewer changes from the situation  $S$ . For instance, consider as an example, the modeling of a simple circuit which has two switches and one lamp. When the two switches are up, the lamp must be lit. If one switch is down then the lamp must not be lit. Assume the situation  $S = \{up(s_1), \neg up(s_2), \neg light\}$ . The action *toggle - switch*( $s_2$ ) change the situation of the circuit to  $S' = \{up(s_1), up(s_2), \neg light\}$ , which is inconsistent. There are two consistent situations  $S_1 = \{up(s_1), up(s_2), light\}$  and  $S_2 = \{up(s_1), \neg up(s_2), \neg light\}$ . It is sensible to light the lamp, whereas downing the switch  $s_2$  isn't. It is reasonable for the lamp to become lit as indirect effect of "upping" another switch, but it is not reasonable to "down" a switch as indirect effect of "upping" another switch. So we prefer  $S_1$  over  $S_2$ . The minimal change approach cannot select one of them, because they are both equally close to the original situation  $S$ .

The solutions based on the *categorization of fluents* [9,10,11] solve the above problem. The fluents are categorized in *primary* and *secondary*. A primary fluent

can change only as a direct effect of an action, while a secondary one can change only as an indirect effect of an action. After an action takes place, we choose the situation with the fewer changes in primary fluents. In the above example, the separation is  $F_p = \{up(s_1), up(s_2)\}$  and  $F_s = \{light\}$ , where  $F_p$  and  $F_s$  are the primary and secondary fluents respectively. Now we choose situation  $S_1$  because it does not contain any changes in the primary fluents. The categorization of fluents solves the ramification problem only if all fluents can be categorized. If some fluents are primary for some actions and secondary for some other this solution is not satisfactory. For example, consider the circuit in Figure 1. The integrity constraints specifying the behavior of this system are expressed as the following formulas:

$$\begin{aligned} light &\equiv up(s_1) \wedge up(s_2) \\ relay &\equiv \neg up(s_1) \wedge up(s_3) \\ relay &\supset \neg up(s_2) \end{aligned}$$

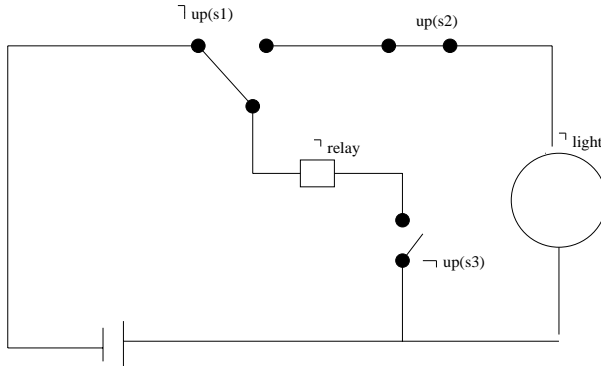


Fig. 1. The complex circuit

Now, the fluents  $up(s_1)$  and  $up(s_3)$  are primary, while the fluents  $relay$  and  $light$  are secondary. The fluent  $up(s_2)$  is primary for the action *toggle – switch*( $s_2$ ) and secondary for the action *toggle – switch*( $s_1$ ). When  $up(s_1)$  and  $up(s_3)$  hold after the execution of action *toggle – switch*( $s_1$ ), the proposition  $\neg up(s_1) \wedge up(s_3)$  holds. This means that the fluent  $relay$  become true. When the fluent  $relay$  is true, it must be the case that  $\neg up(s_2)$  holds. Thus the action *toggle – switch*( $s_1$ ) has as indirect effect  $\neg up(s_2)$ . This means that the fluent  $up(s_2)$  is secondary for the action *toggle – switch*( $s_1$ ). As we can observe, the indirect effect of an action depends on the context of the database. The context is a conjunctive proposition made up of fluents in the database. and provide the enable condition for the effects of actions to be realized In the above example, the context which must be in database in order for the action *toggle – switch*( $s_1$ ) to have as indirect effect  $\neg up(s_2)$  is the fluents  $up(s_1) \wedge up(s_3)$ .

The above solutions suffer from the drawback that they cannot capture the dependence that exists between the indirect effects of action and the context present in the database.

This dependence is captured by the solution of *causal relationships* [2,12,5,21,22]. Each causal relationship consists of two parts. The former part, called **context**, consists of one fluent formula which when true, establishes a causal relationship between an action and its effect. The latter part, is the indirect effect of an action (called the cause of this effect). A causal relationship has the form

$$\epsilon \text{ causes } \rho \text{ if } \Phi$$

where  $\epsilon$  is an action,  $\rho$  is the direct/indirect effect and  $\Phi$  is the context.

One solution based the idea of causal relationships is the language proposed by McCain and Tuner [12]. This language includes static and dynamic laws. A static law is an expression of the form

$$\text{caused } F \text{ if } G.$$

The meaning of static law of this form is that when a formula  $G$  is true the fluent  $F$  must be true. A dynamic law is an expression of the form

$$U \text{ causes } F \text{ if } G.$$

The meaning of a dynamic law of this form is that an action  $U$  has the direct effect  $F$  if the proposition  $G$  holds. For instance, in the example of the previous section, the following dynamic law is defined

$$\text{move}(x,l) \text{ causes } \text{on}(x,l) \text{ if } \text{free}(l).$$

Also, we can define the static law

$$\text{on}(x,l) \text{ if } \text{on}(y,l) \wedge \text{on}(x,y).$$

This law means that if one object  $x$  is on another object  $y$  which is at position  $l$  (possibly after some move), then  $x$  must be at  $l$  as well. Note that static laws capture the indirect effects while dynamic laws capture the direct effects of actions.

## 2.2 The Qualification Problem

We now review briefly solutions proposed for solving the qualification problem. The so-called *default* solution [4] suggests that, for each action  $a$ , we must determine a formula  $F^a$  which, when true, prohibits action  $a$  from executing. The formula  $F^a$  is a disjunction of the form

$$F^a \equiv \bigvee F_i,$$

where each  $F_i$  is a fluent formula. When any of the disjoin  $F_i$  is true, the action  $a$  can not execute. Returning to our example, the disabling fluent formula of the action  $move(x, l)$  has one disjunct:

$$F^{move(x,l)} \equiv on(y, l) \wedge x \neq y.$$

We say that when the formula  $F^a$  holds then the action  $a$  is *disqualified* and thus it cannot execute. This is represented by employing a predicate *disq* as

$$F^a \supset disq(a).$$

Another solution [24] is an extension of the minimal-change possible-worlds approach that has been suggested for solving the ramification problem. After each action  $a$  executes, we try to find a consistent situation which contains all direct and indirect effects of  $a$ . If there is at least one such situation, then the action can execute, otherwise it cannot.

### 3 Temporal Databases

In temporal database systems all action occur at specific points in time. Also objects and relationships among objects exist over time. The value of a fluent is dependent on the time instant at which it is evaluated. Hence, a finer-grained change description mechanism is required here. Recall that, in conventional (non-temporal) databases we only need to determine the value of fluents only after an action occurs.

In this section, we describe the ramification and qualification problems in the context of temporal databases. We describe these problems by means of an example. Assume that the following rule is in effect: if a public employee commits a misdemeanor, then for the next five months he is considered illegal. When a public employee is illegal, then s/he must be suspended for the entire time interval over which s/he is considered illegal. A public employee can receive promotion only if s/he has stayed in the same position for at least five years and is not under suspension. These are expressed in propositional form by the following constraints<sup>1</sup>:

$$occur(misdemeanor(p), t) \supset illegal(p, t_1) \wedge t_1 < t + 5m$$

$$illegal(p, t_1) \supset suspended(p, t_1)$$

$$suspended(p, t_1) \vee (sameposition(p, d) \wedge d < 5y) \supset \neg receivepromotion(p, t_1),$$

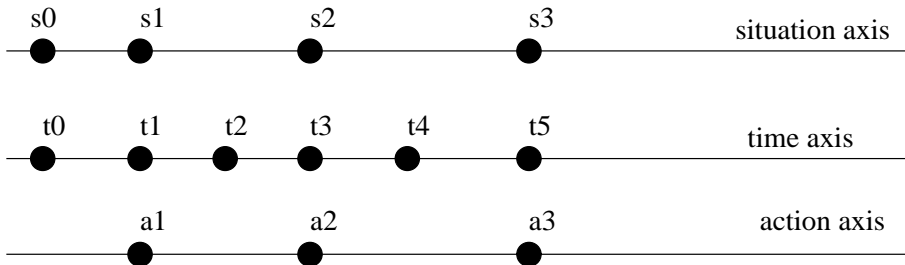
where  $t$  and  $t_1$  are temporal variables and the predicate  $occur(crime(p), t)$  denotes that the action  $crime(p)$  is executed at time  $t$ . In a temporal database we need to describe the direct and indirect effects of an action not only in the

<sup>1</sup> Quantifiers are committed in the expression of these propositions. They are considered to be implicitly universally quantified over their temporal and non-temporal arguments.

immediately resulting next situation but possibly for many future situations as well. In the above example, the action  $\text{misdemeanor}(p)$  has the indirect effect that the public worker is in suspension in the next five months. In this five-month period, a number of other actions may execute leading to many different situations. In all these situations, the action  $\text{misdemeanor}(p)$  has the indirect effect  $\text{suspended}(p)$ .

The causal relationships cannot solve the ramification problem in temporal databases because they determine the direct and indirect effects only for the next situation. The same weakness characterizes all other solutions of the ramification problem in conventional databases. Furthermore, as we can observe, the execution of the action  $\text{misdemeanor}(p)$  disqualified the action  $\text{receivepromotion}$  for the subsequent five-month period. The solutions proposed for the qualification problem in conventional databases cannot address the qualification problem in temporal databases because they cannot represent the fact that one action can disqualify another for a specific time span.

The above weakness can be alleviated by constructing a correspondence between situations and actions with time. Such a correspondence was suggested in previous works [14,9,7]. We adopt the correspondence which was initially suggested in [14] and which is shown in Figure 2. There are three parallel axes: the first is the situation axis, the second is the time axis and the third is the actions axis. We assume that all actions are instantaneous. When an action takes place, the database changes into a new situation.



**Fig. 2.** The correspondence situations and actions with the time

In [14], we have proposed a solution for the ramification problem in temporal databases. More specifically, for each pair  $(a, f)$  of an action  $a$  and fluent  $f$  we define two axioms:

$$\begin{aligned} a(t) \text{ causes } f(t') \text{ if } E_{f_a}^+ \\ a(t) \text{ causes } \neg f(t') \text{ if } E_{f_a}^-, \end{aligned}$$

where the  $E_{f_a}^+$  and  $E_{f_a}^-$  are the formulas which must hold, for fluent  $f$  to become true or false respectively at time  $t'$ , after the execution of action  $a$  at time  $t$ . The above axioms must be specified for any action and the fluents that

can be affected by its execution. The maximum number of axioms that need to be defined is  $O(2 * F * A)$ , where  $F$  is the number of fluents and  $A$  the number of actions. In the next section, we present an improvement to this solution in terms of the number of axioms needed. The improved solution requires the specification of  $O(A + 2 * F)$  such causal laws.

## 4 An Improved Solution

In this section we present an improvement to our previously proposed solution [14] for the ramification and qualification problems in temporal databases. This solution is an extension of the solution of McCain and Tuner [12] for the ramification problem in conventional databases.

We represent each action  $A$  as  $A(t)$ , meaning that the action  $A$  occurs at time  $t$ . Each fluent  $F$  is represented as  $F(t')$ , meaning that the fluent  $F$  is true for time  $t'$  after the current moment. In other words,  $F$  is true in time interval  $[currentmoment, currentmoment + t']$ . When  $\neg F_i(t')$  holds, this means that the fluent  $F$  is false for time  $t'$  after the current moment. As time progresses, the value of  $t'$  is decreased by one time unit.

For each action  $A$ , we define a law of the form:

$$A \supset \bigwedge L_i(t'),$$

where  $L_i(t')$  is  $F_i(t')$  or  $\neg F_i(t')$ . These laws are dynamic and describe the direct effects of an action. Each of these laws are evaluated only when the corresponding action is executed.

Subsequently, for each fluent  $F$ , we define two laws

$$\begin{aligned} G(t) &\supset F(1) \\ B(t) &\supset \neg F(1), \end{aligned}$$

where  $G(t)$  is a proposition which when true causes the fluent  $F$  to become true for the next time-unit. Similarly,  $B(t)$  is a proposition which when true causes the fluent  $F$  to become false for the next time-unit. These laws are static and describe the indirect effect of the execution of actions. They are evaluated in every state of the database. The formula  $G(t)$  and  $B(t)$  are more general than the formulae  $E_{f_a}^+$  and  $E_{f_a}^-$  which are described in the previous solution, because  $E_{f_a}^+/E_{f_a}^-$  specify what must hold in order for the fluent  $f$  to become true/false after the execution of a specific action  $a$ , while the formulae  $G(t)/B(t)$  specified what must hold in order for the fluent  $f$  to become true/false independently of the specific actions.

Notice that, in reference to the correspondence drawn in Figure 1, the dynamical laws are evaluated only when the corresponding action is executed. The static laws are evaluated at each time unit (on the second axis). The execution of static laws does not necessarily change the situation of the database.

The specification of these causal laws solve the ramification problem in temporal databases, since the dynamic laws capture the direct effects of each action



whereas the static ones capture the indirect effects of each action in every state of the database. It is easy to conclude that we need  $A + 2 * F$  such laws, where  $F$  is the number of fluents and  $A$  is the number of actions.

To address the qualification problem we use the predicate *duration* as has been defined in [14]. The interpretation of this predicate is that when  $duration(A, t)$  is true, then the action  $A$  is disqualified for time  $t$  after the current moment. Hence, it represents the duration of the disqualification of the action from executing. At each time unit the value of  $t$  is decreased by one time unit. Then, for each action  $A$  we define one static law.

$$K_A(t, t') \supset duration(A, t'),$$

where  $K_A(t, t')$  is a proposition which when true at time-moment  $t$ , disqualifies the action  $A$  for a time interval of length  $t'$  after the current moment. If some action is disqualified at time instant  $t$ , then it is not necessary to examine the above static law. Its examination becomes necessary only when  $duration(A, 0)$  holds<sup>2</sup>. Hence, to address the qualification problem we need  $A$  laws, where  $A$  is the number of actions.

In total, the specification of  $O(2*(A + F))$  laws is required for the solution of the ramification and qualification problem in the context of temporal databases. Now let us see how the above solution solves these problems for the example we presented in the previous section<sup>3</sup>.

We have one dynamic and one static law, namely:

$$misdemeanor(p, now) \supset illegal(p, 5m) \quad (1)$$

$$illegal(p, t) \wedge (t > 0) \wedge publicemployee(p, t_1) \supset suspended(1) \quad (2),$$

where  $misdemeanor(p, now)$  means that  $p$  commits a misdemeanor at the present moment. The first law is dynamic and captures the direct effect of the action *misdemeanor*. The second law is static and captures the indirect effects of the action *misdemeanor*.

The action *receive - promotion* has the following precondition: first the employee must have been in the same position for at least five years, and second, s/he must not have been suspended. These preconditions are represented as:

$$suspended(t) \wedge (t > 0) \wedge sameposition(p, t_1) \wedge (t_1 < 5y) \wedge t' = max(t, 5y - t_1) \supset duration(receive - promotion, t') \quad (3).$$

The proposition  $K_{receive-promotion}(now, t')$  is specified as

$$\begin{aligned} K_{receive-promotion}(now, t') \equiv \\ suspended(t) \wedge (t > 0) \wedge sameposition(p, t_1) \wedge (t_1 < 5y) \wedge \\ t' = max(t, 5y - t_1). \end{aligned}$$

<sup>2</sup> This means that the action  $A$  is not disqualified.

<sup>3</sup> We do not deal with the problem of changing time granularities in this paper. We assume that different time units are understood and appropriate conversion functions are available.

Law (3) means that, at any time instant, if a public employee is in suspension or has been in the same position for time less than  $5y$ , then the action *receive – promotion* becomes disqualified as long as at least one of these two conditions is true. This ensure because the fluent *sameposition*( $p, 5y$ ) is true at time  $5y - t_1$  from now and the fluent  $\neg$ *suspended* is true for time  $t$  from now. Thus at time  $t' = \max(t, 5y - t_1)$  from the current monet the two fluents are true.

The above problem becomes even more complex if the actions are not instantaneous but have duration. In that case, it is necessary to draw a different correspondece among situations, actions and the time axis than the one of Figure 1. Furthermore, the direct and indirect effects of an action must be determined with regards to the start and/or end of this action. We assume that an action  $A$  with duration is equivalent with two instantaneous actions one for the start ( $start(A, t)$ ) and one for end ( $end(A, t')$ ). We also assume that the action occurs without interaction through at this interval. The above laws are now defined for each action for two time instants, one for the starting point and one for the end point.

In the previous example, assume that the action *misdemeanor*( $p, t$ ) executes during the interval  $[t, t']$ . Then the public employee  $p$  is considered to be illegal for the interval  $[t, t' + 5m]$ . Now we must rewrite the dynamic laws as follows

$$\begin{aligned} start(misdemeanor(p, t)) &\supset illegal(p, \infty) \\ end(misdemeanor(p, t)) &\supset illegal(p, 5m). \end{aligned}$$

The symbol  $\infty$  is used to denote that we do not know when the action of committing the misdemeanor ended. The second law changes  $\infty$  to  $5m$ . We need to specify  $O(2 * A)$  such dynamic laws. Notice that the static laws do not need to change. Hence, for the solution of the ramification problem we need  $O(2 * A + 2 * F)$  laws and for the solution of the qualification problem we do not need to change the previous specification in the case of actions with duration.

## 5 Summary and Future Research

The ramification and qualification problems in temporal database are complex and many-faceted problems. We have described a solution to these problems by adherenig to one such facet, namely that the effects of an action (direct and indirect) refer to the current and future situations only. It is very interesting to investigate the case in which actions can change our beliefs about the past. In that case, the effects may be periodically recursive and for the solution of the ramification and qualification problems, it may be necessary to determine what things can change in the past and what things cannot. It is also worth investigating these problems in the presence of concurrent actions (instantaneous or with duration), or in the case of non-deterministic actions. These are topics of current research.

## References

1. A. Borgida, J. Mylopoulos and R. Reiter. On the Frame Problem in Procedure Specifications. *IEEE Trans. on Software Engineering*, 21(10), Oct. 1995, pp.785-798.
2. C. Elkan. Reasoning about action in first order logic. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI)*, pages 221-227, Vancouver, Canada, May 1992.
3. M. Ginsberg and D. Smith. Reasoning about action I: A possible worlds approach. *Artificial Intelligence*, 35:165-195, 1988.
4. M. Ginsberg and D. Smith. Reasoning about action II: A possible worlds approach. *Artificial Intelligence*, 35:311-342, 1988.
5. J. Gustafson. *Extending Temporal Action Logic for Ramification and Concurrency*, Thesis No 719 of Linkoping Studies in Science and Technology, 1998.
6. R. Fikes and N. J. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189-208, 1971.
7. A. Fusaoka. Situation Calculus on a Dense Flow of Time. *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 633-638, 1996
8. A. Haas. The Case for Domain-Specific Frame Axioms. In F. Brown, editor. *The frame problem in artificial intelligence. Proceedings of the 1987 workshop*, pages 343-348, 1987.
9. V. Lifshitz. Towards a metatheory of action. In J.F. Allen, R. Fikes, and E. Sandewall, editors, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 376-386, Cambridge, MA, 1991.
10. V. Lifshitz. Frames in the space of situations, *Artificial Intelligence*, 46:365-376, 1990.
11. V. Lifshitz. Restricted monotonicity. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 432-437, Washington DC, July 1993.
12. N. McCain and Hudson Turner. A causal theory of ramifications and qualifications. In C. S. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1978-1984, Montreal, Canada, August 1995.
13. J. McCarthy and P.J. Hayes. Some philosophical problem from the standpoint of artificial intelligence. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*, pages 463-502. American Elsevier, New York, 1969.
14. Nikos Papadakis and Dimitris Plexousakis. *Action Theories in Temporal Databases*. 8th Panhellenic Conference on Informatics. Nicosia, Cyprus 8-11 Nov 2001.
15. E. Pednault. ADL: Exploring the Middle Ground between STRIPS and the Situation Calculus. In R.J. Brachman, H. Levesque, and R. Reiter, editors, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning (KR' 89)*, pages 324-332. Morgan Kaufmann, 1989.
16. Dimitris Plexousakis, John Mylopoulos: Accomodating Integrity Constraints During Database Design. *Proceedings of EDBT 1996*, pages 497-513
17. J. Pinto. *Temporal Reasoning in the Situation Calculus*. Ph.D. Thesis, Dept. of Computer Science, Univ. of Toronto, Jan. 1994.
18. J. Pinto and R. Reiter. Temporal Reasoning in Logic Programming: A Case for the Situation Calculus. *Proc. 10th Int. Conf. on Logic Programming*, Budapest, Hungary, June 21-24, 1993.
19. R. Reiter A logic for default reasoning. *Artificial Intelligence*, 13:81-132, 1980.

20. R. Reiter. Knowledge in Action: Logical Foundation for specifying and implementing Dynamical Systems, MIT Press, 2001.
21. M. Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1-2):317-364, 1997.
22. M. Thielscher. Reasoning about actions: Steady versus stabilizing state constraints. *Artificial Intelligence*, 104:339-355, 1988.
23. M. Thielscher. Nondeterministic actions in the fluent calculus: Disjunctive state update axioms. In S. Holldobler, editor, *Intellectics and Computational Logic*. Kluwer, 1999.
24. M. Thielscher. Qualified ramifications. In B. Kuipers and B. Wbber, editors, *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 466-471, 1997.
25. M. Winslett. Reasoning about action using a possible models approach. In *Proceeding of the AAAI National Conference on Artificial Intelligence*, pages 89-93, Saint Paul, MN, August 1988.