Using Temporal Constraints Networks to manage Temporal Scenario of Multimedia Documents

Hélène Fargier[1], **Muriel Jourdan, Nabil Layaïda**[2] and **Thierry Vidal**[3]

Abstract. The purpose of this paper is to show that multimedia applications introduce new open problems in temporal constraint-based reasoning. In particular, we adress three issues related to scenario specification, namely the distinction of controllable and uncontrollable durations, Hierarchical structuration and Interruption-like behaviour. This paper mainly outlines this new requirements in multimedia authoring applications through an example. Then, we partially tackle the raised problems and give future research directions and the work we intend to achieve in this area.

Keywords: multimedia documents, temporal constraints networks, interactivity.

1 Introduction

Multimedia documents combine in time and space different types of elements like video, audio, still-picture, text, synthesized image, ... Compared to classical documents, multimedia documents are characterized by an inherent temporal dimension. Basic media elements, like video, have intrinsic durations. Furthermore, they can be temporally organized by the author which adds to the document a temporal structure called the **temporal scenario.** As far as authoring multimedia documents is concerned one challenging task is to find:

- A temporal representation which is able to handle temporal scenarii.
- Efficient algorithms to analyze temporal scenario in order to check some properties, like the absence of contradictory requirements.
- Efficient algorithms that compute, at the execution phase, a temporal schedule of the document objects which respects the temporal scenario.

There is currently no agreement on the best way to handle these three points. Approaches based on the use of temporal axis, scripts, tree-structures, timed petri-nets or temporal constraints are proposed (see [7] for a survey on these approaches). As [3][9], the Opera Project has promoted a constraint based approach in an authoring environment for multimedia documents named Madeus [8]. Roughly speaking, the author describes the temporal organization of his document by defining temporal objects (one temporal object is associated to each basic media element), and specifying temporal relations between these objects by means of basic (non disjunctive) Allen constraints [1]. A detailed description of Madeus and a comparison of this system with other multimedia authoring environments can be found in [7][8]. In short, this constraint-based approach is motivated by the following reasons:

- The author can **declaratively** express relationships without concern for how they are processed.
- The author can **incrementally** modify his specifications by adding or removing constraints from the current constraint set, or replacing an object by another which may have a different duration, without concern on any other global information: the constraint resolution phase will in charge of checking the consistency of this new scenario and preserving temporal synchronization. The adaptability to the incremental nature of the editing process is very important since building an interactive multimedia document is a cyclic "specify, test and modify" process: one never reaches the right temporal layout at the first stage.
- The author being restricted to use basic Allen's constraints, the constraints resolution phase relies on **efficient** polynomial time algorithms.

Simple temporal constraints satisfaction problems used in Madeus, as in [9] is a formalism that both presents the advantage to be equipped with efficient algorithms and to be rich enough to capture metric informations (typically, the durations of the tasks) and basic Allen's relations. Unfortunately, this framework, and more generally classical temporal constraints satisfaction frameworks (numeric or symbolic), do not suit some important characteristics of the multimedia authoring process: the presence of objects whose duration is not controllable (typically videos), the hierarchical structuration of some objects or some interruption-like behaviors. The aim of this paper is to show through a working example how the application of constraint-based temporal reasoning to multimedia authoring raises challenging open problems to the constraint-reasoning community.

[1] IRIT Toulouse, 118 route de Narbonne, 31062 Toulouse. E-mail: fargier@irit.fr
[2] Opera project, INRIA Rhône-Alpes, 38330 montbonnot St Martin. E-mail: {Firstname.Name@inrialpes.fr}
[3] LGP/ENIT, 47 av d'Azereix, BP 1629, F-65016 Tarbes cedex. E-mail: thierry@enit.fr

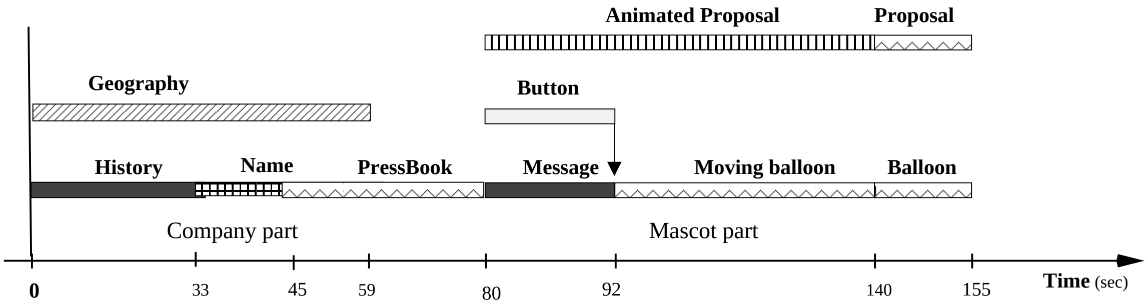

Figure 1. A possible execution of the working example

2 Working Example

"BestCom" is a communication company that answers a call of the International Football Organization for the design of a mascot. In order to provide an attractive and a complete response, BestCom has created a multimedia document to be presented to its client. The scenario is organized in a sequence of two parts: (1) a presentation of the company (called Company) and (2) a presentation of the mascot proposal (called Mascot).

The Company part is composed of a sequence of three objects: an audio clip (History) which gives the history of the company, followed by a textual message displaying the name of the company on the screen (Name) and ends with a graphic listing its main achievements (PressBook). In addition, a movie that gives an overview of the company together with its geographical localization (called Geography) starts at the beginning of the document.

The Mascot part is mainly composed of a virtual animation of the proposal (Animated Proposal). This animation ends with a last picture of the mascot (Proposal). This picture remains displayed on the screen during about 20 seconds together with a balloon (Balloon) on the right of the mascot mouth, which contains its name. In addition, in order to see the mascot name faster, the reader is asked by an audio message (Message) to click on a button (Button) during the presentation of the Animated Proposal. When the reader clicks on the button, the audio message stops and the balloon appears at the top of the screen and moves until it reaches its final position (near the mouth of the mascot) exactly when the animation ends. Figure 1 gives a possible execution of this document, in which the author decides to see the mascot name faster: he clicks on the button 92 seconds after the beginning of the document.

3 Basic Representation Issues

3.1 Syntax and Semantic Issues

At first glance, a temporal scenario is composed of two parts : object declarations and temporal relation (between objects) declarations. The temporal qualification of an object is basically the set of its possible durations, which in our case will always amount to a simple arithmetic interval [min, max]. The relation declaration part specifies a list of Allen's basic constraints between two objects [1]. The formal definition of a scenario is given below :

```
Scenario ::= Scenario_Name ;
             Decl_Obj ;
             Temp_Rel
with

Decl_Obj ::= Obj= [min,max],
Temp_Rel ::= {Obj Rel Obj}*, and
Rel ::={STARTS|MEETS|...|FINISHED_BY|EQUAL}.
```

This definition is illustrated by giving the scenario of the Company part (see Figure 2).

Company = {
History: [30",35"]
Name: [5", 15"]
Pressbook: [20",40"]
Geography: [55",65"]

History *MEETS* Name
Name *MEETS* PressBook
History *STARTS* Geography
}

Figure 2. Scenario of the Company part

A solution (or a schedule) of this specification is a list of n couples (i0,d0), where n is the total number of objects in the scenario, i0 is the beginning point of the object O and d0 its duration, such as d0 belongs to [min0,max0] and all the Allen's constraints are satisfied. A scenario is consistent iff there exists at least one solution of the scenario. During the authoring phase, the consistency property must be checked each time the scenario is modified, in order to ensure its correctness. Solutions have to be computed before the document presentation and also during the authoring phase, since in Madeus a visualization interface helps the author to understand its specification by providing one schedule view.

3.2 Internal Representation

While Allen's relations provide a perfect way of directly handling temporally persisting objects and qualitative relations between them, this formalism is less suitable to efficiently handle metric temporal data in a satisfactory way, especially rather complex information such as delays between objects beginning and/or ending points [6]. We choose to rely on a slightly different internal representation, namely the Simple Temporal Problem formalism [5]: the scenario is translated without any loss of information into a STP by translating both the Allen's relations and the duration constraints into a set of linear inequalities on time points Xi's which are the beginning and ending points of each object. For instance, Figure 3 shows the STP obtained from the scenario given in figure 2.

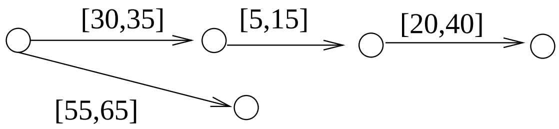

Figure 3. STP of the company part

3.3 Forthcoming issues

We briefly introduce the three open-problems which could not fit in the basic framework presented below. In the working example,

- It is not fair to consider that durations of History and Geography which are respectively an audio and a video object could be controlled in the same way as the other objects. One knows that their duration will surely lie between some bounds a and b, but the effective duration during the presentation of the document strongly depends on the CPU overload and cannot be fixed by the system (see section 4).
- It would be great for the author to express his specification in a structural way which respect the informal presentation: it should be possible to specify both a company part and a mascot part and to put in relation these two structures entities with a MEETS operator (see section 5).
- Last, the Mascot part introduces the notion of interruption (see section 6).

4 Controllable and uncontrollable durations

In classical CSPs, constraints and variables are implicitly such that one can always choose one value in the interval domain when building a solution. This kind of approach is not realistic in our application, since some durations are not under control but are observed at the execution, i.e. during the presentation phase. For instance, it arises with buttons pushed by the document reader : the corresponding event temporal occurrence cannot be controlled by the scheduling system managing the document presentation. Hence checking the consistency cannot be done in the classical way, since we should now check that the presentation will be consistent whatever the values that are taken by the uncontrollable durations in their interval of possible values.

4.1 Previous work

Handling such incontrollable variables in simple temporal problems has been studied in [13]. The idea is to introduce a classification of time-points into two classes: the *activated* ones (which date can be decided beforehand) and the *received* ones (which date will be observed). A similar distinction is made between constraints:

- *Free constraints* are constraints in the classical way: a Free [a,b] between time-points i and j is satisfied by a presentation iff, in this presentation, the delay between occurrences of i and j is greater than a and lower than b: the scheduler is allowed to overconstrain the Free constraint by reducing the interval [a, b]. For instance, the duration of a text presentation, or a precedence constraint between two objects, will be expressed by means of Free constraints.
- A *Contingent constraint* [a,b] between time-points i and j expresses that time point j is not under control but will be observed between a and b time units after the occurrence of time point i. Hence j is necessarily a received time-point (i can be for instance the beginning of a video, activated by the scheduler, and j its ending time which time of occurrence depends on the CPU load at execution time).

It has been shown in [13] that in presence of Contingent constraints, the classical consistency property must be redefined in terms of controllability, for which different alternative definitions are given, corresponding to different requirements at execution time. The one that is specially interesting in reactive applications in which real-time development of a solution plays a crucial role, as in Madeus, is called the Dynamic Controllability. We recall hereafter its informal definition: one can consider that a solution amounts to a totally ordered set of successive instantiations by the scheduler of the activated time-points. Let's call those instantiations "decisions".

The scenario will be *Dynamically controllable* iff one can build a solution such that: for any decision, considering the "past situation" (i.e. the set of received time-points already observed), this decision must ensure that the part of the solution built so far will extend to a complete solution whatever "the situations to come" (i.e. the set of received time-points still to be observed).

This controllability property, is also directly connected to the actual dynamic management of the scheduling process at document presentation time. It reflects indeed the fact that the "execution" will be nothing but a "game against the nature", here the "nature" being both the computer (which for instance influences the actual video processing speed) and the

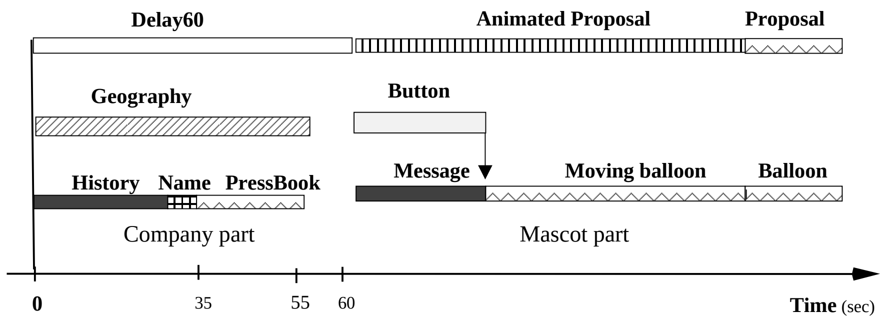

Figure 4. An undesirable schedule

reader (who clicks on buttons). The scheduling system plays "against" these opponents, deciding what it should do next accounting for what they have done. This suggested in [14] a discrete game simulation-like process for checking Dynamic controllability.

4.2 New Issues

The context of multimedia authoring adds new features to this first theoretical work. We indeed have observed that this classification of constraints must be extended to take into account various cases of Free constraints. More precisely, a Free [a,b] between time points i and j will always be such that one can "freely" overconstrain this interval, but we will now distinguish between two distinct behaviors at the presentation phase:

- The Free is a *"Start-Decide"* if and only if the scheduler must choose a precise duration in [a,b] (and hence choose when precisely j will occur), just as soon as i has occurred. For instance, the balloon moving from one point of the screen to another with a fixed speed is a Start-Decide: as soon as the move begins, the speed must be fixed and hence the time at which it will reach its target is also fixed beforehand (so one can say that in this case "Start-Decide" constraints allow to encompass the spatial and temporal features of the object in a unified way).
- The Free is a *"Late-Decide"* if and only if, after the occurrence of i, the scheduler can let time fly before actually choosing a value for the Free (provided that this value is greater than a and lower than b). Hence, the scheduler can wait for at most b time units after the occurrence of i. If j is a received time point, the scheduler can thus possibly wait until j occurs (and fix the value of the Free with the observed delay). For instance, the display of a text is a temporal object whose duration need not be decided at soon as the display begins. The display can be stopped whenever the scheduler decides it, for instance as soon as some other (e.g. received) event stops it.

The approach developed so far in [13] is based on the assumption that all Free are "Late-Decide" constraints. The new types added here for the first time might appear as a rather subtle distinction, but it is very important in the multimedia document environment. Considering all Free as Late-Decide may indeed allow scenarii to be checked as Dynamically controllable whereas there are actually some cases that will lead to a failure. This comes from the definition of the Dynamic controllability itself: each decision to be taken must be valid whatever observations still to be done. Well, considering the Free as a Late-Decide, since one can decide up to the occurrence of j, only observations subsequent to j need be taken into account. On the other hand, in the case of a Start-Decide, since the decision will be taken at once, all observations subsequent to i should be considered as "yet to come". And it is easy to prove that a decision valid for a set of remaining observations is not necessarily valid for a larger set of remaining observations.

We hence need to adapt the definition of Dynamic controllability. Actually the definition itself will remain the same, only the computation of the "past situation" and "situations to come" sets will be modified so as to take into account the two distinct cases of Late-Decide and Start-Decide.

5 Hierarchical structure

Our experiments with multimedia document suggest that the best way to allow modular descriptions is to provide the author with the notion of composite objects and to compose them in time and space (e.g. Figure 4). More formally, it means that the wished syntax of Madeus is the following:

```
Scenario::= Scen_Name;
            Decl_Obj;
            Scenario;
            Temp_Rel

Temp_rel::= {Obj Rel Obj | Obj Rel Scen_Name |
            Scen_Name Rel Scen_Name }*.
```

Recursive definition of a scenario is not allowed.

The intuitive meaning associated with a composite object is: its starting (resp. ending) point is the minimum (resp. maximum) of the starting (resp. ending) points of the objects inside the composite. It is important to notice that this meaning

```
Doc = {
        Company { see figureFig. 2 }
        Mascot {see figure Fig. 6}

        Company MEETS Mascot
}
```

Figure 5. The working example scenario

is not equivalent to a "contains" like definition which would be: its starting point is before (resp. after) the starting (resp ending) points of the objects inside the composite. This last semantics is the one taken in IxTet [10], but is not satisfying in our application context: suppose, the author wants that the company part takes exactly 60". He could add the constraint Delay60 EQUALS Company where Delay60 is an object without content with an exact duration of 60". The author does not want that the schedule shown by the figure Fig. 4 is a solution of his specification, whereas this is a consistent one if we take a "contains" like meaning.

The meaning of composite objects being defined by minimum and maximum operations, we thus have to express disjunctions in the internal temporal representation. Notice that using general TCSP [5] instead of STP would not be sufficient: the constraints that have to be represented are not binary, but can involve more than two time points. In our working example, the ending point of the Company part (say i) must be the maximum of the Pressbook ending point (say j) and of the Geography ending point (say k). This is expressed by :

$$i \geq j \text{ and } i \geq k \text{ and } (i \leq j \text{ or } i \leq k).$$

6 Interruption-like behavior

In this section, we outline the main problems encountered when introducing the so-called interruption operators. In our example, the audio message is presented with a button which is used to interrupt the audio whenever activated by the reader to get faster to the remaining part of the scenario. We would like to express this kind of behavior by providing the author with a `PARMIN` operator, used like Allen's constraints (see Figure 6 for the Mascot part specification).

A relation A *parmin* B means that objects Aand B start together and the shortest terminates the other element. Its semantic could be defined by some equations on start (A_s and B_s) and end instants but we need to distinguish between an expected end (A_{ex} and B_{ex}): the one computed in the schedule, and the effective one (A_{ef} and B_{ef}): the one caused by another object. We obtain ternary constraints:

$$\begin{cases} A_s = B_s, A_{ef} = B_{ef}, \\ A_{ef} = min(A_{ex}, B_{ex}), \\ A_{ex} - A_s \in [minA, maxA] \\ and \\ B_{ex} - B_s \in [minB, maxB] \end{cases}$$

```
Mascot = {
        Animated_Prop.:[55,65]
        Proposal:[10, 25]
        Message:[10,20]
        Moving_Bal: [45,65]
        Balloon:[10,25]
        Button:[0,20]

        Animated_Prop MEETS Proposal
        Message STARTS Animated_Prop
        Button PARMIN Message
        Message MEETS Moving_Bal.
        Mooving_Bal. FINISHES Animated_Prop.
        Proposal EQUALS Balloon
}
```

Figure 6. Scenario of the Mascot part

One difficulty is to merge these equations with those ones deduced from the other constraints of the scenario: the translation which could be done taking constraints one by one before introducing the *PARMIN* operator, must now be more global: depending on A is the operand of a *PARMIN*, the end variable used in the other equations must be either A_{ef} or A_{ex}.

The other difficulty is to extend consistency and schedule algorithms to take into account this constraint. An idea of solution, whose advantage is to be close to the classical one is:

- In the case where both of the intervals are Free. We need only to replace every interval A involved in a parmin relation with another interval B by [min (minA, minB), min (maxA, maxB)]. Then, we proceed with classical consistency checking as for the STP case. When a solution is to be computed, for each of the intervals A and B the one is assigned the value choosen in the schedule and the other one a greater or equal value.
- In the second case, where both intervals are Contingent, a similar transformation is required. In this case we proceed a dynamic controllability checking for interval values for A and B of [min(m1, m2), min(M1, M2)] value. This new interval is a Contingent one.
- The third case, where one interval only is Contingent, for instance B, is much trickier. If the interval values of A and B overlap, for example A=[1,4] Free and B=[3,6] Contingent then one cannot define if the obtained value is Free or Contingent. This is because this latter is Free if the solution is taken in the value range [1,3] while if in the [3,4] range the interval is Constraint. If we take the same example, with A Contingent and B Free, we end with an Contingent where a restriction, through constraint propagation, is possible ! In our case it can only affect the upper bound of the resulting interval. Therefore, we obtain a new type of interval which is a combination of Free and Contingent.

Another "interruption like" operator which also introduces a disjunction with two kinds of end variables is called parmaster where a designated object A when it ends interrupts an object B iff it has not yet ended. Details about difficulties introduced by this operator are given in [11].

7 Discussion and Intended work

Multimedia authoring applications raise different open problems in temporal constraint-based reasoning:

- The existence of temporal objects with incontrollable durations leads to account for new consistency paradigms (and associated algorithms), e.g. the Dynamic controllability.
- The need for hierarchical structure of temporal objects introduces new kinds of disjunctive temporal constraints, e.g. non-binary constraints, that lie outside the classical TCSP model.
- The need to introduce new synchronization primitives like parmin and parmax. These primitives introduces ternary constraints obliging us to make a distinction between two kinds of ending points.

To address those problems that are crucial in this domain, we have started considering that points 1 and 3 can actually be seen as conditional schedule specifications. A possible direction of research lies indeed in the ambitious research area of conditional planning and scheduling: the uncontrollable durations imply different possibilities for the schedule that will only be assignable at execution time, this feature being emphasized by the interruption-like constraints introducing synchronizing behaviors at presentation time.

Then the advantages of having a compact STP model, close to the specifications (constraint-based description), where consistency checking is easy, is counter-balanced by their basic static feature, and hence their inability of representing and reasoning upon conditional issues. An alternative is to rely upon a simulation-based model, close to the final schedules one gets, and hence close to the "execution" presentation phase, both in terms of model and of efficiency. Discrete Event Systems formalisms, for instance finite-state automata, are well-suited for this kind of approaches. Their advantage is that conditional and synchronizing features are naturally expressed since the model encompasses all the possible schedules that might be processed. But this strength is also a weakness, as such models get generally rapidly huge, and are therefore not easy to manage and to view, especially in an incremental mechanism.

So our current work that has just begun consists in mixing both representations, in the spirit of [4], keeping a compact constraint-based model, extending it with the threefold aspects introduced in this paper, but only as a specification and viewing tool. Reasoning issues should then be saved for discrete-event based simulation tools, for which we hope to find more compact tools than the classical automata. We are especially interested in recent works in the area of controllers playing a "continuous game against the nature" [2].

REFERENCES

[1] Allen J.F., "Maintaining Knowledge about Temporal Intervals", *Communications of the ACM,* vol. Vol. 26, num. No. 11, pp. 832-843, November 1983.

[2] Asarin E., Maler O. and Pnueli A., "Symbolic controller synthesis for discrete and timed systems", *Hybrid Systems II,* LNCS 999, Springer Verlag, 1995 .

[3] Buchanan M.C., Zellweger P.T., "Automatic Temporal Layout Mechanisms", *Proceedings of the First ACM International Conference on Multimedia,* pp. 341-350, Anaheim, California, August 1993.

[4] Coradeschi S. and Vidal T., "Accounting for temporal evolutions in highly reactive decision-making"', *5th Int. Workshop on Temporal Representation and Reasoning (TIME98),* Sannibel Island (Florida), USA, May 1998 .

[5] Dechter R., Meiri I., Pearl J., "Temporal Constraint networks", *Artificial Intelligence,* vol. 49, , pp. 61-95, 1991.

[6] Dorn J., "Hybrid Temporal Reasoning", *Proceedings of the 11th European Conference in Artificial Intelligence (ECAI),* pp. 625-629, Amsterdam (NL), 1994.

[7] Jourdan M., Layaïda N. , Roisin C., "A survey on authoring techniques for temporal scenarios of multimedia documents", *vol. to be published in Handbook of multimedia,* CRC Press, April 1998 .

[8] Jourdan M., Layaïda N., Sabry-Ismail L., Roisin C., "An Integrated Authoring and Presentation Environment for Interactive Multimedia Documents", *Proceedings of the 44th Conference on Multimedia Modelling, World Scientific Publishing,* Singapore, 18-19 November 1997.

[9] Kim M. Y., Song J., "Multimedia Documents with Elastic Time", *Proceedings of the Third ACM International Conference on Multimedia,* ACM Press, ed., pp. 143-154, Polle Zellweger, San Francisco, California, 5-9 November 1995.

[10] Laborie P. and Ghallab M, "IxTeT: an Integrated Approach for Plan Generation and Scheduling", *Proceedings ETFA-95,* pp. 485-495, Paris, 1995.

[11] Layaïda N., *Madeus: Système d'édition et de présentation de documents multimédia,* Thèse de l'université Joseph Fourier, Grenoble, juin 1997.

[12] Layaïda N., Sabry-Ismail L., "Maintaining Temporal Consistency of Multimedia Documents Using Constraint Networks", *Multimedia Computing and Networking 1996, M. Freeman, P. Jardetzky, H. M. Vin, ed., pp. 124-135, SPIE 2667,* San-José, USA, February 1996.

[13] Vidal T and H. Fargier, "Contingent durations in temporal CSP : from consistency to controllabilities", *4th Int. Workshop on Temporal Representation and Reasoning (TIME97),* Daytona Beach (Florida), USA, May 10 -11 , 1997.

[14] Vidal T and H. Fargier, "Handling contingency in Temporal Constraint Networks: from consistency to controllabilities", *Submitted to a special Track of the Journal of Experimental and Theoretical Artificial Intelligence (JETAI),* 1998.

[15] Vilain M., Kautz H. A., Van Beek P., "Constraint Propagation Algorithms: a revised report", *Readings in Qualitative Reasoning about Physical Systems,* Morgan Kaufman, 1989.