Representing and Matching Temporal Scenarios

Temporal Reasoning
Common Sense Reasoning
Case-based Reasoning
Pattern Recognition

Abstract

This paper presents a unified schema for representing temporal scenarios with rich internal temporal aspects, supporting both absolute and relative temporal knowledge. A temporal scenario is defined as a vector of various states of the word in the discourse, together with the corresponding temporal constraints, whereas a state is represented as a collection of fluents whose truth-values are dependent on the time. A temporal network, called scenario graph, is formally introduced to graphically represent temporal scenarios formalized in terms of the unified schema. A navigation-based algorithm is proposed, and the simulation experiments demonstrate that graph-matching algorithms/methodologies can be adopted to match and cluster scenario graphs.

1 Introduction

Like the concept of case in case-based reasoning, the notion of state in state-based systems is fundamental for many real-time applications. In conventional state-based systems, various states of the world in the discourse are usually represented in terms of isolated snapshots, while the state histories (which will be formally characterized as temporal scenarios) with rich internal temporal aspects are neglected in most approaches. Over the past three decades, it has been noted that temporal representation and reasoning is essential for many areas of Artificial Intelligence, where one is interested not only in the representation of distinct episodes of an enterprise, but also in the history of earlier/future situations [Snodgrass and Ahn, 1985; Shahar, 1990; Nakhaeizadeh, 1994; Keravnou, 1995; Jare et al., 2002]. In particular, an appropriate representation and reasoning for temporal knowledge is necessary for many state-based systems, where the history of states, rather than distinct episodes, plays an important role in solving problems including explanation/diagnosis, prediction/forecast, planning/scheduling, process management, and history reconstruction, etc. For instance, in the area of medical information systems, the patients' medical histories are obviously very important. In fact, to prescribe the right treatments, the doctor needs to know not only the patients' current status, but also their previous health records. Similarly, in the weather forecasting, without a good understanding of climate phenomena based on past observations, the weather expert cannot make good predictions of the future.

A natural approach to represent the temporal constraints on certain states is to associate the states with time elements. Generally speaking, there are three known choices as for the sort of objects to be taken as time elements: (1) points, i.e., instant without duration; (2) intervals, i.e., periods with positive duration; and (3) both points and intervals. In addition, in temporal systems where time intervals are modelled as time elements, there are two different approaches. In the first, intervals are modeled as derived objects constructed from points, e.g., as sets of points [McDermott, 1982; van Benthem, 1983], or as pairs of points [Bruce, 1972; Shoham, 1987; Dechter et al., 1991; Ladkin, 1992; van Beek, 1992]. However, it has been argued in the literature that defining intervals as objects derived from points may lead to the so-called Dividing Instant Problem [van Benthem, 1983; Allen, 1984; Galton, 1990; Vila, 1994; Ma and Knight, 2003], which is in fact an ancient historical puzzle encountered when attempting to represent what happens at the boundary instant that divides two successive intervals over which two incompatible propositions hold true respectively. Such a problem is closely associated with the question of whether point-based intervals are characterized as closed or open at their ending-points. The second treatment takes intervals as primitive objects, without insisting on the existence of "ending-points", "internal-points", or any points at all. Allen's interval logic [Allen, 1981; Allen, 1983; Allen, 1984; Allen and Hayes, 1989], Vilain's temporal system [Vilain, 1982], and Ma and Knight's general time theory [Ma and Knight, 1994] are examples that treat intervals as primitive.

By common sense, on the one hand, both time points and intervals seem to be needed for general treatments. In fact, while some time-dependent statements can be only validly and meaningfully associated with time points (e.g., "The light was automatically switched on at 8:00pm"), or only with time intervals (e.g., "John read a book for about two hours"), some others may be validly and meaningfully associated with both time points and intervals (e.g., "The robot was stationary"). On the other hand, while many applications require temporal knowledge to be expressed by means of associating time-dependent statements with absolute time values (e.g., “The database was updated at 0:00 midnight”, “The court was opened at 9:15 and was adjourned at 12:30”, etc.), there are some other situations in which there may be just some relative temporal knowledge about the time-depended statements to hand, where their precise time characters (e.g., the exact starting and finishing time) are not available (e.g., “John ran 3 miles yesterday morning”, “John arrived at the office before Mary went to home”, etc.).

Generally speaking, pattern recognition aims at the operation and design of technologies to pick up meaningful patterns in data [Tveter, 1998]. While pattern classification is about putting a particular instance of a pattern in a category, the goal of pattern matching is to determine how similar a pair of patterns are [Theodoridis, 2003]. In state-based or case-based reasoning, certain states/cases may be associated with specific time elements, where various temporal relations between the involved time elements will characterize different scenario patterns. For example, Figure 1 and Figure 2 illustrate two distinct medical scenario pattern samples with the same collection of symptoms but different temporal relations:

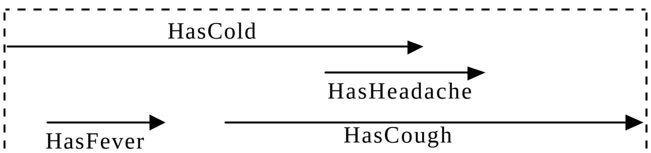

Figure 1. Sample 1

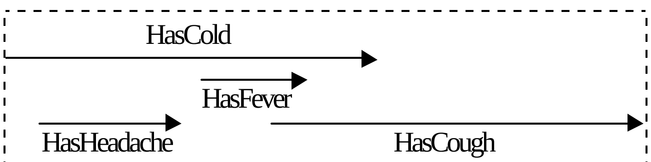

Figure 2. Sample 2

The objective of this paper is to introduce a formal framework in terms of a unified schema for formalizing scenario patterns and the corresponding graphical representation. The formalism is presented in sections 2. In section 3, a temporal network, called Scenario Graph, is introduced for graphical representation of temporal scenarios. A navigation-based algorithm is proposed in section 4, where the simulation experiments demonstrate that graph-matching algorithms/methodologies can be used to recognize temporal scenarios. Finally, section 5 provides a brief summary and concludes the paper.

2 The Formalism

For general treatments, as the time basis, we shall adopt a theory which takes a non-empty set T of points and intervals as time elements, together with a primitive temporal relation *Meets* over time elements, and a function *Dur* from time elements to non-negative real numbers. A time element t is called an interval if $Dur(t) > 0$; otherwise, t is called a point. Such a time model is indeed an extension to the interval-based theory of Allen and Hayes [1989]. Analogous to those introduced by Allen [1983], other order relations over intervals/points can be derived in terms of the primitive relation *Meets*, including *Equals*, *Before*, *After*, *Meets*, *Met-by*, *Overlaps*, *Overlapped-by*, *Starts*, *Started-by*, *During*, *Contains*, *Finishes* and *Finished-by* (Detailed axioms can be found in [Allen and Hayes, 1989] and [Ma and Knight 1994]).

In this paper, we shall use the term fluents to represent Boolean-valued propositions whose truth-values are dependent on the time. We shall denote the set of all the fluents as F, and use a "global predicate" [Allen, 1984; Shoham, 1987]: $Holds(f, t)$, to state that fluent f holds true with respect to time t.

A state of the world in the discourse is defined as a collection of fluents. The set of states is denoted as S. We shall use $Belongs(f, s)$ to represent fluent f belongs to state s [Shanahan, 1995].

Without confusion, we also use $Holds(s, t)$ to denote that state s holds for time t, provided that every fluent f belongs to state s holds for time t.

In addition, we introduce two binary operators, *Union* and *Intersection*, so that $Union(s_1, s_2)$ and $Intersection(s_1, s_2)$ denote the *union*, and the *intersection*, of states s_1 and s_2, respectively.

A temporal scenario, st, is formalized in terms of a unified scheme, represented as a quadruple, $st = \langle State^{st}, Holds^{st}, Meets^{st}, Dur^{st} \rangle$, such that

$State^{st} = \{s^{st}_i \mid s^{st}_i \in S,\ i = 1, \ldots, m\}$,
$Holds^{st} = \{Holds(s^{st}_i, t^{st}_i) \mid t^{st}_i \in T,\ 1 \le i \le m\}$
$Meets^{st} = \{Meets(t^{st}{'}, t^{st}{''}) \mid \text{for some } t^{st}{'}, t^{st}{''} \in T^{st}\}$
$Dur^{st} = \{Dur(t) = r \mid \text{for some } t \in T^{st}, r \in R\}$

where T^{st} is the minimal subset of T closed under the following rules:

$t^{st}_i \in T^{st}$, i = 1, …, m.
$t \in T^{st} \Leftrightarrow \exists t' \in T^{st}(Meets(t, t') \vee Meets(t', t))$

N.B. Here, the temporal relations involved in scenario st is simply presented in the “unified” form in terms of the set of Meets relations, since all the other order relations can be derived from Meets.

3 Scenario Graphs

A graphical representation for expressing temporal knowledge has been introduced previously in terms of a directed and partially weighted graph [Knight and Ma 1992]. It can be extended to express temporal scenarios presented in the unified structure as proposed in section 2. In fact, any given temporal scenario st can be denoted as a temporal network, defined as a directed, partially weighted/labeled simple graph G^{st}, called *Scenario Graph*, where:

- Each time element t in T^{st} is denoted as a directed arc of the graph labeled by t that is bounded by a pair of nodes, which are called the *tail-node*, and the *head-node*, of the arc, respectively.

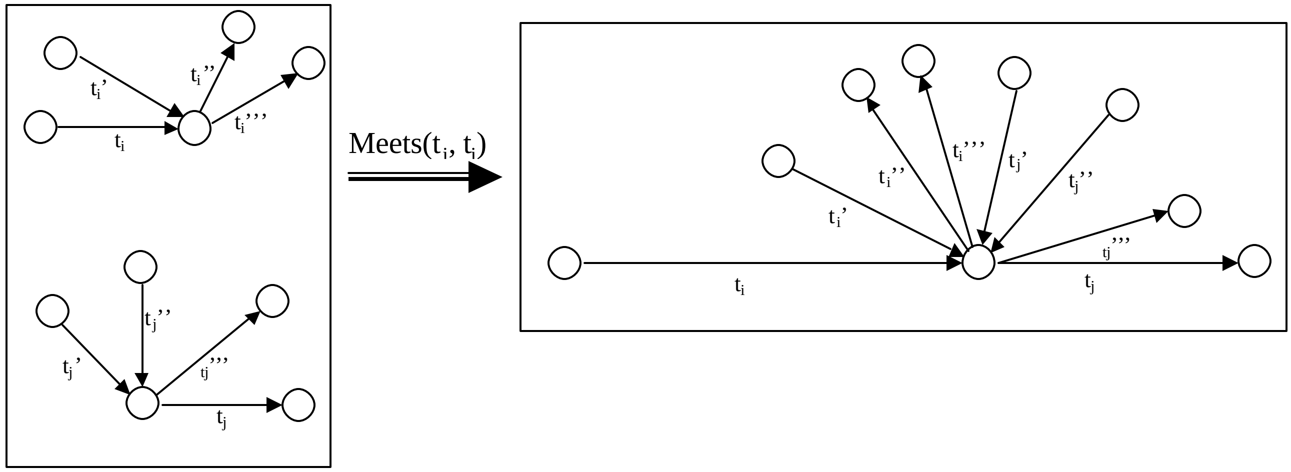

Figure 3. Merging the head-node of t_i and the tail-node of t_j

- Each relation *Meets*(*ti*, *tj*) in $Meets^{st}$ is represented by means of merging the *head-node* of t_i and the *tail-node* of t_j as a common node, of which *ti* is an in-arc and *tj* is an out-arc, respectively (see Figure 3). In this case, arc t_i is said to be adjacent to arc t_j.
- Each formula $Holds(s^{st}_i, t^{st}_i)$ in $Holds^{st}$ is represented by means of simply adding s^{st}_i as an additional label to the arc labeled by the corresponding t^{st}_i. For any time element t in T^{st}, if there is no *Holds* knowledge, it will be labeled by the empty state {}.
- Each piece of duration knowledge $Dur(t) = r$ in Dur^{st} is expressed as a real number, r, alongside the corresponding arc t.

For scenario graph G^{st}, we define a $|T^{st}|$-by-$|T^{st}|$ Meets-adjacency matrix M^{st}, where

$$M(i,j) = \begin{cases} 1, & if\ Meets(t_i, t_j) \\ 0, & otherwise \end{cases}$$

As an example, consider a scenario *st* = <*State*, *Holds*, *Meets*, *Dur*> where

$State^{st} = \{\{c_1,c_2\}, \{c_1\}, \{c_2\}, \{c_1,c_3\}\}$,
$Holds^{st} = \{Holds(\{c_1,c_2\},t_1), Holds(\{c_1\},t_2), Holds(c_2,t_4), Holds(\{c_1,c_3\}, t_5)\}$
$Meets^{st} = \{Meets(t_1,t_2), Meets(t_1,t_3), Meets(t_2,t_4), Meets(t_3,t_5), Meets(t_4,t_6), Meets(t_5,t_6)\}$
$Dur^{st} = \{Dur(t_1) = 1, Dur(t_2) = 0.5, Dur(t_3) = 0.7, Dur(t_4) = 0.5, Dur(t_5) = 0.3\}$

The graph representation of such a scenario, G^{st}, is shown in Figure 4 as below:

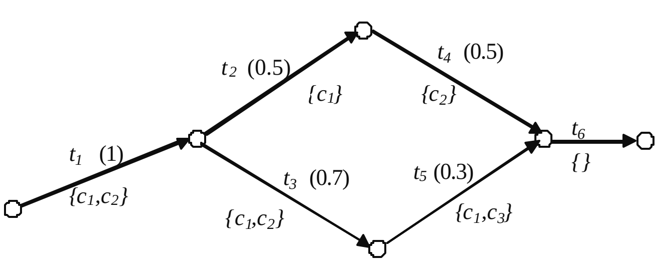

Figure 4. A scenario graph sample

and the corresponding *Meets*-adjacent matrix is:

$$M^{st} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4 A Navigation-based Algorithm

In what follows, by assuming the set of fluents F is finite, i.e., $F = \{f_1, \ldots, f_n\}$, we shall propose a navigation-based algorithm for matching scenarios graphs.

4.1 The Algorithm

Given two scenarios, st_1 and st_2, we assume $|T_1^{st}| \le |T_2^{st}|$. We use f to denote a one-to-one function from $\{1, \ldots, |T_1^{st}|\}$ to $\{1, \ldots, |T_2^{st}|\}$, and compute the following similarity degrees:

- Similarity of graph size:

$$sim_{size}(st_1, st_2) = \frac{|T_1^{st}|}{|T_2^{st}|}$$

- Similarity of *Holds* relations:

$$sim_{Holds}(st_1, st_2, f) = \frac{1}{|T_1^{st}|} \sum_{i=1}^{|T_1^{st}|} \frac{|Intersection(s_1^i, s_2^{f(i)})|}{|Union(s_1^i, s_2^{f(i)})|}$$

- Similarity of duration assignments:

$$sim_{Dur}(st_1, st_2, f) = 2 \frac{\sum_{i}^{|T_1^{st}|} |Dur(t_1^i) * Dur(t_2^{f(i)})|}{\sum_{i=1}^{|T_1^{st}|} |Dur(t_1^i)|^2 + |Dur(t_2^{f(i)})|^2}$$

- Similarity of *Meets* relations:

  For scenarios st_1 and st_2, we use M_1 and M_2 to denote their corresponding Meets-adjacency matrices respectively, and, with respect to f, we derive a $|T_1^{st}|*|T_1^{st}|$ matrix M_2^f from M_1 and M_2, such that:

$$M_2^f(i,j) = M_2(f(i), f(j))$$

Then, the similarity of Meets relations between scenarios st_1 and st_2, with respect to function f, is defined as:

$$sim_{Meets}(st_1, st_2, f) = 2\frac{\left|< M_1, M_2^f >\right|}{\|M_1\|^2 + \|M_2^f\|^2}$$

where

$$< M_1, M_2^f >= \sum_{i=1}^{|T_1^{st}|}\sum_{j=1}^{|T_1^{st}|} M_1(i,j) * M_2^f(i,j)$$

- The overall similarity between scenarios st_1 and st_2, with respect to function f, is defined as:

$$\begin{aligned} sim(st_1, st_2, f) &= sim_{size}(st_1, st_2) \\ &* (w_1 * sim_{Holds}(st_1, st_2, f) + w_2 * sim_{Dur}(st_1, st_2, f) \\ &+ w_3 * sim_{Meets}(st_1, st_2, f))/(w_1 + w_2 + w_3) \end{aligned}$$

- Finally, the similarity between scenarios st_1 and st_2 is defined as:

$$sim(st_1, st_2) = \max_f sim(st_1, st_2, f)$$

4.2 Experimental Results

The algorithm has been implemented in MatLab. What follows describes some experiments conducted, where the corresponding weights taken in the algorithm are: $w_1 = w_2 = 0.25$ and $w_3 = 0.5$.

Test 1.
As shown in Figure 5, for the given 7 pairs of scenario graphs (ST_1 and ST_2), the corresponding results computed from the algorithm provide a well stable similarity measurement as expected.

Test 2.
Figures 6 – Figure 8 represent three graphs corresponding to real-world scenarios in medical records, respectively:

The computed similarities produced from the algorithm s for these three scenarios are listed in Table 1.

Table 1. Similarity list

	Scenario 1	Scenario 2	Scenario 3
Scenario 1	1	0.7211	0.6878
Scenario 2	0.7211	1	0.6042
Scenario 3	0.6878	0.6042	1

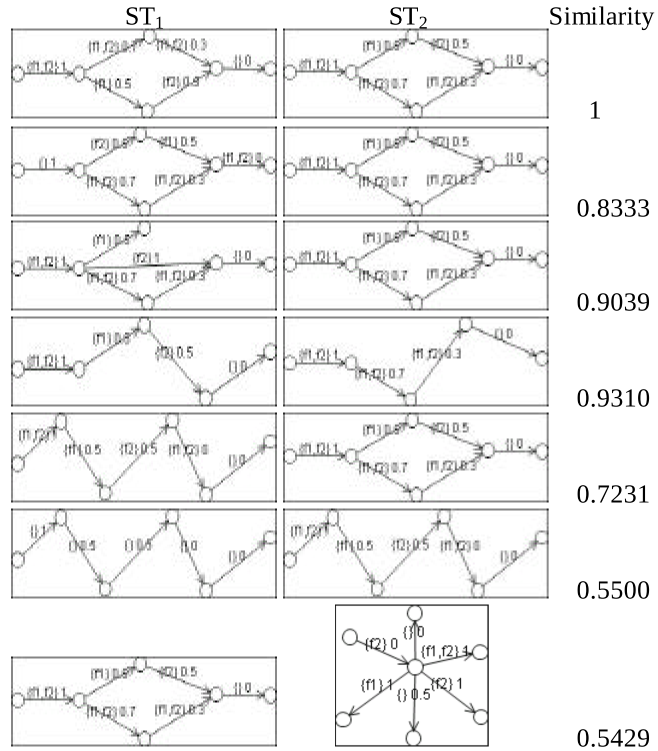

Figure 5. Test 1

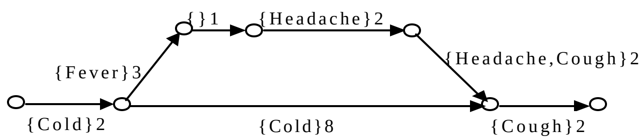

Figure 6. Scenario 1

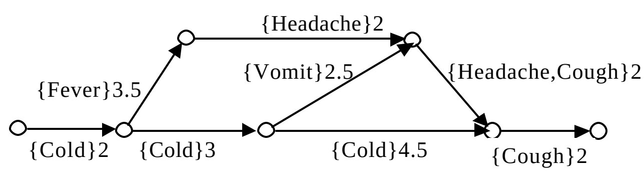

Figure 7. Scenario 2

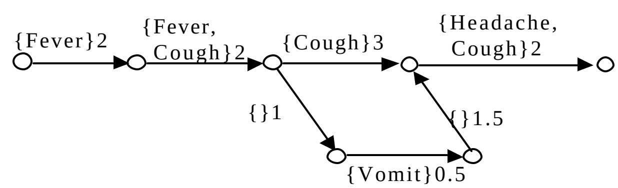

Figure 8. Scenario 3

Taking the above given three scenario graphs as the model graphs, we modified each of them in various ways to obtain a series of modified scenario graphs. Figure 9 shows the relationships between the similarity and the corresponding noise/modification, where each line represent a collection of scenario graphs including one of the 3 model scenarios and the series of the corresponding modified ones.

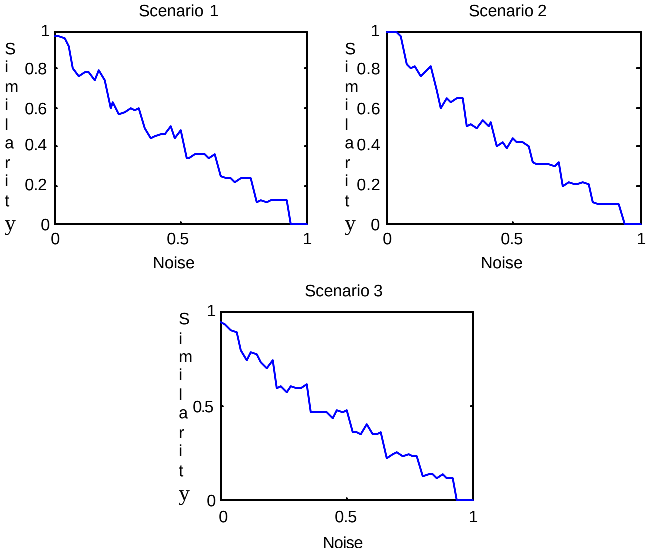

Figure 9. Similarity vs. Noise

Finally, Figure 10 provides an overview of the similarity/dissimilarity among the three collections of scenario graphs generated in Test 2. It demonstrates that the similarity defined here reflects the conventional idea of edit distance in graph matching. In other words, the more similar a pair of two scenario graphs is, the closer they are to each other. Therefore, given some certain criteria in terms of the similarity/dissimilarity, it will be straightforward to use the proposed algorithm to cluster scenario graphs into various groups.

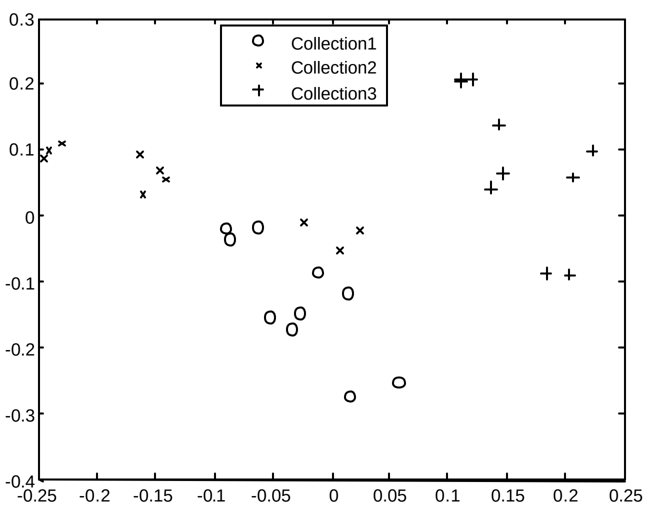

Figure 10. The three collections of scenarios

5 Conclusions

In this paper, we have introduced a framework for representing and recognizing scenario patterns with rich internal temporal aspects. The framework consists of a unified scheme for scenario formalization and a temporal network for graphical representation. It is shown that scenario pattern recognition and matching can be simply transformed into graph matching. However, due to the embedded checking of all permutations, the computational complexity associated with the proposed navigation-based algorithm is actually NP-hard. On the other hand, it is easy to see that, by means of re-indexing the arcs of any given scenario graph, the corresponding *Meets*-adjacent matrix can be turned out to be strictly upper-diagonal. In other words, scenario graphs in general are quite regularly sparse. Therefore, it is believed that exploiting such kind sparsity can lead to more efficient algorithms/methods to improve the corresponding computational complexity. This remains the main target as for the future work. The future work will also concern real world applications such as medical treatments and weather forecasting.

References

[Allen 1981] Allen, J.: An interval-based representation of temporal knowledge. *Proceedings of The 7th International Joint Conference on Artificial Intelligence*, pages 221-226, Vancouver, B.C., Canada, August 1981, William Kaufmann, LA, California.

[Allen 1983] Allen, J.: Maintaining knowledge about temporal intervals. *Communications of the ACM* 26 (11): 832-843, 1983.

[Allen 1984] Allen, J.: Towards a General Theory of Action and Time. *Artificial Intelligence* 23: 123-154, 1984.

[Allen and Hayes 1989] Allen, J. and Hayes, P.: Moments and Points in an Interval-based Temporal-based Logic. *Computational Intelligence* 5: 225-238 1989.

[van Beek 1992] van Beek, P.: Reasoning about qualitative temporal information. *Artificial Intelligence* 58: 297-326, 1992.

[van Benthem 1983] van Benthem, J.: *The Logic of Time*, Kluwer Academic, Dordrech, 1983.

[Bruce 1972] Bruce, B.: A Model for Temporal References and Application in a Question Answering Program. *Artificial Intelligence* 3: 1-25, 1972.

[Dechter et al. 1991] Dechter, R., Meiri, I. and Pearl, J.: Temporal Constraint networks, *Artificial Intelligence* 49: 61-95, 1991.

[Galton 1990] Galton, A.: A Critical Examination of Allen's Theory of Action and Time, *Artificial Intelligence* 42: 159-188, 1990.

[Jare et al. 2002] Jare, M., Aanodt, A. and Shalle, P.: Representing Temporal Knowledge for Case-Based Reasoning. Proceedings of the 6th Euroupean Conference, ECCBR 2002, pages 174-188, Aberdeen, Scotland, UK, September 2003.

[Keravnou 1995] Keravnou, E.: Modelling medical concepts as time-objects. Eds. M. Stefanelli and J. Watts, eds., *Lecture Notes in artificial Intelligence* 934: 67-90, 1995, Springer.

[Knight and Ma 1992] Knight, B. and Ma, J.: A General Temporal Model Supporting Duration Reasoning, *Artificial Intelligence Communication,* Vol.5(2): 75-84, 1992.

[Ladkin 1992] Ladkin, P.: Effective solutions of qualitative intervals constraint problems, *Artificial Intelligence* 52: 105-124, 1992.

[Ma and Knight 1994] Ma, J. and Knight, B.: A General Temporal Theory, *the Computer Journal* 37(2): 114-123, 1994.

[Shoham 1987] Shoham, Y.: Temporal Logics in AI: Semantical and Ontological Considerations, *Artificial Intelligence* 33: 89-104, 1987.

[Ma and Knight 2003] Ma, J. and Knight, B.: Representing The Dividing Instant, *the Computer Journal 46:2*: 213-222, 2003.

[McDermott 1982] McDermott, D.: A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science* 6: 101-155, 1982.

[Nakhaeizadeh 1994] Nakhaeizadeh, G.: Learning Prediction of Time Series: A Theoretical and Empirical Comparison of CBR with Some Other Approaches. In *Proceedings of the Workshop on Case-Based Reasoning, AAAI-94*. Seattle, pages, 67-71, Washington, 1994.

[Shahar 1990] Shahar, Y.: Timing is everything: temporal reasoning and temporal data maintenance medicine. Eds. W Horn et al., *AIMDM'99, LNAI* 1620, pages, 30-46, 1999, Springer Verlag.

[Shanahan 1995] Shanahan, M.: A Circumscriptive Calculus of Events, *Artificial Intelligence* 77: 29-384, 1995.

[Snodgrass, R. T. and Ahn, I: A Taxonomy of time in Databases", in Proceedings of ACM SIGMOD International Conference on Management of Data, Ed. S. Navathe. Association for Computing Machinery, Austin, TX: 236-246, May 1985.

[Vila 1994] Vila, L.: A survey on temporal Reasoning in Artificial Intelligence, *AI Communication 7*: pp.4-28, 1994.

[Vilain 1982] Vilain, M.: A System for Reasoning about Time, *Proceedings of The 1st National Conference on Artificial Intelligence*, pages 197-201, Pittsburgh, Pennsylvania, August 1982, The AAAI Press, Menlo Park, California.

[Theodoridis and Koutroumbas 2003] Theodoridis, S. and Koutroumbas, K.: *Pattern Recognition*. Second Edition, 2003, Academic Press.

[Tveter 1998] Tveter, D.: *The Pattern Recognition Basis of Artificial Intelligence*, 1998, Wiley-IEEE Computer Society Press.