A Schedule/Layout Computation Model

Timothy K. Shih and Anthony Y. Chang
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taiwan
R.O.C.
email: `TSHIH@CS.TKU.EDU.TW`
fax: `Intl. (02) 620 9749`

Abstract

The importance of relations among temporal intervals was found in a wide of disciplines. In the paper, we propos a fast mechanism for temporal relation compositions. A temporal transitive closure table is derived, and an interval-based temporal relation algebraic system is constructed. Thus, we propagate the time constraints of arbitrary two objects across long distances n by linear time. The contributions of these algorithms can be used to generate the schedule and layout of multimedia presentations, to model the interaction of multimedia applications, to analyze the Virtual Reality timing constraints, and to compose multimeida documents.

Key words: Temporal Interval Relations, Spatial/Temporal Model, Multimedia Presentations, Algebraic system, Automatic System

1 Introduction

Temporal relationships are a characteristic feature of multimedia objects. Multimedia refers to the integration of text, image. audio, and video in variety of application environments. These data can be heavily time-dependent, such as audio and video in a motion picture, and can require time-orderd presentation. Composite objects can have arbitrary timing relationships. These might be specified to achieve some particular visual effect of sequence. The time-dependent resources need to be analyzed to ensure that there is no time conflict among resources. Moreover, some of these resources, such as animations or video layouts occupy screen space. The spatial relations among resources need to be computed and represented in the multimedia program. The work discussed in [1] only states temporal interval relations. We found that these relations can be generalized for spatial modeling. The use of spatio-temporal relations serves as a reasonable semantic tool for the underlying representation of objects in many multimedia applications. The contributions of our paper are to give a complete discussion of different possible domains of interval relations based on graph representations. Most importantly, we propose a fast computation mechanism to compute spatio-temporal relations. In section 2, we give an analysis of the domains of temporal relations. In section 3, we propose a generalized properties of spatio-temporal relations. Algorithms for composing relations are introduced in section 4. Issues of realizing an automatic system and other multimedia applications is given in section 5. And finally, our conclusions are given.

2 Spatial/Temporal Domains

In this subsection, we described the symbolic constraint propagation. The general idea is to use the existing information about the relations among time intervals or instants to derive the composition relations. For example, there exists interval or instant events X and Y and Z, if " X before Y " and " Y before Z ", then plainly event X has to be before Z. Allen proposed that there are 13 relations between two temporal intervals. Based on Allen's work , the transitivity table for the twelve temporal relations (omitting "=") showing the composition of interval temporal relations. We use the Time Interval Relations to construct a new 1-D relation composition table. Based on the table, we propose two algorithms in this paper, using the directed graph, for temporal relation compositions. These algorithms can be used to compute the binary relation between an arbitrary pair of intervals.

We analyze the domain of interval temporal relations and use an directd graph to compute the relations of all possibilities : the unknown derivations, the multiple derivations, and the conflict derivations. Some terms used in the graph are defined as the following:

0-8186-8073-3/97 $10.00 © 1997 IEEE

Definition: An user edge denotes a relation between a pair of objects defined by the user. The relation may be reasonable or non-reasonable. ■

Definition: A derived edge holds a non-empty set of reasonable relations derived by our algorithm. The relation of the two objects connected by the derived edge can be any reasonable relation in the set. ■

For each pair of objects in the time line, there exists a set of possible binary relations held between the pair of objects. For an arbitrary number of objects (denoted by nodes), some of the relations (denoted by edges) are specified by the user while others are derived. If there exists a cycle in the directed relation graph, a conflic derivation may occur. If there exists no cycle, there is no conflict. However, there may exist an unknown derivation which represents that there is no enough information to derive a relation between a pair of objects. Based on the above considerations, we suggest that the computation domain reveals four types, as discussed below.

- The complete relation domain (a complete graph) : contains user edges and derived edges , with possible cycles and possible conflicts.
- The reasonable relation domain (a graph) : contains user edges and derived edges, with possible cycles but no conflict.
- The reduced relation domain (a graph) : contains only user edges, with possible cycles and possible conflicts.
- The restricted relation domain (a tree) : contains only user edges, without cycle.

The four domains are used in the analysis and computation of object relations. In section 4, we propose two algorithms computing the reasonable relation domain.

3 Finite Relations Group

In Allen's temporal transitive table, a composition of two temporal relations results in a set of relations. Compositions of three or more relations are computed using algorithms based on set operations, such as set union and intersection. These set operations are expensive. We argue that, an extension of *Table*13 (Allen's Table), named *Table*29, can be calculated. The compositions of three or more relations can be obtained directly from our table. Algorithm *ComputeTable*29 calculates *Table*29, which consists of the compositions of 29 temporal relation sets. Firstly, we define some terminology used in our discussion.

$$Name == \mathbb{P}\, string$$
$$13Rel == \{ <, >, d, di, o, oi, m, mi, s, si, f, fi, e \}$$

$$<^{-1} = > \;\wedge\; d^{-1} = di \;\wedge\; o^{-1} = oi \;\wedge\; m^{-1} = mi \;\wedge$$
$$s^{-1} = si \;\wedge\; f^{-1} = fi \;\wedge\; e^{-1} = e$$

$$29RelSet \subset \mathbb{P}\, 13Rel$$
$$\forall\, rs : 29RelSet \bullet rs^{-1} = \{ r^{-1} \in 13Rel \mid r \in rs \}$$

$$TemporalTuple == Name \times 29RelSet \times Name$$
$$\forall\, tt : TemporalTuple \bullet$$
$$\quad tt = (A, rs, B) \Leftrightarrow tt^{-1} = (B, rs^{-1}, A)$$

$$\circ : TemporalTuple \times TemporalTuple \rightarrow$$
$$\quad TemporalTuple$$
$$\forall\, tt_1, tt_2, tt_3 : TemporalTuple \bullet tt_1 = (A, rs_1, B) \wedge$$
$$\quad tt_2 = (B, rs_2, C) \wedge tt_3 = (A, rs_3, C) \bullet$$
$$\quad tt_1 \circ tt_2 = tt_3 \Leftrightarrow$$
$$\qquad (A = C \wedge rs_2 = rs_1^{-1} \Rightarrow rs_3 = \{ e \} \vee$$
$$\qquad A = C \wedge rs_2 \neq rs_1^{-1} \Rightarrow rs_3 = \perp \vee$$
$$\qquad A \neq C \Rightarrow rs_3 = Table29(rs_1, rs2))$$

An interval has a name, which is a string. The term $\mathbb{P}(X)$ represents a power set of object of type X. The 13Rel is the domain of the 13 interval relations defined by Allen. The 29*RelSet* is a domain of relation sets. Each element in 29*RelSet* contains one or more interval relations which represent the possible results between two intervals. The following table gives a summary of the 29 relation sets which contain all possible composition results: The 29 Relation Sets

The 29 Relation Sets

ID	Relation Sets
1	$\{ < \}$
2	$\{ > \}$
3	$\{ d \}$
4	$\{ di \}$
5	$\{ o \}$
6	$\{ oi \}$
7	$\{ m \}$
8	$\{ mi \}$
9	$\{ s \}$
10	$\{ si \}$
11	$\{ f \}$
12	$\{ fi \}$
13	$\{ e \}_*$
14	$\{ o, di, fi \}$
15	$\{ oi, d, f \}$
16	$\{ o, d, s \}$
17	$\{ oi, di, si \}$
18	$\{ <, o, m \}$
19	$\{ >, oi, mi \}$
20	$\{ f, fi, e \}_*$
21	$\{ s, si, e \}_*$
22	$\{ <, o, m, d, s \}$
23	$\{ >, oi, mi, di. si \}$
24	$\{ <, o, m, di, fi \}$
25	$\{ >, oi, mi, d, f \}$
26	$\{ o, oi, d, di, s, si, f, fi, e \}_*$
27	$\{ <, m, d, di, o, oi, f, fi, s, si, e \}$
28	$\{ >, mi, di, d, oi, o, fi, f, si, s, e \}$
29	$\{ <, >, d, di, o, oi, m, mi, f, fi, s, si, e \}_*$

*Table*29 is generated by our program implemented based on the following algorithms.

```
$Algorithm : RelComp$
$Input : rs_1 \in 29RelSet, rs_2 \in 29RelSet$
$Output : rs \in 29RelSet$
$Preconditions : true$
$Postconditions : true$
$Steps :$
1 : $rs = \bigcup_{\forall r_1 \in rs_1, \forall r_2 \in rs_2 \bullet (r1, r2) \in rs_1 \times rs_2}$
        $Table13(r_1, r_2)$
```

In Function *RelComp*, the reasonable computed must be the union of all possible combinations of the pair of relations obtained from the two input relation sets, name rs1 and rs2 . The function uses a table lookup function to obtain a set of relations. *Table*13 uses the relation composition table discussed in [1], if the algorithm is to compute relations of objects in a 1-D space.

```
$Algorithm : ComputeTable29$
$Input : Table13$
$Output : Table29$
$Preconditions : true$
$Postconditions$ : relation composition is closed under $I$
$Steps :$
1 : Construct a set of 13 atomic sets from the
      13 relations, assuming that this set is called
      $I$, which is an index set for table look up
2 : Let $Table29(i, j) = Table13(i, j), i \in I, j \in I$
3 : $\forall\ Table29(i, j), i \in I, j \in I, do$
      2.1 : if $k = Table29(i, j) \notin I$ then
            2.1.1 : $I = I \cup Table29(i, j)$
            2.1.2 : $\forall\ m \in I, do$
                  2.1.2.1 : $Table29(k, m) = RelComp(k, m)$
                  2.1.2.2 : $Table29(m, k) = RelComp(m, k)$
```

Algorithm *ComputeTable*29 adds new relation sets computed by RelcCmp to the index set I, and computes the new elements of *Table*29. There are

$$C(13, 0) + C(13, 1) + C(13, 2) + ... + C(13, 13) = 2^{13}$$

possible elements of I. However, from the computation of algorithm *ComputeTable*29, the cardinality of I results in 29. Based on this result, we argue that, for an arbitrary pair of temporal intervals, the possible relations between them must be an element of set I. Using *Table*29, when composing temporal relations (see section 4), the set union operation is replaced by a table look up operation. Therefore, the time complexity of relation composition is reduced. The cost of memory used in *Table*29 is tolerable.

4 Relation Inference

We discuss two algorithms for computing the reasonable relation domain, which contains no conflict and thus can be used for generating schedule and layout of a presentation. We can easily compute the temporal relation constraints and detect all possible conflicts to maintain knowledge by using *Table*29.

```
$Algorithm : ComputeRD1$
$Input : G = (GV, GE)$
$Output : K_n = (K_n V, K_n E)$
$Preconditions : true$
$Postconditions : GV = K_n V \wedge GE \setminus UE \cup UE' \subseteq K_n E$
$Steps :$
1 : $G = EliminateConflicts(G)$
2 : $K_n = G \wedge pl = 2$
3 : repeat until $\mid K_n E \mid = \mid K_n V \mid * (\mid K_n V \mid -1) \ / \ 2$
      3.1 : for each $e = (a, b) \wedge e \notin K_n E \wedge a \in K_n V \wedge$
            $b \in K_n V$such that there is a path of user
            edges from $a$ to $b$, with path length $= pl$
      3.2 : suppose $((n_1, n_2), (n_2, n_3), ..., (n_{k-1}, n_k))$
            is a path with $a = n_1 \wedge b = n_k \wedge k = pl + 1$
      3.3 : set $e.rs = Table29((a, n_{k-1}).rs, (n_{k-1}, b).rs)$
      3.4 : $K_n E = K_n E \cup \{\ e\ \}$
      3.5 : $pl = pl + 1$
```

```
$Algorithm : EliminateConflicts$
$Input : G = (GV, GE)$
$Output : G' = (G'V, G'E)$
$Preconditions$ : $G$ contains only user edges $\wedge\ G' = G$
$Postconditions$ : $G' = G$, but the reasonable sets of
                  edges in $G'$ may be changed
$Steps :$
1 : for each $P = ((n_1, n_2), (n_2, n_3), ..., (n_{k-1}, n_k))$ in $G'$,
      with $n_1 = n_k \wedge k > 3$
      1.1 : for each $i, 1 \leq i \leq k - 2$
            1.1.1 : set $(n_i, n_{i+2}).rs =$
                  $Table29((n_i, n_{i+1}).rs, (n_{i+1}, n_{i+2}).rs)$
      1.2 : $rs = Table29((n_k, n_{k-2}).rs, (n_{k-2}, n_{k-1}).rs)$
      1.3 : if $(n_k, n_{k-1}).r \notin rs$ then
            1.3.1 : ask user to choose an $r' \in rs$
            1.3.2 : set $(n_k, n_{k-1}).r = r'$
```

The purpose of the first algorithm is to add derive edges to the reduced relation domain. If there is a conflict cycle in the original reduced relation domain, the algorithm eliminates that conflict first by altering the user to select a reasonable relation to replace the original one (i.e. UE). This is why the resulting graph may contain some new user edges (i.e. UE'). Thus, the resulting reasonable relation domain is a complete graph, which is equal to the complete relation domain. This conflict elimination is achieved by involking the EliminateConflicts algorithm. Supose G is a graph of the reduce relation domain, and GV and GE are the vertex set and edge set of G. Initially the reasonable relation domain is set to the reduced relation domain.

The first algorithm, ComputeRD1, starts from taking each path of user edges of length 2. and computes a derived edge from that path. The insertion of edge e = (a, b) results a cycle, but no conflict. The reasonable set of edge e is computed from two edges, $(a, \ nk_1 \ )$

and (nk_1, b), which are user edges or derived edges. Since we increse the path length, pl, of the path of user edges one by one. The derived edge (a, nk_1) (or user edge, if pl = 2) must have been computed in a previous interation. The algorithm repeats until all edges are added to the complete graph K_n , which contains n * (n-1) / 2 edges.

5 The Applications

Spatial/Temporal relations can be used in many multimedia related applications. Using our proposed algorithms in this paper, inference relus can be generalized to generate a better presentation. The generation of presentation schedule includes the following steps:

- compute temporal relations among presentation resources. Possible conflicts are eliminated by asking the user to give a correct relation.
- use a partial order set to denote the topological order of presentation objects.
- generate a relative time table for presentation objects from the partial order set.
- allocate multimedia devices for each resources. Hardware limitations are considered.

The mechanism proposed in this paper can also be used in generating multimedia presentation layouts. As long as the spatial relations of objects are decided, the algorithm can compute the location of each presentation resource in a window.

6 Conclusions

In this paper, we analyze temporal interval relations and propose four domains for relation composition. We provide a set of algorithms for the automatic generation of multimedia presentation from temporal and spatial relations among multimedia resources. Possible conflicts of relations are eliminated.

The main contributions of this paper is in a set of algorithms for spatio-temporal relation composition. These algorithms deal with an arbitrary number of objects in an arbitrary n-dimensional space. We also argue that, many interesting researches in multimedia applications can benefit from using these spatio-temporal relations and our algorithms.

The algorithm proposed in this paper can be used in other computer applications. We hope that, with our analysys and aigorithms, the knowledge underlying temporal interval relations can be used in many computer applications.

References

[1] James F. Allen. "Maintaining Knowledge about Temporal Intervals," Communitions of the ACM , Vol. 26 , No. 11 , 1983.

[2] Chi-Ming Chung, Timothy K. Shih, Jiung-Yao Huang, Ying-Hong Wang, and Tsu-Feng Kuo, "An Object-Oriented Approach and System for Intelligent Multimedia Presentation Designs," in proceeding of the Internation Conference on Multimedia Computing and Systems (ICMCS 95) , Washington DC, U.S.A. May 15-18,1995, pp 278-281.

[3] Young Francis Day, et. al. "Spatio-Temporal Modeling of Video Data for On-Line Object-Oriented Query Processing," in proceedings of the International Conference on Multimedia Computing and Systems, Washington DC,U.S.A., May 15-18 ,1995 , pp 98-105 .

[4] Cherif Keramane and Andrzej Duda. "Interval Expressions -a Function Model for Interactive Dynamic Multimedia Presentations," in proceedings of the 1996 Internation Conference on Multimedia Computing and Systems , Hiroshima, Japan, June 17-23, 1996 , pp 119-133.

[5] John Z. Li, M. TamerOzsu, and Duane Szafron, "Spatial Reasoning Rules in Multimedia Management Systems," in proceedings of the 1996 Multimedia Modeling internation conference (MMM96),Toulouse, France, Novemner 12-15,1996,pp 119-133.

[6] Thomas D. C. Little and Arif Ghafoor. "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," IEEE Computer, October 1991, pp 42-50.

[7] Thomas D. C. Little and Arif Ghafoor. "Interval Based Conceptual Models for Time-Dependent Multimedia Data," IEEE transactions on knowledge and data engineering , Vol. 5, No. 4,1993, pp 551-563.

[8] Timothy K. Shih, Steven K. C. Lo, Szu-Jan Fu, and Julian B. Chang. "Using Interval Temporal Logic and Inference Rules for the Automatic Generation of Multimedia Presentations," in Proceedings of the IEEE Internation Conference on Multimedia Computing and Systems, Hiroshima, Japan, June 17-23, 1996, pp. 425-428.

[9] Timothy K. Shih, Chin-Hwa Kuo, Huan-Chao Keh, Chao T. Fang-Tsou, and Kuan-Shen An. "An Object-Oriented Database for Intelligent Multimedia Presentations," in proceedings of the 1996 IEEE Internation Conference on Systems, Man and Cybernetics, Beijing, China, October 14-17, 1996.

[10] Michael Vazirgiannis, Yannis Theodoridis, and Timos Sellis. "Spatio-Temporal Composition in Multimedia Applications," in proceedings of the Internation Workshop on Multimedia Software Development, March 25-26, Berlin, Germany, 1996, pp 120-127.