中華民國八十六年全國計算機會議

利用時序推理方法解決簡報資源排程衝突
Solving Presentation Resource Schedule Conflicts A Temporal Reasoning Approach

施國琛　　羅光志　　王俊傑
Timothy K. Shih, Steven K. C. Lo, Ken C. Wang
淡江大學資訊工程研究所
Multimedia Information Network (MINE) Lab
Department of Computer Science and Information Engineering of Tamkang University
email: tshih@cs.tku.edu.tw

摘要

在此論文中，我們提出解決時間性與空間性資源同步的問題，而此方法與理論基礎正是針對多媒體資源在時序上的關係提出相關的研究與實作。首先在理論部份，針對所提出的簡報規格需求與推論規則相結合，來自動產生多媒體簡報系統。此外，我們也實作了一個推理引擎，其中使用者可以對推理規則做合理性的修正，來適合自己的需求。另外我們也提出了一個演算法來解決簡報資源在排程上所產生的衝突問題。使用這個推理規則與推理引擎不僅會降低簡報設計者的負擔而且還會提供很大的幫助來指導使用者使用這套多媒體展示系統。

關鍵字：多媒體簡報，推理引擎，遞降遞歸分析法

Abstract

A presentation specification contain three parts: the resources information, the temporal information and spatial information. Different relationships among the temporal intervals can be used to help the automatic generation of multimedia presentations. I propose a mechanism for the automatic generation of interactive multimedia presentations from their specifications and use a recursive descent parsing technique to implement the inference rules. The inference rules are based on interval temporal logic and important issues in multimedia presentations. I also propose an algorithm (MES) to solve the multimedia presentation resource schedule conflicts. We use the inference and algorithm to design a reasoning system to automatic generate the schedule. Our system running on Microsoft Windows 95.

Keywords：Multimedia Presentations, Temporal Intervals Relations, Recursive Descent Parsing

1. Introduction

The role that the computer acts nowaday society has transformed from simply science application to variable and businessful machine. "Multimedia" has become the most important development in computer technology field. It not only acts a more and more important role in human culture activities but also expends speedly which affects everyone's daily life and sense world. For this sake, when we want to exhibit our work piece or express some ideas or notions, we always consider about using "Multimedia" (composed of text, images, graphics, audio, animation and video etc.) as the present system[7,8,9]. In the market are many relevant multimedia presentation systems which provide users way to produce vivid presentation. But when programmers design the presentation systems, they always think about how to handle the problems of temporal and spatial resources Therefore, it becomes important to consider of relations of the temporal

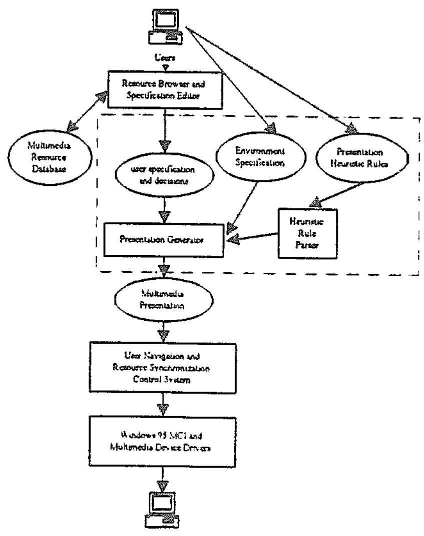

Figure1:A System for Multimedia Presentation Designs

resources, especially you can use the rules you infered to solve mutual relationship in different multimedia resources and the whole playback schedules. Besides, when users want to make reasonable modification to the inference rules so to correspond to their own requirements. They will not have to modify the main programs but simply modify the text files of the inference parts to reach what they request. These parts are according to the methods of the recursive descent analysis to accomplish the implementation. It will not only reduce the programmer's burden but also provide a great assistance to use the multimedia presentation system. Figure 1 shows the system for multimedia presentation designs base on temporal inference rules. It describes clearly the parts of theory and implement.

2. Temporal Operators and Functions

In order to show the temporal information from multimedia resource database, it is necessary to propose a time model. On the time model of our multimedia presentations system contain some continue temporal intervals[3,10], and show the embeded resource information from the temporal intervals. So our inference engine has to define a set of operators and functions to express the information among the temporal resources.We define some base operators and functions about temporal logic nowadays in order to represent the temporal information of multimedia resources .So, structuring the temporal model with temporal operators and functions is necessary. We define seven kinds of operators to represent the base acts of time intervals. The temporal operators and functions as following specify.

Thus, a number of temporal operators and functions are define in our inference rules for the representation of resource temporal information. The usage of those operators/functions and some examples show as following:

- ◇ **concatenation:** "^", is connect two sequential resource streams.
- ◇ **silent:** the operator is "_", while applying to a number, denotes a silent stream of many cycles.For instance, "_15" is a stream of no action which lasts 15 cycles.
- ◇ **extension:** the extension operator "~", extends a resource stream, according to the synchronization extent of the resource.
- ◇ **truncate:** the truncate operator can be used in two ways. "r 15!" says the resource r is played for the first cycles only. And "!15 r" denotes r is played after cutting the first 15 cycles.
- ◇ **concurrent:** the concurrent function, "$", is an overloaded function accept one or more parameters.
- ◇ **sequential:** the sequential function, "-", the resources are present one by one. There is no semantic difference between the sequential and the concatenation operator. However, sequential functions serve as the principal functors of the final representation of a multimedia presentation. The concatenation operators are used in the intermediate process, or used as parts of the final representation of a presentation.
- ◇ **identical:** the identical function "#", is similar to the concurrent function with a further restriction indicates that all resources end at the same as well.

3. Multimedia Presentation Specifications

In this section, we propose some multimedia presentation specifications for the automatic generating the multimedia presentations. We discuss the concern later between the multimedia presentation specification statements and inference rules . If we do not think about the consequence, it will be difficult to be used directly by the presentation user.

Multimedia Resources Properties

Multimedia resource are usually recorded or captured via camera, tape recorder, or video camera, converted to their digital formats, and saved on the disk. These resources are not only provide the users to access, but also share the resource files to others for different presentation software. For every one of the resource exist some properties. The following illustrate will represent the resources properties.

- **name:** name of the resource.
- **keyword:** one or more keywords are used as the description of a multimedia resource.
- **usage:** how the resource is used.
- **medium:** what multimedia device is used to carry out this resource.
- **model:** how the resource is presented
- **temporal endurance:** how long does the resource last in a presentation.
- **synchronization tolerance:** how does a particip-ant feel about the synchronization delay of a resource.
- **detectability:** how does the resource attract a listener.
- **hardware limitation:** what kind of hardware is essential for carrying out the resource.
- **date:** the date and time this resource file is created.
- **resolution:** the resolution of this resource file,

specified by X×Y screen units.

start/end time: for non-permanent resources, the starting cycle and the ending cycle of some resource play.

For above descriptions, the properties of multimedia resources apply to the multimedia database. Some of them are alluded in my multimedia presentation specifications. According to these attribute, we use the below statement to explain that we are more care in our presentation.

Multimedia Resources Relations and Priority

After discussing the resource properties, we have to consider about the relations among the resources. A multimedia presentation consists of multiple resources streams. In some condition, any two resources are not appropriate to played at the same time, for instance, to watch two video simultaneously. Or, according to the presentation experiences, some resource streams will make the good effect to present concurrently. For instance, it is nice to have a MIDI music background for an animation play. Our approach is to reduce the burden of the user by given these good suggestions. However, if the user is not agree our suggestions, they can still change the final presentation what he/she want. So given any two type of resources in a specification, we can consider their relations as following.

- **mutual exclusive(represented by an X):** any two resources can not be played at the same time.
- **possible concurrent(represented by a ?):** any two resources could be played at the same time, or not be played concurrently. It depends on users to choice the possible relation what he want.
- **mutual inclusive(represented by an O):** any two resources can be played at the same time.

If two types of multimedia resources want to present in a same interval and the hardware is able to carry out only one, then we must decide which one that have higher priority to play. The use of priority is to assist the inference to make a better presentation. The higher number decides the higher priority.

In general, dynamic resources such as sound and video, have higher priorities since they can easily catch the attention of the user. However, the resource detectability is considered in multimedia presentation priority calculations.

3.3 Temporal Specification

James F. Allen in [3] proposes thirteen types of relations between temporal intervals. Although the thirteen relations cover all possible condition, but we can only realize some time information. For multimedia presentation synchronization, it needs precise timing information of multimedia resources. Thus, some modifications to the relations are necessary in order to achieve multimedia resources synchronization. The following specified the revised relations from thirteen types of the relations. We add the time parameter to make the relation representations abundant.

- always(r1, n): r1 present for n cycles.
- meets(r1, r2): r2 is presented after r1 finishes.
- before(r1, r2, n): r2 is presented n cycles after r1 finishes.
- starts(r1, r2): r1 and r2 are synchronized at the beginning.
- finishes(r1, r2): r1 and r2 are synchronized at the end.
- overlaps(r1, r2, n): r1 starts first, r2 starts n cycles after r1 starts, r1 ends before r2.
- embraces(r1, r2, n): r1 starts first, r2 starts n cycles after r1 starts, r2 ends before r1.
- equal(r1, r2): r1 and r2 present the same duration concurrently.
- simultaneous(r1, n1, r2, n2): the n1th cycle of r1 and the n2th cycle of r2 happen at the same time.

In order to reduce the load of the user by means of automation. We try to make the specification statements as general as possible while still maintaining all possible temporal relations between two multimedia resources. The specification statements as following.

- always(r1, n): r1 present for n cycles.
- sequential(r1, r2, n): r2 is presented n cycles after r1 finishes.
- intersects(r1, r2, n): r1 and r2 intersects each other. r1 starts first, and r2 starts after n cycles.
- synchronizes(r1, n1, r2, n2): the n1th cycle of r1 and the n2th cycle of r2 are synchronized.

The time arguments are provided by the user and the temporal endurance of resources can be obtained from the database, so we can make a comparison between specification statements and revised relations.

4. The Temporal Specification Inference Rules

We infer the rules according to section 2 and section 3 before. We can use the inference rules to automatic generating temporal relation between any two multimedia resources each other. The generating relation not only make the multimedia presentation powerful and useful but also become the base theory of temporal knowledge. The inference rules represent as following

Temporal Inference Rules

```
always(r1,n)        if r1.TE >= n
                     then   -(r1 n!)
                     else   -(r1~n-r1.TE)

overlaps(r1,r2,n)   if r1 X r2
                      then
                        if r1 >> r2
                          then -(r1,!r1.TE-n r2)
                          else -(r1 n!,r2)
                      else
                        if r1 >> r2
                          then
                            if r1.TE > r2.TE
                              then   $(r1~r2.TE+n-
r1.TE,_n^r2)
                              else   $(r1,_n^r2
r1.TE!)
                          else
                            if r1.TE > r2.TE
                              then   $(r1~r2.TE+n-
r1.TE,_r1.TE-r2.TE^r2)
                                     | $(!r1.TE-r2.TE
r1~r2.TE+n-r1.TE,r2)
                              else   $(_r2.TE-
r1.TE^r1~r2.TE+n-r1.TE,r2)
                                     | $(r1~r2.TE+n-
r1.TE,!r2.TE-r1.TE r2)
```

5. Use Recursive Descent Analysis to Implement the System

In order to make our system with the extension future, I will use the well-known parsing technique called recursive descent to implement the inference rules. When user want to update the rule file to conform himself, but do not change the system program. Using the recursive descent analysis is necessary. At first, we define the grammar text according to the inference rules as following description.

Grammar:

```
<TRule> ::= <TSpec> <TExp>
<TSpec> ::= always(<R>,<N>)
          | meets(<R>,<R>)
          | before(<R>,<R>,<N>)
          | starts(<R>,<R>)
          | finishes(<R>,<R>)
          | overlap(<R>,<R>,<N>)
          | embrace(<R>,<R>,<N>)
          | equal(<R>,<R>)
          | simultaneous(<R>,<N>,<R>,<N>)
<TExp> ::= <IfExp>
         | <TStmt>
         | <TSpec>
         | <TSpec> || <TExp>
         | <TSpec> && <TExp>
         | Terror
<TStmt> ::= -(<TStmtParm>)
          | $(<TStmtParm>)
<TStmtParm> ::= TStr
              | TStr,<TStmtParm>
<R> ::= r
      | r1
      | r2
      | <X>
<X> ::= r1.TE
      | r2.TE
      | <N>
<N> ::= n
      | n1
      | n2
<IfExp> ::= if <Cond> then <TExp> else <TExp>
<Cond> ::= <R> <Comp> <R>
<Comp> ::= > | >= | X | O | >>
```

We scanning the tokens with input files according to the grammars, and build the binary tree to store those. At last, we match the data from internal nodes and leaf nodes in the binary tree. Our input files have four parts: the inference rules file, the relations table file, the priority table file, and the 4 kinds of statements convert to the 9 kinds of revised relations file.

6. Multiple Resources Mutual Exclusion

When the multimedia presentation generates, multiple resource streams usually play currently. To due to the limitation of hardware, they may make those resources mutual exclusive in the same intervals. For solving to the condition, system has to decide only one from these resources. Our inference rules solve only the mutual exclusive between two resources, so we provide one method to improve the rules. At first, we define start time and end time for every resource. When some resources have the same interval between them, we must consider the modify of start/end time and temporal endurance. Then I propose one algorithm to solve the mutual exclusive called MES(Mutual Exclusive Solving). If we find the same duration according to the new revised relation coming in, it will be processed by the MES algorithm. The MES algorithm mainly deal with the overlap duration whether mutual exclusive.

MES Algorithm:

Revised_Temporal_Relations(r_j [, n1] [, r_n [, n2]])
 for (i=1 ; i<n ; i++)
 { if $i \neq j$ and $r_i \, X \, r_n$
  then
   if $r_i.st \leqq r_n.st < r_i.et$ or
    $r_i.st < r_n.et \leqq r_i.et$ or
    ($r_i.st \geqq r_n.st$ and $r_i.et \leqq r_n.et$)
    then
     intersects(r_i , r_n , $n_{(new)}$) }

r_i : original i-th resource
r_n : new enter resource
Revised_Temporal_Relations() : call revised relation function of the inference rules
$r_i.st$: start time of r_i
$r_i.et$: end time of r_i
$n_{(new)}$: $r_n.st - r_i.st$

After using the algorithm, the inference will be change. I will take one of the re-infer rules to represent as following.

always(r1,n) if r1.TE >= n
  then
   -(r1 n!)
   $r1.TE_{(new)} = n$
   $r1.st_{(new)} = r1.st$
   $r1.et_{(new)} = r1.st+n$
  else
   -(r1~n-r1.TE)
   $r1.TE_{(new)} = n$
   $r1.st_{(new)} = r1.st$
   $r1.et_{(new)} = r1.st+n$

7. Multimedia Presentation Generation System

A presentation specification is a set of temporal relations among multimedia resources with the spatial information for those resources. Even presentation specifications have their representations and intermediate forms, it is quite difficult for a presenter who is not familiar with computer programming to design a presentation using presentation specification statements. For the convenience of the user, we develop a multimedia presentation generation system as well as its graphical user interface.

When a presentation designer is about to design his/her presentation, after the presentation topics and a rough schedule are designed, the designer will collect resources, convert them to some digital forms, and store these resources with the properties discussed in section 3.1 in our multimedia database. Figure 2 shows a resource browser user interface allows the presenter to choice or change resources, and to retrieve multimedia resources according to properties given in separated text or list boxes. When these properties are decided, the user is able to push the Select button which makes a number of resource icons shown in the Resource icons subwindow. The Video, the Picture/Text/Animation, and the Sound/MIDI subwindows are to preview these resources selected before the user pushes the Use button to add these resources to a presentation. These resources, while passed to our presentation generator, are represented in the resource specification format discussed in section 3.2.

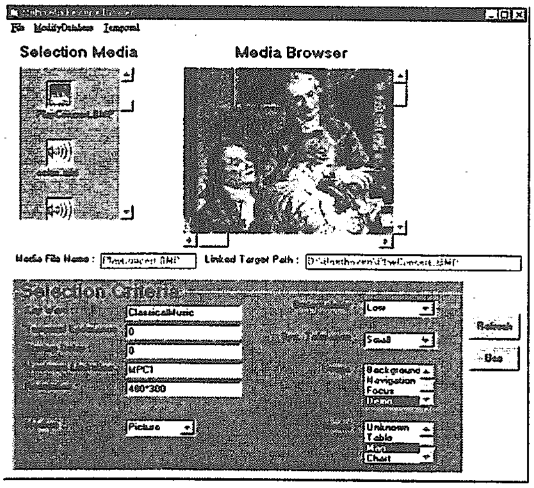

Figure2 : Multimedia Resource Browser

A box with four surrounding thick lines in a icon represents a place holder for a multimedia resource (indicated by its name "r", "r1", or "r2"). A box with thin lines is for the user to fill in a timing parameter (shown as "n", "n1", or "n2"). After the user selects the resources that he/she plans to use from our multimedia resource browser, in figure 3, when the user pushes one of the temporal icons listed in the tool bar, the user is able to drag and place the temporal icon in the Temporal Specification Design area. Next, the user has to fill in the resources and timing parameters by dragging resource icons from the

Multimedia Resource area and typing in integers in the temporal icon. After a temporal icon has all of its parameters filled, the icon is highlighted. The user thus can drag this temporal icon to the temporal specification deposit area which contains the final presentation schedule design.

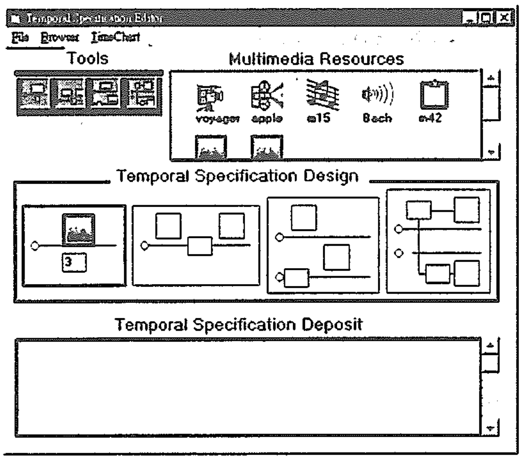

Figure3 : Temporal Specification Editor

8. Conclusion

We propose a mechanism for automatic generation of multimedia presentation. The mechanism use the temporal intervals to infer the rules. In this paper, some issues are important. First, inferring temporal rules with specifications. It not only can apply to the multimedia presentation but also can apply to the information science. On the other hand, it is important to consider relations between temporal resources. Especially, these inference rules are used to solve the mutual relationship conflict problem among different multimedia. Besides, users can make reasonable modifications to the inference rules to fit their own requirement. Using the inference rules and inference engine, the system not only reduces the presenter's burden but also provide a great guidance to use the multimedia presentation system.

Consequently, our contributions in the paper are: firstly, we propose four useful temporal specification statements by showing the mapping between the statements and the interval temporal relations. Secondly, a number of inference rules are developed and propose the MES algorithm that will solving the multiple resources conflict problem in the presentation schedule. Thirdly, use the recursive descent parsing technique to implement the inference rule, and a ICON programming interface is designed for the convenience of the user.

9. References

[1] Yahya Y.Al-Salqan, et. al., "MediaWare: On Multimedia Synchronization" in proceedings of the international conference on multimedia computing and systems, Washington DC, U.S.A.,May 15-18, 1995, pp 150-157.

[2] D.S. Backer, " Multimedia Presentation and Authoring," In Multimedia Systems, edited by J. F. K. Buford, ACM Press, pp 285-303, 1994.

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

[4] B. Prabhakaran and S. V. Raghavan, " Synchronization models for multimedia presentation with user participation," Multimedia Systems, Vol. 2, pp 53-62, Springer-Verlag, 1994.

[5] James Schnepf, et. al., "Doing FLIPS: Flexible Interactive Presentation Synchronization " in proceedings of the international conference on multimedia computing and systems, Washington DC, U.S.A., May 15-18, 1995, pp 213-222.

[6] Chi-Ming Chung, Timothy K. Shih, Chin-Hwa Kuo, and Ying-Hong Wang, "On the Construction of Intelligent Multimedia Presentations," accepted for publication in Information Sciences: An International Journal, USA, 1996.

[7] Chi-Ming Chung and Timothy K. Shih, "On Automatic Generation of Multimedia Presentations," accepted for publication in Information Science: An International Journal, USA, 1997.

[8] Timothy K. Shih, " An Artificial Intelligent Approach to Multimedia Authoring, " In proceedings of the second IASTED/ISMM international conference on distributed multimedia systems and applications, Stanford, California, August 7-9, pp 71-74, 1995.

[9] Timothy K. Shih, "On Making a Better Interactive Multimedia Presentation," in Proceedings of the International Conference on Multimedia Modeling, Singapore, 1995.

[10] 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" IEEE International Conference on Multimedia Computing and Systems, Hiroshima, JANPAN, June 17-23, 1996, pp 425-428.