Using Interval Temporal Logic and Inference Rules for the Automatic Generation of Multimedia Presentations

Timothy K. Shih, Steven K. C. Lo, Szu-Jan Fu, and Julian B. Chang
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taiwan
R.O.C.
email: TSHIH@CS.TKU.EDU.TW
fax: Intl. (02) 623-8212

Abstract

We propose a mechanism and system for the automatic generation of interactive multimedia presentations from their specifications. A presentation specification contains three parts: the resources, the temporal information, and the spatial information. We use a ICON programming technique and a graphical user interface allow the presentation designer to quickly specify what he/she wants. Our system takes these requirements and relies on inference rules written in Prolog to generate interactive presentations. These inference rules are based on interval temporal logic and important issues in multimedia presentations, such as the hardware limitations of an ordinary personal computer and the properties of a multimedia resource. Our prototype system run under MS Windows. The early experience of using the system shows that it is feasible to use logic inference rules to assist the design of good multimedia presentations.

1. Introduction

As multimedia technologies largely increase communication effectiveness between human and computers, the importance of efficient multimedia authoring tools brings the attention to both researchers and software venders. Many presentation or authoring tools were developed for presenters or artists in various fields. Some researchers developed domain specific presentations using artificial intelligence techniques. For example, COMET (COordinated Multimedia Explanation Testbed) [3] uses a knowledge base and AI techniques to generate coordinated, interactive explanations with text and graphics that illustrates how to repair a military radio receiver-transmitter. We also proposed a system [2, 7, 8, 9] that uses object-oriented techniques to incorporate Expert System inference mechanisms into multimedia presentation designs. The system supports interactive designs of presentation layouts and navigation, incorporated with an underlying inference and learning system that deduces useful outputs for intelligent presentations. In this paper, we propose some new results of our research, especially in multimedia inter-stream synchronizations. Articles related to solving multimedia synchronization problems using interval temporal relations/temporal logic includes those discussed in [1, 5, 4]. Other researchers [6, 4] use timed Petri nets. Our approach, based on the thirteen relationships between two time intervals proposed in [1], considers multimedia resource properties as important issues in order to make good multimedia presentations. Other authoring systems discuss synchronization of multimedia streams can be found in [10].

The purpose of our presentation generator is to reduce a designer's load as much as possible while still allowing the designer to make high quality presentations. We support the stepwise refinement of a presentation design. As long as the user knows the relation between two resources, he/she can start the initial design. Other resources are added one by one to the initial design. We also provide an efficient mechanism for the user to precisely control the synchronization points by means of a "synchronizes" specification statement. Multimedia presentations can be designed in our proposed specification language. Or, for the convenience of the user, we also develop a ICON programming graphical user interface. A multimedia presentation specification, according to our system, contains three parts:

- *Resource Specification* carries information of multimedia resources to be used in the presentation, which is obtained from our multimedia resource database via a resource browser.
- *Temporal Specification* describes the temporal relations among resources, which is specified in a predicate format.
- *Spatial Specification* provides the layout of a presen-

0-8186-7436-9/96 $5.00 © 1996 IEEE
Proceedings of MULTIMEDIA '96

tation.

Our presentation generator takes as input the above specifications, uses logic inference rules defined in our system, and generates presentation implementation information for a *presentation frame* [2, 8] (e.g., screen layouts, presentation schedule, or possible error diagnosis). A presentation contains a number of presentation frames. These frames activate each other via message passing. The navigation of a presentation is based on the user's interaction which introduces messages. The interaction, as well as the temporal synchronization requirements and the spatial layouts of multimedia resources, are the three important elements construct a multimedia presentation.

This paper is organized as the following. Section 2 introduces some temporal operators and functions in order for us to represent the temporal information of a multimedia resource within a presentation. Section 3 presents our specification language, as well as a number of inference rules to generate presentation implementations from specifications. And, section 4 highlights our contributions and points out some difficult problems as our further research direction.

2. Temporal Operators and Functions

In order to represent the temporal information of multimedia presentation resources, a representation of time model is necessary. The time model of our presentations is discrete. That is, a presentation consists of a number of continuous time intervals (or cycles). Next, we need a representation to embed presentation resources within these time intervals. Thus, a number of temporal operators and functions are defined in our inference system for the representation of resource temporal information. The concatenation operator, "^", is to connect two sequential resource streams. The silent operator, "_", while applying to a number, denotes a silent stream of many cycles. For instance, "_10" is a stream of no action which lasts 10 cycles. Note that a silent stream can be concatenated with another stream using the concatenation operator. The extension operator, "~", extends a resource stream, according to the *synchronization extent* of that resource (e.g., repeat, keep the last frame, no extension, etc.). Synchronization extent of multimedia resources describes how the resources should be presented after its regular ending. We discuss this concept in section 3. The truncate operator, associated with a number, can be used in two ways. "`r 10!`" says that resource `r` is played for the first 10 cycles only. And "`!10 r`" denotes `r` is played after cutting the first 10 cycles. These two operators can be applied together to a resource if the total cutting time is smaller than or equal to the duration of that resource. Otherwise, the presentation of that resource is omitted. The concurrent function, "$", is an overloaded function accepts one or more parameters. All resources with their names specified as parameters of the concurrent function start concurrently. The sequential function, denoted by "-", is also an overloaded function. The resources specified as parameters of a sequential function are presented one by one. There is no semantic difference between the sequential function 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. The last function is the identical function "#". This function is similar to the concurrent function with a further restriction indicates that all resources end at the same time as well.

3. Presentation Specifications

In this section, we propose a number of presentation specification statements as well as some inference rules for the automatic generation of multimedia presentations. The specification statements are used in our system as internal representations. They are quite difficult to be used directly by the user.

3.1. Resource Specification

The kinds of properties we consider here are essential for presentation generations. For instance, properties related to time and space are included. Other properties, such as key words, are not included while the user is issuing the resource specification. Only those resources used in a presentation are specified in the resource specification by the resource browser. And only those attributes related to the automatic generation of presentations will be included in the resource specification statements. Each resource is given a unique name, which maps to a resource descriptor (e.g., a file name, or a database entry). Temporal endurance, indicated by an integer as the number of cycles, specifies how long does a resource last in the presentation. the reserved word ∞ represents a permanent temporal endurance. For example, a picture can last as long as possible until it is dropped from its presentation window. Synchronization extent specifies how a resource is extended if requested. Some resources may not be extended. And some resources could end with a fade out effect. Detectability indicates how sensitive a resource attracts the user. Resources occupy screen space will be given their resolutions. Otherwise, a resolution of 0*0 is given (e.g., for sound, or music). Note that, we introduce a silent resource as a time slot holder which takes no action in a presentation.

A presentation consists of multiple streams carry multiple resources. Some streams, due to the limitation of hardware, may not be played concurrently. In some occasions, two resources are not appropriate to be played at the same time, since it is hard for a person, for instance, to watch two video simultaneously. Or, according to the every day presentation experiences, some streams are encourged to be played concurrently. For instance, it is nice to have a MIDI music background for an animation play. Our approach is to reduce the load of the user by given these good suggestions. However, if the user strongly against our suggestion, it is still possible for he/she to change the final presentation generated.

Resource Specifications are given in a Prolog predicate format containing seven parameters. Note that, some restrictions may be applied to resource specifications. For instance, there is no screen resolution for sound and MIDI music[1]. Moreover, the temporal endurance for pictures and text files, theoretically, should be infinite. However, we also allow the user to specify a finite temporal endurance for a picture or text. The notion of screen resolution is important. Not only it decides the screen layouts, the resolution information also works with the temporal information in that any screen region can not be occupied by two or more resources at the same time for our current system.

3.2. Temporal Specification

The research discussed in [1] proposes thirteen types of relations between two temporal intervals. The thirteen relations cover all possible situations. However, multimedia presentation synchronization needs precise timing information of resources. Some modifications to the relations are necessary in order for us to achieve multimedia resource synchronization. Some statements are given additional arguments to explicitly indicate a synchronization point. The following table shows our revised relations:

Revised Interval Temporal Relations

- always(r1, n): Always present r1 for n cycles.
- meets(r1, r2): r2 is presented right 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 overlaps r2, r1 starts first, r2 starts n cycles after r1 starts, r1 ends before r2.
- embraces(r1, r2, n): r1 embraces r2, r1 starts first, r2 starts n cycles after r1 starts, r2 ends before r1.
- equal(r1, r2): r1 and r2 carry the same duration concurrently.
- simultaneous(r1, n1, r2, n2): The n1-th cycle of r1 and the n2-th cycle of r2 happen at the same time.

Our purpose is to reduce the load of the user by means of automation. We try to make specification statements as general as possible while still maintaining all possible temporal relations between two multimedia resources. Considering our revised relations, it is possible to combine some of them by adding extra parameters. For example, by introducing a delay parameter, we can combine the "meets" and the "before" relations. If the delay parameter in the "sequential" specification is zero, the specification indicates a "meets" relation. Otherwise, it indicates a "before" relation. Note that, all timing parameters in our specifications are non-negative. The following are specification statements used in our system:

Temporal Specification Statements

- always(r1, n): The system always presents r1 for n cycles. This statement applies to picture or text resources only.
- 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 n1-th cycle of r1 and the n2-th cycle of r2 are synchronized.

In order to generate temporal implementation from temporal specification, we have a number of inference rules defined in our system to carry out the derivation. Two types of inference rules are designed, specification inference rules and implementation inference rules, to generate intermedia representations from the specification and to derive the final implementation from the intermediate representations. We use a short hand to extract the temporal endurance (i.e., **TE**) from a resource (designated by `r`, `r1`, `r2`, etc.). And, `r1 >> r2` is the priority checking expression compares the presentation priorities of resources `r1` and `r2`. The relation

[1]Even sound can be recorded in different sampling frequency and 8-bit or 16-bit option, we are not considering the difference in the generated presentation, as long as the underlying hardware works properly.

between two resources (i.e., mutual exclusive, possible concurrent, or mutual inclusive) is represented by an expression like "r1 op r2" where "op" can be an "X", a "?", or an "O" for the three type of relations, respectively. In the inference rules, we use the temporal operators and functions discussed in section 2. After presentation specifications are translated to their intermediate form, the system takes a further step to simplify the intermediate representations.

3.3. Spatial Specification

Most of the authoring tools allow users to specify spatial information of presentation objects directly on the screen via interactive drag and drop tools. Our current system also takes the same approach. However, it is possible to take a further step in how to specify the spatial specification of a presentation. For instance, a presentation system may allow the user to specify the spatial relation between two presentation objects, or among three, four, or more objects. This approach has the advantage similar to using our temporal specification statements. That is, when the user changes part of the layout, only one or two spatial specification statements are changed instead of re-positioning most of the objects on the screen. However, it is very difficult to generate presentations automatically based on these spatial relations. In this section, we propose a possible solution toward the automation. Similar to temporal specifications, spatial specifications are given in a predicate format.

4. Conclusions

We propose a mechanism and a system for the automatic generation of interactive multimedia presentations. The mechanism uses interval temporal logic inference rules as a tool to achieve multimedia resource synchronization. These rules incorporate issues such as hardware limitations, properties of multimedia resources, and good presentation principles. The specification statements we proposed covers all temporal relations given in [1]. To precisely specify synchronization points, we annotate the relations with timing parameters. A ICON programming user interface and the presentation generator are developed. We use our prototype system to generate some simple presentations, such as a city tour and an introduction lecture to multimedia PCs.

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 [1]. Secondly, a number of inference rules are developed and a presentation generator is implemented. Thirdly, a ICON programming interface is designed for the convenience of the user. Using our system, the presentation designer is able to specify *what* he/she wants instead of telling the computer exactly *how* the presentation show be scheduled.

References

[1] James F. Allen "Maintaining Knowledge about Temporal Intervals," Communications 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 proceedings of the ICMCS'95 conference, pp 278–281, 1995.

[3] S. K. Feiner and K. R. McKeown, "Automating the Generation of Coordinated Multimedia Explanations," In Intelligent Multimedia Interfaces, edited by Mark T. Maybury, American Association for Artificial Intelligence, pp 117–138, 1993.

[4] Thomas D. C. Little and Arif Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, April, pp 413–427, 1990.

[5] 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.

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

[7] 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.

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

[9] Timothy K. Shih, Chin-Hwa Kuo, and Kuan-Shen An, "Multimedia Presentation Designs with Database Support," in Proceedings of the NCS'95 conference, Taiwan, 1995.

[10] Thomas Wahl, et. al., "TIEMPO: Temporal Modeling and Authoring of Interactive Multimedia" in proceedings of the international conference on multimedia computing and systems, Washington DC, U.S.A., May 15–18, 1995, pp 274–277.