TEMPORAL AND SPATIAL SEMANTIC MODELS FOR MULTIMEDIA PRESENTATIONS

*Shu-Ching Chen and R. L. Kashyap**

School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285, U.S.A.
Email: shuching[kashyap]@ecn.purdue.edu

ABSTRACT

An abstract semantic model based on augmented transition network (ATN) to model multimedia presentations is presented in this paper. The inputs for ATNs are modeled by regular expressions. Regular expressions provide an efficient means for iconic indexing of the temporal/spatial relations of media streams and semantic objects. An ATN and its subnetworks are used to represent the appearing sequence of media streams and semantic objects. The arc label is a substring of a regular expression. In this design, a presentation is driven by a regular expression. User interactions, loops, and embedded presentations in multimedia presentations are also provided in ATNs.

1 Introduction

Many semantic models have been proposed to model the temporal and spatial relations. Some of the semantic models developed in the recent past are graphical structures to model the multimedia presentations. Semantic models such as OCPN (Little and Ghafoor, 1990), MDS (Chang et al.,1995), Firefly (Buchanan and Zellweger,1993), and Hirzalla et al's graphical temporal model (Hirzalla et al.,1995) all fall into this category. In other than providing a browsing facility for a multimedia presentation sequence, these models do not allow any specific temporal and spatial multimedia database queries. In other multimedia database systems (Oomoto and Tanaka, 1993; Özsu et al., 1995; Flickner, 1995), the emphasis is to present different media streams to users with query specifications. These models provide the searching capabilities to allow users to retrieve information from the database. However, these models do not consider the synchronization and quality of service issues after the requested media streams are obtained. Schloss and Wynblatt (1995) proposed a layered multimedia data model (LMDM). Their model has the capabilities to share and reuse both data objects and structures, but it has no query capability. Augmented transition network (ATN), developed by Woods (1970), has been used in natural language understanding systems and question answering systems for both text and speech. We use ATN as a semantic model to model a multimedia presentation and the temporal, spatial, or spatio-temporal relations of various media streams and semantic objects. A regular expression (Kleene, 1956) consists of one or more media streams and is used as an input for an ATN.

*Partially supported by National Science Foundation under contract 9619812-IRI and the office of Naval Research under contract N00014-91-J-4126.

The organization of this paper is as follows. Section 2 discusses how to use an ATN to model a multimedia presentation and to incorporate with multimedia database searching. The input for an ATN which is modeled by regular expressions is illustrated in section 3. Section 4 shows an example which use ATNs and regular expressions to model multimedia presentations. Conclusions are in section 5.

2 The Augmented Transition Network (ATN)

A multimedia environment should not only display media streams to users but also allow two-way communication between users and the multimedia system. The multimedia environment consists of a multimedia presentation system and a multimedia database system. If a multimedia environment has only a presentation system but without a multimedia database system then it is like a VCR or a TV without feedback from user. A multimedia database system allows users to specify queries for information. The information may be relative to text data as well as image or video content. By combining multimedia presentation and multimedia database system, users can specify queries which reflect what they want to see or know. A semantic model that models the presentation has the ability to check the features specified by users in the queries, and maintains the synchronization and QoS desired.

A finite state machine (FSM) consists of a network of nodes and directed arcs connecting them. The FSM is a simple transition network. Every language that can be described by an FSM can be described by a regular grammar, and vice versa. The nodes correspond to states and the arcs represent the transitions from state to state. Each arc is labeled with a symbol whose input can cause a transition from the state at the tail of the arc to the state at its head. This feature makes FSM have the ability to model a presentation from the initial state to some final states or to let users watch the presentation fast forward or reverse. However, users may want to watch part of a presentation by specifying some features relative to image or video contents prior to a multimedia presentation, and a designer may want to include other presentations in a presentation. These two features require a pushdown mechanism that permits one to suspend the current processing and go to another state to analyze a query that involves temporal, spatial, or spatio-temporal relationships. Since FSM does not have the mechanism to build up the hierarchical structure, it cannot satisfy these two features.

This weakness can be eliminated by adding a recursive control mechanism to the FSM to form a *recursive transition network* (RTN). A recursive transition network is similar to an FSM with the modifications as follows: all states are given names which are then allowed as part of labels on arcs in addition to the normal input symbols. Based on these labels, subnetworks may be created. Three situations can generate subnetworks. In the first situation, when an input symbol contains an image or a video frame, a subnetwork is generated. A new state is created for the subnetwork if there is any change of the number of semantic objects or any change of the relative position. Therefore, the temporal, spatial, or spatio-temporal relations of the semantic objects are modeled in this subnetwork. In other words, users can choose the scenarios relative to the temporal, spatial, or spatio-temporal relations of the video or image contents that they want to watch via queries. Second, if an input symbol contains a text media stream, the keywords in the text media stream become the input symbols of a subnetwork. A keyword can be a word or a sentence. A new state of the subnetwork is created for each keyword. Keywords are the labels on the arcs. The input symbols of the subnetwork have the same order as the keywords appear in the text. Users can specify the criteria based on a keyword or a combination of keywords in the queries. In addition, the information of other databases can be accessed by keywords via the text subnetworks. For example, if a text subnetwork contains the keyword "Purdue University Library" then the Purdue University library database is linked via a query with this keyword. In this design, an ATN can connect multiple existing database systems by passing the control to them. After exiting the linked database system, the control is back to the ATN. Third, if an ATN wants to include another existing presentation (ATN) as a subnetwork, the initial state name of the existing presentation (ATN) is put as the arc label of the ATN. This allows any existing presentations to be embedded in the current ATN to make a new design easier. The advantage is that the other presentation structure is independent of the current presentation structure. This makes both the designer and users have a clear view of the presentation. Any change in the shared presentation is done in the shared presentation itself. There is no need to modify those presentations which use it as a subnetwork.

Before the control is passed to the subnetwork, the state name at the head of the arc is pushed into the push-down store. The analysis then goes to the subnetwork whose initial state name is part of the arc label. When a final state of the subnetwork is reached, a pop occurs and the control goes back to the state removed from the top of the push-down store. Examples to illustrate the process will be demonstrated in Section 4.

However, the FSM with recursion cannot describe cross-serial dependencies. For example, network delays may cause some media streams not to be displayed to users at the tentative start time and the preparation time for users to make decisions is unknown when user interactions are provided. In both situations, there is a period of delay which should be propagated to the later presentations. Also, users may specify queries related to semantic objects across several subnetworks. The information in each subnetwork should be kept so that the analysis across multiple subnetworks can be done. For example, the temporal, spatial, or spatio-temporal relations among semantic objects may involve several video subnetworks. The cross-serial dependencies can be obtained by specifying conditions and actions in each arc. The arrangement of states and arcs represents the surface structure of a multimedia presentation sequence. If a user wants to specify a presentation which may be quite different from the surface structure then the actions permit rearrangements and embeddings, and control the synchronization and quality of service of the original presentation sequence. The cross-serial dependencies are achieved by using *variables* and they can be used in later actions or subsequent input symbols to refer to their values. The actions determine additions, subtractions, and changes to the values of *variables* in terms of the current input symbol and conditions. Conditions provide more sensitive controls on the transitions in ATNs. A condition is a combination of checkings involving the feature elements of media streams such as the start time, end time, etc. An action cannot be taken if its condition turns out to be false. Thus more elaborate restrictions can be imposed on the current input symbol for synchronization and quality of service controls. Also, information can be passed along in an ATN to determine future transitions. The recursive FSM with these additions forms an *augmented recursive transition network* (ATN).

3 Formulation of Input Symbols Using Regular Expressions

Originally, an ATN is used for the analysis of natural language sentences. Its input is a sentence composed of words. This input format is not suitable to represent a multimedia presentation since several media streams need to be displayed at the same time, to be overlapped, to be seen repeatedly, etc.

Regular expressions are useful descriptors of patterns such as tokens used in a programming language. Regular expressions provide convenient ways of specifying a certain set of strings. In this study, these strings are used to represent the presentation sequences of the temporal media streams, spatio-temporal relations of semantic objects, and keyword compositions. Information can be obtained with low time complexity by analyzing these strings. Regular expression goes from the left to right which can represent the time sequence of a multimedia presentation but it cannot represent concurrent appearance and spatial location of media streams and semantic objects. In order to let regular expressions have these two abilities, several modifications are needed. There are two levels need to be represented by regular expressions. At the coarse-grained level, the main presentation which involves media streams is modeled. At the fine-grained level, the semantic objects in image or video frames and the keywords in a text media stream are modeled at subnetworks. Each keyword in a text media stream is the arc label at subnetworks. New states and arcs are created to model each keyword. The details to model coarse-grained level are discussed in the follows.

Two notations $\mathcal{L}$ and $\mathcal{D}$ are used to define regular expressions and are defined as follows:

$\mathcal{L}$ = {A, I, T, V} is the set whose members represent the media type, where A, I, T, V denote audio, image, text, and video, respectively.
$\mathcal{D}$ = {0, 1, ..., 9} is the set consisting of the set of the ten decimal digits.

Definition 1: Each input symbol of a regular expression contains one or more media streams which are enclosed by a parentheses and are displayed at the same time interval. A media stream is a string which begins with a letter in $\mathcal{L}$ subscripted by a string of digits in $\mathcal{D}$. For example, V_1 represents video media stream and its identification number is one. The same video or audio media stream may appear in more than one consecutive input symbol and a superscripted string of digits is used to distinguish them such as V_1^1, V_1^2, and so on. The following situations can be modeled by a regular expression.

- **Concurrent:** The symbol "&" between two media streams indicates these two media streams are displayed concurrently. For example, $(T_1 \& V_1)$ represents T_1 and V_1 to be displayed concurrently.

- **Looping:** $m^+ = \bigcup_{i=1}^{\infty} m^i$ is the regular expression of positive closure of m to denote m occurring one or more times. We use the "+" symbol to model loops in a multimedia presentation to let some part of the presentation be displayed more than once.

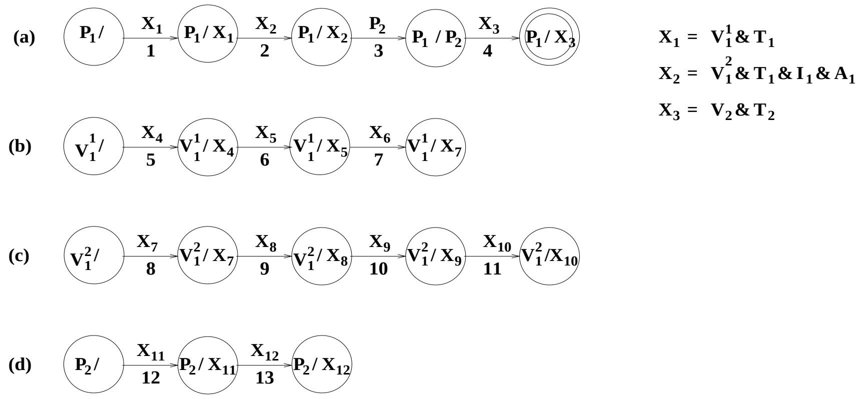
- **Optional:** $m^* = \bigcup_{i=0}^{\infty} m^i$ is the regular expression of Kleene closure of m to denote m occurring zero or more times. In a multimedia presentation, when the network becomes congested the original specified media streams which are stored in the remote server might not be able to arrive on time. The designer can use "*" symbol to indicate the media streams which can be dropped in the on-line presentation. For example, $(T_1 \& V_1^*)$ means T_1 and V_1 will be displayed but V_1 can be dropped if some criteria cannot be met.

- **Contiguous:** Input symbols which are concatenated together are used to represent a multimedia presentation sequence and to form a regular expression. Input symbols are displayed from left to right across time sequentially. ab is the regular expression of a concatenated with b such that b will be displayed after a is displayed. For example, $(A_1 \& T_1)(A_2 \& T_2)$ consists of two input symbols $(A_1 \& T_1)$ and $(A_2 \& T_2)$. These two input symbols are concatenated together to show that the first input $(A_1 \& T_1)$ symbol is displayed before the second input symbol $(A_2 \& T_2)$.

- **Alternative:** A regular expression can model user selections by separating input symbols with the "|" symbol. So, $(a|b)$ is the regular expression of a *or* b. For example, $((A_1 \& T_1) \mid (A_2 \& T_2))$ denotes either the input symbol $(A_1 \& T_1)$ or the input symbol $(A_2 \& T_2)$ to be displayed.

4 An example

Figure 1 shows an example to use an ATN to model a multimedia presentation. In Figure 1(a), after X_1 and X_2 are displayed, an existing presentation (say P_2) is displayed. Since P_2 is an existing presentation, it becomes a subnetwork of P_1 (as shown in Figure 1(d)). Since P_2 is embedded in P_1, X_3 is displayed after P_2 and the final state P_1/X_3 is reached.



(e)

Arc	Symbol	Condition	Action
1	X_1	Bandwidth < Θ	Get CV_1^1
		Bandwidth>= Θ	Get V_1^1
		Current_time - Tentative_start_time(X_1) < Duration	Display
		Current_time - Tentative_start_time(X_1) >= Duration	Next_symbol(X_2) and Next_State

Figure 1: Augumented Transition Network: (a) is the ATN network for a multimedia presentation which starts at the state names $P_1/$. (b)-(d) are part of the subnetworks of (a). (b) and (c) model the semantic objects in video media stream V_1, and (d) is an embedded presentation. The conditions and actions for input symbol X_1 are shown in (e). CV_1^1 stands for the compressed version of the video media stream V_1^1. The "Get" procedure is to access an individual media stream. "Display" procedure is to display the media streams. "Next_Symbol(X_i)" reads the input symbol X_i. "Next_State" is a procedure to advance to the next state. θ is a parameter.

In this presentation, the regular expression is:

$$\underbrace{(V_1^1 \& T_1)}_{X_1}\underbrace{(V_1^2 \& T_1 \& I_1 \& A_1)}_{X_2}\underbrace{(P_2)}_{P_2}\underbrace{(V_2 \& T_2)}_{X_3}$$

The input symbol X_1 contains V_1^1 (video stream 1) and T_1 (text 1) which start at the same time and play concurrently. Later, I_1 (Image 1) and A_1 (Audio 1) begin and overlap with V_1^2 and T_1. Therefore, the input symbol X_2 contains the media streams V_1^2, T_1, I_1, and A_1. Each media stream has its own regular expression and is a subnetwork of P_1 (as shown in Figures 1(b) and 1(c) for video media stream V_1). V_1^1 and V_1^2 are the first and the second parts of V_1, respectively. The delay time for I_1 and A_1 to display needs not to be specified in regular expression explicitly since the regular expression is read from left to right so that the time needed to process X_1 is the same as the delay time for I_1 and A_1.

Figure 1(e) shows the conditions and actions for input symbol X_1. In this presentation, when the current input symbol X_1 ($V_1^1 \& T_1$) is read, the bandwidth condition is first checked to see whether the bandwidth is enough to transmit these two media streams. If it is not enough then the compressed version of V_1^1 (CV_1^1) will be transmitted instead V_1^1. Then the condition whether the pre-specified duration to display V_1^1 and T_1 is reached is checked. If it is not, the display continues. The tentative start time is defined to be the time when the displaying of the media streams starts. The difference between the current time and the tentative start time is the total display time so far. The last condition is met when the total display time reaches the pre-specified duration. In this case, a next input symbol X_2 is read. The same conditions are checked for X_2, too. The process continues until the final state is reached.

5 Conclusions

In this paper, we describe an ATN based model together with regular expressions for multimedia presentations. Unlike the existing semantic models which only model user interactions, loops, or embedded presentations, our ATN model provides these three capabilities in one framework. User interaction feature allows two-way communication between users and multimedia information systems. Loops can be used to let some part of a presentation be watched more than once. Embedded presentations emphasize the modularity and reuse of existing media streams and presentation structures. Under this design, the storage intensive multimedia data can be stored into large shared databases. This feature greatly reduces the design complexity and makes the design easier. Additional details of our method is in the (Chen and Kashyap, 1998).

References

[1] M. Buchanan and P. Zellweger, "Automatically Generating Consistent Schedules for Multimedia Documents," ACM Multimedia Systems Journal, 1(2), Springer-Verlag, 1993.

[2] S.C. Chen and R.L. Kashyap, "A Spatio-Temporal Semantic Model for Multimedia Presentations and Multimedia Database Systems," will submit to IEEE Trans. on Knowledge and Data Eng, special section on Data and Knowledge Management in Multimedia Systems.

[3] H.J. Chang, T.Y. Hou, S.K. Chang, "The Management and Application of Teleaction Objects," ACM Multimedia Systems Journal (1995) Volume 3, November 1995, pp 228-237.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, "Query by Image and Video Content: The QBIC System," IEEE Computer, Vol. 28, No. 9, pp. 23-31, September 1995.

[5] N. Hirzalla, Ben Falchuk, and Ahmed Karmouch, "A Temporal Model for Interactive Multimedia Scenarios," IEEE Multimedia, Fall 1995, pp. 24-31.

[6] S.C. Kleene, "Representation of Events in Nerve Nets and Finite Automata, Automata Studies," Princeton University Press, Princeton, N.J., 1956, pp. 3-41.

[7] T.D.C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE J. Selected Areas in Commun., Vol. 9, pp. 413-427, Apr. 1990.

[8] E. Oomoto, and K. Tanaka, "OVID: Design and Implementation of a Video Object Database System," IEEE Trans. on Knowledge and Data Engineering, Vol. 5, No. 4, pp. 629-643, August 1993.

[9] M.T. Özsu, D. Duane, G. El-Medani, C. Vittal, "An object-oriented multimedia database system for a news-on-demand application," ACM Multimedia Systems Journal (1995) Volume 3, November 1995, pp 182-203.

[10] G.A. Schloss and M.J. Wynblatt, "Providing definition and temporal structure for multimedia data," ACM Multimedia Systems Journal (1995) Volume 3, November 1995, pp 264-277.

[11] W. Woods, "Transition Network Grammars for Natural Language Analysis," Comm. ACM, **13**, October 1970, pp. 591-602.