A Multimedia Semantic Model for RTSP-Based Multimedia Presentation Systems

Shu-Ching Chen[1]*, Sheng-Tun Li[2], Mei-Ling Shyu[3], Chengjun Zhan[1], Chengcui Zhang[1]
[1]Distributed Multimedia Information System Laboratory
School of Computer Science
Florida International University
Miami, FL 33199, USA

[2]Department of Information Management
National Kaohsiung First University of Science and Technology
Juoyue Rd. Nantz District, Kaohsiung 811, Taiwan, R.O.C.

[3]Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124, USA

Abstract

In this paper, an abstract multimedia semantic model called Multimedia Augmented Transition Network (MATN) model is proposed to model the RTSP-based (Real-Time Streaming Protocol) multimedia presentation system. RTSP provides a set of methods to support the functionality of "VCR-like" remote control on the client side so that the users are allowed to control the playback of the streaming media. The MATN model is used to model the RTSP actions as well as the temporal relations and synchronization control of the media streams. The advantages of using the MATN model are its simplicity, effectiveness, and ease of modification. Furthermore, the formal definition of the MATN model and the modeling of the RTSP actions using the MATN model are presented in details.

Keywords: Multimedia Augmented Transition Network (MATN), Multimedia Semantic Model, Real-Time Streaming Protocol (RTSP).

1. Introduction

The cooperation of the World Wide Web (WWW) and multimedia technologies results in Web-based multimedia presentations so that one can enjoy watching TV-like programs via general browsers. However, a multimedia presentation system should not only display media streams to the users but also allow two-way communications between the users and the multimedia system. A high-level multimedia semantic model can help to systematically model the temporal relations and user interactions in a multimedia presentation. In order to choose a semantic model that is especially suitable for a multimedia presentation system, the abstract semantic model called the Multimedia Augmented Transition Network (MATN) model [4] is considered effective for this purpose.

As a promising graphical model, the MATN model has been applied successfully in many application domains [6] [7]. It can describe and study the multimedia systems that are characterized as being parallel, concurrent, and/or alternative. The advantages of using an MATN are its simplicity, effectiveness, and ease of modification. It is simple in that it uses state-transition graph and regular-expression like grammars to represent the temporal and synchronization control information. It is powerful in modeling the semantics of a multimedia presentation system in that it permits recursions and uses the condition/action table to enable the QoS (quality of service) control. It enables scalability in that an MATN can model user interactions in a single framework, unlike the timeline models [3] that need to have several timelines to model the same situation. Moreover, it has a hierarchical structure composed of subnetworks, which makes an MATN being ease of modification.

*This research was supported in part by NSF (CDA-9711582, EIA-0220562) and Telecommunications & Information Technology Institute (IT2/FIU) under IT2 BA01.

In this paper, we present the semantic modeling process using the MATN model for an RTSP-Based Multimedia Presentation System developed in [12], which is a platform-neutral system and incorporates the VCR-like functionality to allow more interactions between the users and the multimedia servers. This system is built on top of Java Media Framework (JMF) [2] to achieve the platform-neutrality and to realize the Real-Time Streaming Protocol (RTSP). How the RTSP actions as well as the temporal relations and synchronization control of media streams are modeled using the MATN model will also be presented.

This paper is organized as follows. In Section 2, the formal definition of the MATN model is presented. Section 3 gives a brief introduction of an RTSP-based multimedia presentation system and discusses in details the use of the MATN in modeling the RTSP actions as well as the temporal relations and synchronizations of the media streams. Section 4 concludes the paper.

2. The Multimedia Augmented Transition Network (MATN) Model

A Multimedia Augmented Transition Network (MATN) model is similar to a finite state automaton. It originates from the Augmented Transition Network (ATN) [15]. The input of an MATN is the multimedia input string, which will be introduced later. Similar to finite automata, an MATN contains nodes and arcs to represent the states and transitions from one state to another. However, unlike finite automata, an MATN allows recursions and has the condition/action tables. These features of an MATN make it more powerful and effective than finite automata. The capability of modeling recursions allows users to play some part of a presentation more than once. With the condition/action tables, an MATN can control the synchronization and QoS of multimedia streams. Another important feature of an MATN is its support for subnetworks, where each subnetwork is another MATN that is part of the current MATN.

```
1.   Multimedia_Input_String = ( Arc_Name, [‘+’ | ‘$’] )
     { ‘|’, (Arc_Name, [‘+’ | ‘$’ ]) };
2.   Arc_Name = Control_Command | Media_Streams;
3.   Media_Streams = ‘(’ Presentation_Name ‘)’ |
     ‘(’ (Single_Media, [‘*’] ) {‘&’, (Single_Media, [‘*’])}‘)’;
4.   Presentation_Name = ‘P’ { ‘a’-‘z’‘A’-‘Z’‘0’-‘9’};
5.   Control_Command = ‘_’ (‘a’-‘z’‘A’-‘Z’) {‘0’-‘9’}-;
6.   Single_Media = (‘A’ | ‘I’ | ‘T’ | ‘V’ | ‘B’) {‘0’-‘9’}-;
7.   Node_Name = Presentation_Name ‘/’ (‘a’-‘z’‘A’-‘Z’) {‘0’-‘9’}-;
```

Table 1. The EBNF production rules for the multimedia input string and the name of the state.

Symbol	Meaning
Unquoted words	Non-terminal symbol
‘...’	Terminal symbol
=	Is defined as
{...}	0 or more repetitions
{...}-	One or more repetitions
[...]	Optional symbols
...|...	Or
;	Rule terminator
,	Concatenation
[c_1 - c_2]	Any character between character c_1 and character c_2 in ASCII order

Table 2. The meaning of the symbols in the EBNF notation.

2.1. Inputs of an MATN

We use the notations similar to the extended version of the Backus Naur Form (EBNF) [8] [14] to define the valid inputs and notations for an MATN. Regular expressions [10] are used to define the symbols of an MATN. The formal definition of the multimedia input string as well as the name of the state is described in Table 1. Also, the meanings of the special symbols in the EBNF notation are described in Table 2.

There are two kinds of basic inputs for an MATN. One is “Control_Command” (Rule 5 in Table 1), which is defined as a string with a leading underscore symbol ”_” followed by a string that is composed of a character followed by numbers. It is used to represent a control message that may occur during the multimedia presentation. ”Media_Streams” (Rule 3 in Table 1), the other basic input of the MATN model, is defined as either ”Presentation_Name” or a combination of ”Single_Media.” There are four basic media stream types that are A (Audio), I (Image), T (Text) and V (Video) in ”Single_Media.” In addition, ”Single_Media” can be extended to include other displayable objects such as selection button (B). The definition of ”Single_Media” (Rule 6) uses one representative character (A/I/T/V/B) and a number to identify different media streams, e.g., A1, I2, and T4. The ”Media_Streams” distinguishes from ”Control_Command” in that ”Control_Command” starts with an underscore ”_” while a ”Media_Streams” starts with one of the 5 representative characters (A/I/T/V/B). The ”Presentation_Name” (Rule 4 in Table 1) is defined as a character string with leading character ”P”. It can represent a whole presentation or a subnetwork of the MATN, where a subnetwork is a presentation that may be part of another presentation. For example, P1 may be a presentation name; whereas P2 may be a subnetwork name.

An MATN is composed of nodes and arcs. We define the ”Node_Name” (Rule 7 in Table 1) as a combination of ”Presentation_Name” and a string that represents the arc string that leads to this node, e.g., P1/X1. The ”Arc_Name” (Rule 2 in Table 1) is defined as either ”Control_Command” or ”Media_Streams.” If an arc is labeled by a control command, it means the control jumps from one state to the next state after executing this command. When an arc is labeled by the media streams, the arc name is either the combination of the single media symbols which are connected by the symbol ”&” to denote concurrent displaying or a presentation symbol. For example, the arc name V1&T1 means the two media streams are displayed concurrently. The media streams may be labeled by the special symbol ”*,” which means optional. For example, (T1 & V1*) means T1 and V1 will be displayed concurrently but V1 could be dropped if some criteria (defined in the Condition/Action table of the MATN) cannot be met.

A multimedia input string is represented by a combination of the arc names and some special symbols (such as ”&”, ”*”, ”+”, ”|” and ”$”) that represent special meanings. The meanings of the special symbols associated with the multimedia input strings are listed in Table 3. As mentioned earlier, the special symbols ”&” and ”*” denote concurrent displaying and optional. The ”+” symbol means looping. For example, T1$^+$ means that media stream T1 can be displayed more than once. The ”|” symbol means alternatives. For example, (A1 & T1) | (A2 & T2) denotes either the input string (A1 & T1) or the input string (A2 & T2) will be displayed. Finally, the symbol ”$” means the end of a presentation.

Special Symbol	Meaning
&	Concurrent
*	Optional (0 or 1)
+	Loop
|	Alternative
$	The end of a presentation

Table 3. The meaning of the special symbols in multimedia input strings.

2.2. Definition of the MATN Model

The MATN model can be formally defined in the following definition.

Definition 1: An MATN is a 8-tuple, (Σ, Γ, **Q**, Ψ, δ, **T**, **S**, **F**), where Σ, Γ, **Q** and **T** are all finite sets.

Σ: The media stream and control command alphabet, which is the basic input of the MATN. Its elements include the underscore character ”_”, any character between ”a”-”z”, ”A”-”Z” and digital numbers 0-9.

Γ: The special input symbol alphabet. $\Gamma = \{\&, +, *, —, (,), \$\}$.

Q: The set of nodes (states). Its grammar is defined in Table 1.

Ψ: The set of arc names. Its grammar is defined in Table 1.

δ: The transition function from one state (node) to another. In an MATN, two states are connected by an arc. δ: $\mathbf{Q}\times\Psi\longrightarrow\mathbf{Q}$.

S: The start state of an MATN, where $\mathbf{S}\in\mathbf{Q}$.

F: The set of final states, where $\mathbf{F}\subset\mathbf{Q}$.

T: The condition/action table of an MATN. It is used to define conditions with the associated actions. **T** could be empty. It includes three fields: input symbols (e.g., an arc name), conditions and actions.

The condition/action table is a very important component of an MATN and can be used to store more information beyond the nodes and arcs. With the condition/action table, an MATN can provide different levels of presentation quality and synchronization and thus support different levels of QoS. The condition/action table is also one of the major differences between the MATN and the finite state automata. An example of the condition/action table will be given later in this paper.

One of the major features of an MATN is its strong support for user selections. Since an MATN can model user interactions and loops, the designer can design a presentation with selections so that users can have the choices to browse or watch the same part of the presentation more than once [4]. We will illustrate how it supports user selections in details by using the following example.

For example, in a presentation P1, suppose the multimedia input string is: (V1 & T1) (V2 & T2 & I1 & A1) ((B1 & B2) (((B1) (V3 & A2) (T3 & A3) $) | ((B2) (V4 & I2) (V5 & A4))))$^+$, which is an example with both user selections and loops. The constructed MATN is shown in Figure 1. Steps 1 to 7 listed below explain how it works.

- Step 1. The initial state is P1, and the outgoing arc (arc 1) is labeled with (V1&T1). Media streams V1 and T1 are displayed concurrently.

- Step 2. After displaying V1&T1, the current state moves to P1/X1 with the outgoing arc (arc 2) labeled with (V2&T2&I1&A1). Media streams V2, T2, I1 and A1 are displayed concurrently.

- Step 3. When the state P1/X2 is met, the input symbol X3 with two selections B1 and B2 is displayed to the users so that they can make their selections. The current state becomes P1/X3.

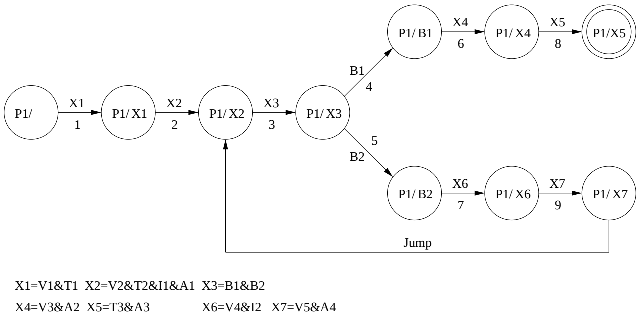

Figure 1. An MATN with user selections and loops.

- Steps 4-5. If B1 is selected, arc 4 is followed and media streams V3 and A2 are displayed. The state moving sequence is P1/X3⟶P1/B1⟶P1/X4. If B2 is selected, arc 5 is then followed and media streams V4 and I2 are displayed. The state P1/X6 will be reached.

- Step 6. Based on the user selection in Step 4, if B1 is selected, the current state will go to P1/X4, and the media streams on arc 8 (T3&A3) are displayed. Then P1/X5 becomes the current state that is a final state. Hence, the presentation stops. If the user selects B2 in Step 4, the state P1/X6 with the outgoing arc 9 (V5&A4) will be reached, and media streams V5 and A4 will be displayed. In that case, the current state moves to P1/X7, which is followed by a ”Jump” arc as shown in Figure 1. Go to Step 7.

- Step 7. Since the current state is P1/X7 that has the ”Jump” arc, the process will go back to Step 3 to let the user make the choice again. Step 3 through Step 7 model a loop scenario that is represented by a ”+” symbol in the multimedia input strings. The ”Jump” action does not advance the input symbol but lets the control go back to the pointing state, which means ”Jump” itself is not an input symbol in the multimedia nput strings. This feature is crucial for the designers who may want some part of the presentation to be seen over and over again.

3. Modeling RTSP Actions Using the MATN Model

3.1. RTSP Action Diagram

Real-time Streaming Protocol (RTSP) is an application-level approach for controlling a single stream or multiple real-time streams such as video or audio delivered over the Internet [13], which is initiated by Netscape, Real Networks, and Columbia University. Similar to the HTTP protocol, RTSP uses textual commands to control stream transmission. Although HTTP and RTSP protocols have similarities, they also have differences. HTTP is a stateless protocol, while RTSP is a state-oriented protocol such that RTSP servers and clients need to maintain the states of the connection sessions labeled by session identifiers. In RTSP, both servers and clients may issue requests. While in HTTP, only clients could issue requests. Also, RTSP supports the functionality of ”VCR-like” remote control so that the client may fast-forward, rewind, pause or stop media streams. A RTSP session has two channels, one is for the control message, and the other is for media transmission. In the control channel, control messages such as Request/response are transmitted between the server and the client, and the underlying protocol could be either TCP or UDP. Media streams will be transmitted using the media channel. The underlying transport protocol for an RTSP session to deliver media streams is very flexible. It could be TCP, UDP, multicast UDP, or RTP (Real-Time Transport Protocol) [13].

RTSP defines a set of methods to support the service requests at different stages in an RTSP session. Such methods includes DESCRIBE, SETUP, PLAY, PAUSE, TEARDOWN, REDIRECT, etc. The general process is described as follows. At the beginning, a client initiates a DESCRIBE method to obtain the information about a presentation which may contain one or more media streams. Such information is often put in a presentation description file adherent to some protocol such as the Session Description Protocol [9]. After receiving and analyzing the description, the client issues the SETUP commands to ask the server to reserve resources for a media stream, and sends the PLAY commands to ask the server to transmit the media stream. If a PAUSE command is issued, the transmission is suspended but the server resources are not released. If a PLAY command is issued again, the transmission will be resumed. When the server receives a TEARDOWN command, it will terminate the transmission of the media streams and free the associated resources. When a client issues a request and the server responds with a REDIRECT message, then the client is asked to connect to another server. The action diagram of RTSP [13] is shown in Figure 2.

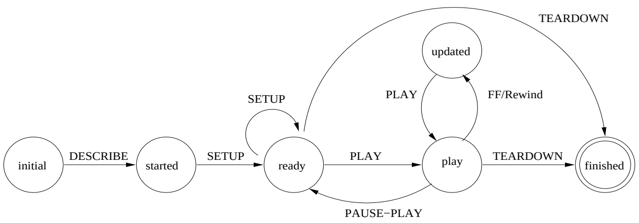

Figure 2. RTSP action diagram.

3.2. Case Study: An Example RTSP-based Multimedia Presentation System

The example RTSP-based multimedia presentation system used in this paper is developed in [12], which is based on the RTSP standard to provide the users a ”VCR-like” remote control to control the playback of streaming media over the Internet. The system can also deal with the Session Description Protocol (SDP)-oriented multimedia session descriptions. The actual implementation of the presentation system is based on Cisco’s IP/TV server [1].

3.2.1. The Example Scenario

In Section 3.1, the general process of RTSP is explained in details, and the RTSP action diagram is shown in Figure 2. In this subsection, we will first present an example of the general case, and then show how to model the RTSP actions using the MATN model. In the example scenario, a client issues the RTSP requests sequentially in the following order: DESCRIBE ⟶ SETUP ⟶ PLAY ⟶ FORWARD ⟶ BACKWARD ⟶ PAUSE ⟶ PLAY ⟶ TEARDOWN. Since this request sequence will trigger the execution of every RTSP command at least once, so it could be taken as a representative of the general cases. In this example, at the initial state, the client issues a DESCRIBE command trying to get information from the server:

```
DESCRIBE rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

Upon receiving this command, the server transmits the requested presentation information back to the client in a presentation description file. Table 4 is an example of a simplified SDP (Session Description Protocol [9]) file with brief comments in the right hand side in Italics. The client side receives the SDP description file and parses it, and then constructs the appropriate SETUP commands to ask the server to prepare the transmission of the desired media streams.

As shown in Table 4, the presentation contains two types of media streams: video and audio streams that are played synchronously. Thus the client will construct the SETUP commands as following:

1. SETUP rtsp://163.18.22.20:8554/ProgID=986149945/audio RTSP/1.0

   to ask for reserving resources for the audio stream.

2. SETUP rtsp://163.18.22.20:8554/ProgID=986149945/video RTSP/1.0

   to ask for reserving resources for the video stream.

```
.........
o=- 986149945 0 IN IP4 163.18.22.20
//session ID=986179945, server IP = 163.18.22.20
a=x-iptv-type:ondemand
//defined by Cisco IP/TV, meaning this session is on-demand
m=video 20730/1 RTP/AVP 32
//media name is video, using ports 20730/20731 for RTP/RTCP
c=IN IP4 163.18.22.20/1
//connection is through 163.18.22.20, TTL=1
a=control:rtsp://163.18.22.20:8554/ProgID=986149945/video
//RTSP URL, media attribute defined by Cisco IP/TV
a=range:npt=0-2836.0
//media’s normal play time is 2836 seconds
a=framerate:30.0
//the maximal video frame rate is 30 frames/sec
m=audio 20728/1 RTP/AVP 14
//media name is audio, using ports 20728/20729 for RTP/RTCP
c=IN IP4 163.18.22.20/1 //same as above
a=control:rtsp://163.18.22.20:8554/ProgID=986149945/audio
//same as above
a=range:npt=0-2836.0 //same as above
...
```

Table 4. An example description file in SDP.

After successfully issuing the SETUP commands and getting responses, the client could issue a PLAY command to ask the server to transmit the requested media streams to the client buffer. In this example, there are two media streams to be transmitted: one audio stream and one video stream. The RTSP request is:

```
PLAY rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

If the client buffer is full, the client will start to present the video and audio streams to the user using the media players. In the process of playing, the client could use a serial of commands to control the play of the media streams. Such commands are PAUSE, FORWARD, BACKWARD, STOP and TEARDOWN. The first four commands could be repeatedly issued, while the last one could only be issued once, because it will ask the server to release the reserved resources and the connection will be terminated. In our example, client will first issue a FORWARD request when playing. The appropriate RTSP requests are:

```
PAUSE rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
FORWARD rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

Note that the FORWARD command needs a preceding PAUSE command. Other commands like REWIND and TEARDOWN also need a preceding PAUSE command. Then the client issues a REWIND request. The RTSP requests are shown as follows.

```
PAUSE rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

```
REWIND rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

Now the client may want to pause the playing of the media streams for a period of time, so it issues the following RTSP request to the server:

```
PAUSE rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

At the end of the time period, the client chooses to continue the playing of the media streams again. Thus another PLAY command is issued to the server to request the continuation of the playing.

Later, after the playing is finished, or the client just wants to stop the playing of the media streams and to close the connection, it will issue a TEARDOWN command to the server to ask for the release of the resources and the close of the connection. The RTSP requests is:

```
PAUSE rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
TEARDOWN rtsp://163.18.22.20:8554/ProgID=986149945 RTSP/1.0
```

Thus we finished the detailed description of the RTSP requests for the example scenario that will be modeled using the MATN model. Next subsection will explain how to model it using the MATN model.

3.2.2. Modeling the Example Multimedia Presentation

The MATN model can be used to model the temporal relations and synchronization control in a multimedia presentation system. In this subsection, we will first show how to model the multimedia presentation discussed in Section 3.2.1. As shown in Table 4, there are two synchronous media streams, audio stream and video steam, which are denoted by A1 and V1 respectively. As described above, V1 and A1 are displayed simultaneously according to the information obtained by parsing the session description file.

Figure 3 shows how to use an MATN to model the temporal relations and synchronization control for this multimedia presentation. As shown in the figure, the state ”P1/” means the start state of the presentation, with the presentation name P1. The arc name (V1 & A1) means that the audio and video streams will be displayed concurrently on the client’s screen. After displaying V1 and A1, the control reaches the state P1/Y1, which is also the final state in this case. The system demonstration of such a multimedia presentation with the JMF players is shown in Figure 4. This example demonstrates the capability of the MATN model to model the temporal relations and synchronization control of a multimedia presentation. Although not shown in Figure 3, we could also design the presentation to let the control go back to any state node it has reached before, thus the users are allowed to view the same part of a presentation more than once.

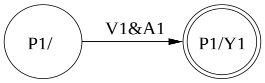

Figure 3. MATN model for the multimedia presentation described in Table 4.

Figure 4. An example system demonstration of the multimedia presentation.

3.3. Modeling the RTSP Actions

Based on the discussion of the RTSP action diagram and our case study, in this section we will show how to model the RTSP actions using the MATN model. The appropriate MATN model for the RTSP actions represented by the RTSP action diagram is shown in Figure 5. The details of MATN modeling will be revealed, and then the advantages of using the MATN to model the RTSP actions will be discussed. As explained earlier, there are two types of inputs for the arcs in an MATN: control commands and media streams. In Figure 5, all the arc inputs are control messages except P1, which is a subnetwork for the media streams (also shown in Figure 5). The corresponding condition/action table for subnetwork P1 is presented in Table 5. The following steps explain Figure 5 in details:

- Step 1. The first state P of the MATN corresponds to the initial state of the RTSP diagram. P is also the presentation name.

- Step 2. The current state shifts to P/X1 with the outgoing arc labeled with command ”_D1” that corresponds to ”DESCRIBE” in RTSP. With arc ”_D1,” there is an implicit action to obtain the description information of the multimedia presentation from the server.

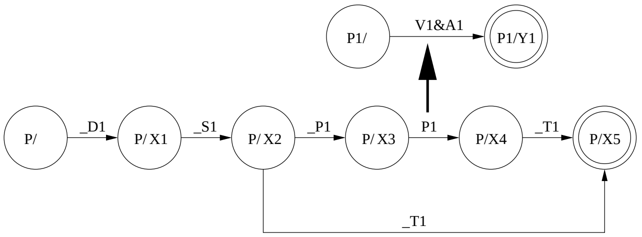

Figure 5. MATN model for the RTSP actions shown in Figure 2.

- Step 3. With the outgoing arc labeled ”_S1,” there is an implicit action by which a subnetwork P1 for the multimedia presentation is constructed and will be used later on. According to the example scenario described above, the construction of the subnetwork P1 is the same as the MATN model in Figure 3. Also, the current state of the MATN moves to P/X2, which corresponds to the ready state of the RTSP diagram, which means the system has already prepared the transmission for the media streams (represented in P1) and is ready to play.

- Step 4. In state P/X2, based on the selection of the client, the system can move to different states. If the client selects the arc labeled ”_T1” (corresponding to ”TEARDOWN” in RTSP), then the final state P/X5 is reached, and the MATN will stop. If the client selects the arc labeled ”_P1” (corresponding to ”PLAY” in RTSP), the MATN will reach state P/X3, and the multimedia presentation starts to play.

- Step 5. In state P/X3, the system will enter the subnetwork named P1 and the control is passed to P1, which will play the multimedia presentation. As shown in the condition/action table of P1 (in Table 5), P1 could respond with appropriate actions to the client activities such as FORWARD, REWIND and STOP, and it could also provide different levels of QoS according to the bandwidth and delay. We will explain the details later in this section.

- Step 6. Upon finishing this play session, the current system state moves to P/X4, which means that the client has already finished the multimedia presentation. With the outgoing arc labeled as ”_T1” (corresponding to ”TEARDOWN” in RTSP), the MATN reaches final state P/X5 and stops.

Input Symbols	Conditions	Actions
V1 & A1	If (bandwidth $\geq \alpha$)	Transmit V1 and A1
	If ((current_time - start_time(V1 & A1)) $\leq$ duration	Display V1 and A1
	If ((current_time - start_time(V1 & A1)) $\geq$ duration	Get_Symbol and Next_State
	If (receiving ”PAUSE” message)	Stop in the current position
	If (receiving ”FORWARD” message)	Advance the video and audio streams in fast speed
	If (receiving ”REWIND” message)	Rewind the video and audio streams
	If (receiving ”PLAY” message)	Recalculate the duration and delay. Create new states to represent the changes.

Table 5. A condition/action table for subnetwork P1.

Table 5 shows an example of how to use conditions and actions to maintain synchronization and QoS. As can be seen from Table 5, if the network bandwidth is large enough, then V1 and A1 are both transmitted. Also, in Table 5, we could see that the MATN will check the temporal relations when it reads an input symbol. If the time elapsed so far from the beginning of the presentation is less than a threshold value, the MATN will display V1 and A1 normally. However, if the time elapsed is equal to or larger than that threshold value, which means the delay is too long and we could not finish the presentation as planned (i.e., displaying V1 and A1 normally), then the MATN will just skip V1 and A1 and read the next input symbol. In Table 5, Get_Symbol is a procedure to read in the next input symbol of the multimedia input string, while Next_State is a procedure that skips the current state and go directly to the next state of the MATN.

Table 5 also shows the actions towards the messages received during the multimedia presentation. If the MATN gets a ”PAUSE” message, it will stop in the current position. If a ”FORWARD” or ”REWIND” message has been received, it will fast advance or rewind the video and audio streams with a predefined speed (faster than the normal playing speed). If a ”PLAY” message arrives, the MATN will recalculate presentation-related variables such as the delay and duration, and create new states and arcs to reflect the changes if applicable.

3.4. Advantages of Using MATN to Model RTSP Actions

In Section 3.1, an action diagram is used to describe RTSP workflow modeling, which is actually an example of Finite State Machine (FSM). According to [4], MATN has advantages over FSM in that it permits recursions. Moreover, the condition/action table in the MATN model can store information related to the media streams, while FSM could only store information within its nodes and arcs. With the condition/action tables, the MATN model can provide different levels of presentation quality and synchronization, and thus support different levels of QoS. However, the action diagram could only represent the basic workflows of RTSP without the capability to express the rich actions hidden in the background.

Unlike the traditional action diagram that serves only as a general abstract model for behavior modeling and cannot be used to see how multimedia objects collaborate to compose a presentation or how a multimedia object is displayed over its lifetime, MATN is capable of effectively modeling a variety of multimedia objects in terms of their structure, behavior, and function, and supporting multimedia presentations.

In addition, the formal definition of MATN provides rules of interpretation that allow the understanding of complex multimedia presentations as well as the actions in RTSP [11]. It also provides the basis for automated support of RTSP-based multimedia systems. Another major advantage brought by the MATN model is its support of subnetworks. With subnetworks, the presentation designers can use another existing presentation sequence in the current presentation design. Also, any change in a subnetwork will automatically change the presentations that include this subnetwork [5].

4. Conclusions

In this paper, we described the MATN modeling process for an RTSP multimedia presentation system. Detailed definitions and descriptions of the MATN model are presented. Unlike the existing semantic models that only model user interactions, loops, or embedded presentations, our MATN model provides these three capabilities in one framework. Given the example RTSP action diagram, the MATN model demonstrates its strong support for user selections and loops as well as its capability to model the temporal relations and synchronization of media streams. Subnetworks in the MATN model emphasize the modularity and reuse of existing media streams and presentation structures. This feature greatly reduces the design complexity and makes the design easier. Also, the condition/action table of the MATN model makes it capable of supporting different levels of synchronization and QoS.

References

[1] *Cisco IP/TV Content Manager Version 2.0, 1998.*

[2] *Java JMF homepage: http://java.sun.com /products/java-media/jmf/.*

[3] G. Blakowski, J. Huebel, and U. Langrehr. Tools for specifying and executing synchronized multimedia presentations. In *Proc. 2nd Int'l Workshop on Network and Operating System Support for Digital Audio and Video*, pages 271–279, 1991.

[4] S.-C. Chen and R. Kashyap. A spatio-temporal semantic model for multimedia presentations and multimedia database systems. *IEEE Trans. on Knowledge and Data Eng.*, 13(4):607–622, July/August 2001.

[5] S.-C. Chen, R. Kashyap, and A. Ghafoor. *Semantic Models for Multimedia Database Searching and Browsing*. Kluwer Academic Publishers, 2000.

[6] S.-C. Chen, M.-L. Shyu, and R. L. Kashyap. Augmented transition network as a semantic model for video data. *International Journal of Networking and Information Systems, Special Issue on Video Data*, 3(1):9–25, 2000.

[7] S.-C. Chen, M.-L. Shyu, C. Zhang, and J. Strickrott. A multimedia data mining framework: Mining information from traffic video sequences. *Journal of Intelligent Information System, Special Issue on Multimedia Data Mining*, 19(1):61–77, July 2002.

[8] P. N. et al. Revised report on the algorithmic language algol 60. *Comm. ACM*, 6(1):1–17, Jan. 1963.

[9] M. Handley and V. Jacobson. Sdp: Session description protocol. *IETF RFC 2327*, Apr. 1998.

[10] S. Kleene. *Representation of Events in Nerve Nets and Finite Automata, Automata Studies*. Princeton Univ. Press, Princeton, N.J., 1956.

[11] A. V. Lamsweerde. Formal specification: a roadmap. In *ICSE - Future of SE Track 2000*, pages 147–159, 2000.

[12] S.-T. Li, I. Chen, and H. Hsieh. *An Open RTSP-based Multimedia Presentation System*. 7th IEEE 2001 International Conference on Distributed Multimedia Systems, Taipei, Taiwan, Sept. 2001.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. Rtp: A transport protocol for real-time applications,. In *IETF RFC-1889, Audio-Video Transport Working Group*. Internet Engineering Task Force, Jan. 1996.

[14] R. Scowen. *Extended BNF A generic base standard (EBNF). ISO 14977.*

[15] W.Woods. Transition network grammars for natural language analysis. *Comm. ACM*, 13:591–602, Oct. 1970.