Modeling Snapshot Browsing for Web-based Integrated Surveillance Services

Shu-Ching Chen
School of Computer Science
Florida International University
Miami, FL 33199, USA
chens@cs.fiu.edu

Sheng-Tun Li
Department of Information Management
National Kaohsiung First University of Science and Technology
1 University Road, Yenchao, Kaoshiung
TAIWAN 824, R.O. C.
stli@ccms.nkfust.edu.tw

Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
shyu@miami.edu

Abstract

In this paper, we incorporate the *augmented transition network (ATN)* model into a distributed surveillance services (DSS) system. The DSS system is a distributed object-based system that integrates the next generation Internet-based applications in the architecture level and in the system design level. The DSS system establishes a firm basis for supporting a variety of potential cost-effective applications in a plug-and-play manner. By utilizing the ATN control structures, complicated browsing operations involving user interactions can be easily implemented for the DSS.

1 Introduction

A security/surveillance service that allows users to remotely watch real-time snapshots on the spot in a universal manner, to retrieve historical suspicious scenarios, to access the heterogeneous media, and even to receive urgent signals in the form of email or pager is important for many applications such as media-on-demand, video conference, and visual query methodology (Ashok and Bansal, 1998). The seamless integration of the overwhelming World Wide Web and the emerging Java technology endorses the universal accessibility to the surveillance service.

In our previous study (Li et al, 1999), a distributed surveillance services (DSS) system was developed to discuss the interoperability issues in heterogeneous computing environments. The ‘‘bridging technologies” (Baker et al, 1997), in particular, the seamless integration of CORBA/RMI/Java/Web [(OMG), (Javasoft)] for supporting integrated surveillance services, were adopted. This DSS system consists of three components - the Capturer Server, Surveillance Server, and Client Monitor. The Capturer Server is equipped with an H.263-compliant video camera. The snapshot signals on the spot are periodically grabbed by Snapshot Grabber and put into the form of an integer array. The Surveillance Server plays the middleware between Capturer Server and Surveillance Monitor to alleviate the overhead of snapshot grabbing in Capturer Server. The Surveillance Monitor offers a variety of services such as browsing snapshots on the spot and image archives, storing images monitored to the local repository on the client side, obtaining a local hardcopy of the image upon request, sending an alert pager call under suspicious situations, and forwarding images with messages via SMTP Client.

However, there is no specific semantic model proposed to model the DSS system operations. In this paper, a semantic model called the augmented transition network (ATN) [(Chen and Kashyap, 1997), (Chen and Kashyap, 2000)] is proposed to facilitate the systematic design for the DSS. By utilizing the ATN control structures, complicated browsing operations involving user interactions can be easily implemented for the DSS.

This paper is organized as follows. Section 2 describes how to use ATN to model snapshot browsing in surveillance monitor. Section 3 concludes the paper.

2 Using ATN to Model Snapshot Browsing in Surveillance Monitor

2.1 ATN

The *augmented transition network (ATN)* is a semantic model for multimedia presentations, multimedia database searching, and multimedia browsing [(Chen and Kashyap, 1997), (Chen and Kashyap, 2000)]. The arcs in an ATN represent the time flow from one state node to another. An arc represents an allowable transition from the node at its tail to the node at its head, and the labeled arc represents the transition function. An input string is accepted by an ATN if there is a path of transitions which corresponds to the sequence of symbols in the string and which leads from a specified initial state to one of a set of specified final states. Each nonterminal symbol consists of a subnetwork which can be used to model detailed information for the corresponding nonterminal symbol.

The arc types together with the notation need to be defined. We adopted the following notation and definition as in (Allen, 1995).

- **Push** arc: succeeds only if the named network can be successfully traversed. The state name at the head of arc will be pushed into a stack and the control will be passed to the named network.
- **Pop** arc: succeeds and signals the successful end of the network. The topmost state name will be removed from the stack and become the return point. Therefore, the process can continue from this state node.
- **Jump** arc: always succeeds. This arc is useful to pass the control to any state node.

2.2 Surveillance Monitor

Surveillance Monitor offers a variety of services; namely, browsing snapshots on the spot and image archives, storing images monitored to the local repository on the client side, obtaining a local hardcopy of the image upon request, sending an alert pager call under suspicious situations, and forwarding images with messages via SMTP Client. The components of Surveillance Monitor are also fully coded as Java applets and can run on any Java-compliant Web browsers, e.g. Netscape Communicator 4.07. To relieve the restriction of the sandbox security model, the applet is digitally signed and is packaged as a JAR format for reducing the loading time and easing the installation of mobile code on the client. When downloading the signed applet, the end-user is prompted a message for confirming the certificate with the applet. The end-user may choose granting, denying or viewing the certificate. Figure 1 gives the outlook of the user interface of Surveillance Monitor. There are several time instants provided on the right sub-window of Figure 1 to allow the users to select the particular snapshots they want to browse. The time instants selected by the user are highlighted as shown in Figure 1.

2.3ATN Operations for Surveillance Monitor Browsing

Table 1 lists part of the time instants shown in Figure 1. Four time durations, denoted by d_1 to d_4, are used to capture those time instants. Here, each time instant is represented by a time instant symbol (e.g., t_4), and each time duration consists of three time instants. The fourth column of

Table 1 indicates whether the time instant is selected by the users. A time instant is selected if the users want to browse the corresponding snapshot. For example, the snapshots for the time instants with time instant symbols t_5 to t_{12} are selected for browsing. This specified browsing sequence will be kept by the ATN model to guide the traversing steps. Figure 2 shows the ATN and its four subnetworks to model the Surveillance Monitor browsing scenario of Table 1. Figure 2(a) is the ATN and Figure 2(b) to Figure 2(e) are the four subnetworks for the four durations, respectively.

The trace of ATN of the example browsing sequence (shown in Table 1) is presented in Table 2. This detailed trace is used to illustrate how the ATN and its subnetworks model the Surveillance Monitor browsing scenario.

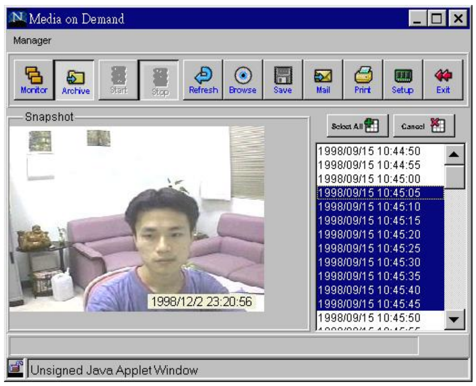

Figure 1: The outlook of user interface control in Surveillance Monitor.

Duration	**Time instant Symbol**	**time instant**	Selected or not?
d_1	t_1	1998/09/15 10:44:50	No
	t_2	1998/09/15 10:44:55	No
	t_3	1998/09/15 10:45:00	No
d_2	t_4	1998/09/15 10:45:05	Yes
	t_5	1998/09/15 10:45:10	Yes
	t_6	1998/09/15 10:45:15	Yes
d_3	t_7	1998/09/15 10:45:20	Yes
	t_8	1998/09/15 10:45:25	Yes
	t_9	1998/09/15 10:45:30	Yes
d_4	t_{10}	1998/09/15 10:45:35	Yes
	t_{11}	1998/09/15 10:45:40	Yes
	t_{12}	1998/09/15 10:45:45	Yes

Table 1: Part of the time durations, time instants, and the corresponding selections for the time instants in Figure 1

Step 1: The current state is in *P/* where *P* represents the starting of the browsing. The input symbol is $d_1\&d_2\&d_3\&d_4$. This input symbol denotes the time instants in durations d_1 to d_4 are displayed concurrently for users to make selection (as shown on the right sub-window of Figure 1). This input symbol is read so that users can select a time instant belonging to a duration.

Step 2: Based on the specified browsing sequence, the input symbol d_2 is read. Since d_2 is a subnetwork name, the state name P/d_2 is pushed into a stack. A stack follows the last-in-first-out (LIFO) policy that only allows retrieving the topmost state name first.

Step 3: The control is passed to the subnetwork with starting state name $d_2/$ (as shown in Figure 2(c)). The input symbol t_4 is read which indicates that the snapshot image for the time instant t_4 is selected for browsing.

Step 4: The current state is in d_2/t_4. The outgoing *Jump* arc passes the control back to $d_2/$ state.

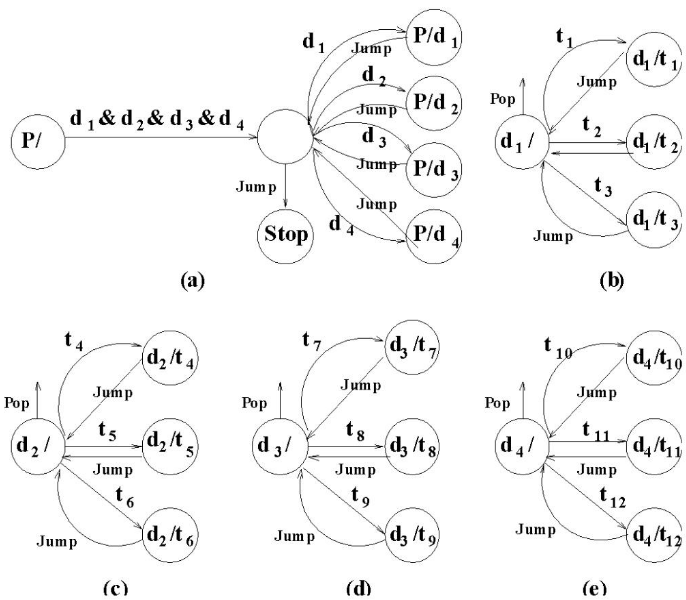

Figure 2: (a) ATN to model the Surveillance Monitor browsing scenario of Figure 1 with respect to four durations. (b) to (e) are the subnetworks for durations d_1 to d_4, respectively.

Step 5: The input symbol t_5 is read which indicates that the snapshot image for the time instant t_5 is selected for browsing.
Step 6: The current state is in d_2/t_5. The outgoing Jump arc passes the control back to $d_2/$ state.
Step 7: The input symbol t_6 is read which indicates that the snapshot image for the time instant t_6 is selected for browsing.
Step 8: The current state is in d_2/t_6. The outgoing Jump arc passes the control back to $d_2/$ state.
Step 9: The *Pop* arc signals the successful end of the subnetwork. The topmost state name P/d_2 is removed from the stack and becomes the return point.
Step 10: The current state is in P/d_2. The outgoing *Jump* arc passes the control back to $P/d_1\&d_2\&d_3\&d_4$ state.
Steps 11 - 19: Perform the similar operations for duration d_3 (as shown in Figure 2(d)) with the time instants t_7, t_8, and t_9 selected.
Steps 20 - 28: Perform the similar operations for duration d_4 (as shown in Figure 2(e)) with the time instants t_{10}, t_{11}, and t_{12} selected.
Step 29: The current state is in $P/d_1\&d_2\&d_3\&d_4$. The outgoing *Jump* arc reaches the final state.
Step 30: Stop the browsing sequence.

3 Conclusion

The goals of a distributed surveillance services (DSS) system is to provide a cost-effective, reliable, flexible, open and universal environment for multimedia message exchanges yet accommodate the overwhelming proliferation of online information and the heterogeneous systems. Under such a system, a proper model and control structures is needed in order to have a control mechanism for the security/surveillance services. In this paper, the *augmented transition* *network (ATN)* model was incorporated into the DSS system to facilitate the systematic design for the DSS. By utilizing the ATN control structures, complicated browsing operations involving user interactions can be easily implemented for the DSS.

Step	Current State	Input Symbol	Backup States
1	$P/$	$d_1\&d_2\&d_3\&d_4$	
2	$P/d_1\&d_2\&d_3\&d_4$	d_2 (Push)	P/d_2
3	$d_2/$	t_4	P/d_2
4	d_2/t_4	Jump	P/d_2
5	$d_2/$	t_5	P/d_2
6	d_2/t_5	Jump	P/d_2
7	$d_2/$	t_6	P/d_2
8	d_2/t_6	Jump	P/d_2
9	$d_2/$	Pop	P/d_2
10	P/d_2	Jump	
11	$P/d_1\&d_2\&d_3\&d_4$	d_3 (Push)	P/d_3
12	$d_3/$	t_7	P/d_3
13	d_3/t_7	Jump	P/d_3
14	$d_3/$	t_8	P/d_3
15	d_3/t_8	Jump	P/d_3
16	$d_3/$	t_9	P/d_3
17	d_3/t_9	Jump	P/d_3
18	$d_3/$	Pop	P/d_3
19	P/d_3	Jump	
20	$P/d_1\&d_2\&d_3\&d_4$	d_4 (Push)	P/d_4
21	$d_4/$	t_{10}	P/d_4
22	d_4/t_{10}	Jump	P/d_4
23	$d_4/$	t_{11}	P/d_4
24	d_4/t_{11}	Jump	P/d_4
25	$d_4/$	t_{12}	P/d_4
26	d_4/t_{12}	Jump	P/d_4
27	$d_{4/}$	Pop	P/d_4
28	P/d_4	Jump	
29	$P/d_1\&d_2\&d_3\&d_4$	Jump	
30	Stop		

Table 2: The trace of ATN for the browsing sequence in Figure 2.

References

[Allen, 1995] Allen, J. (1995); Natural Language Understanding; The Benjamin/Cummings Publishing Company, Inc

[Ashok and Bansal, 1998] Ashok, S. and V. K. Bansal (1998); Java: Network-centric Enterprise Computing; Computer Communications, Vol. 20, (pp. 1467-1480)

[Baker et al, 1997] Baker, S., Cahill, V. and P. Nixon (1997); Bridging Boundaries: CORBA in Perspective; IEEE Internet Computing, (pp. 52-57)

[Chen and Kashyap, 1997] Chen, S.-C. and R. L. Kashyap (1997); Temporal and Spatial Semantic Models for Multimedia Presentations; Proc. International Symposium on Multimedia Information Processing (pp. 441-446)

[Chen and Kashyap, 2000] Chen, S.-C. and R. L. Kashyap (2000); A Spatio-Temporal Semantic Model for Multimedia Presentations and Multimedia Database Systems; Accepted for publication in IEEE Transactions on Knowledge and Data Engineering

[Javasoft]JavaSoft Company; Java Remote Method Invocation; http://www.javasoft.com/products/jdk/rmi/

[Li et al, 1999] Li, S.-T., Chen, W.-S., Chang, H.-C. and C.-M. Lin (1999); Building Integrated Surveillance Services Using Java/CORBA Bridging Technologies in Heterogeneous Computing Environments; Proc. 11th International Conference on Systems Research, Informatics and Cybernetice

[OMG] OMG; CORBA; http://www.omg.org/corba/