

# The Distributed Multimedia Presentation System Based on Extended Timed Petri Nets

Timothy K. Shih, Huan-Chao Keh, Jiung-yao Huang, Lawrence Y. Deng,  
Teh-Sheng Huang and Ding-Jung Chiang

Dept. of Computer Science and Information Engineering Tamkang  
University, Tamsui, Taipei Hsien, Taiwan 251, R.O.C.  
E-mail: tshih@cs.tku.edu.tw

## Abstract

Currently, multimedia presentation technologies among the network are most often used in many communication services. Examples of those applications include video-on demand, interactive TV and the communication tools on a distance learning system and so on. In this paper, we describe how to present different multimedia objects on a web presentation system. The distributed approach is based on an extended timed Petri net model. Using characterization of extended time Petri net, we express the temporal behavior of multimedia objects; on the other hand, we introduce the concepts of user interaction. The main goal of our system is to provide a feasible method to represent a schedule and navigation of different multimedia objects with user interaction. In addition, users can dynamically modify and verify different kinds of conditions during the presentation. To verify the structural mechanism, we implement an algorithm using the Petri net diagram, analyzing the model by time schedule of multimedia objects, and produce a synchronous set of multimedia objects with respect to time duration. Another, we also proposed a communication tool that follows a floor control mode, which provides four types of control (free access, equal control, group discussion, and direct contact). These control mechanisms are sufficient to the use of virtual university environment. Furthermore, we discussed our web presentation based on software metrics.

**Keywords:** Petri Net, Distributed Multimedia Presentation, Distance Learning, Floor Control

## 1 Introduction

To control and demonstrate different types of multimedia objects is one of important functions in distributed multimedia presentation system. Unfortunately, we saw many “Black magic” compromised multimedia presentation systems; there is little theory to describe the methodologies of such compromised system. The concept of our model is based on the Petri net [1-3]. Petri net is a graphical and mathematical modeling tool applicable to many systems. Its features can be used with both practice and theory. Thus, it provides a powerful medium of communication between them. Additional extensions have been proposed, and this has led to the following types of Petri nets: the timed Petri net, the stochastic Petri net, colored Petri net, and object-related Petri net [6-12]. The “Object Composition Petri Net”(OCPN) and the “extended Object Composition Petri Net”(XOCPN) are two graphic-based models that propose synchronous theoretical for multimedia. The OCPN is a comprehensive model for specifying timing relations among multimedia data. The XOCPN can specify temporal relationships for the presentation of pre-orchestrated multimedia data, and to set up channels according to the required Qos of the data [4, 5]. These two models lack methods to describe the details of synchronization across distributed platforms and do not deal with the schedule change caused by user interactions in interactive multimedia

systems [13]. However, when considering the network transport issue of multimedia and the floor control with multiple users, OCPN/XOCPN model are not sufficient to deal with those problem. In this paper, we use the extended timed Petri net to construct the web operations on a distance learning system. When multimedia objects are represented on the system, we have to consider different situations of multimedia objects such as asynchronous operations, time scheduling, and flow-control. In addition to system operations, dynamical operations of users are important issues. Thus, we can apply characteristic of Petri net to implement our mechanism and study the theory.

This paper is organized as follows. The Petri net and other extended models of Petri net are introduced in Section 2. Section 3 defines multimedia objects based on Petri net. Section 4 constructs an algorithm for our web system based on the Petri net and uses an example to verify the algorithm and the group communication mechanism with floor control mode. Section 5 discusses multimedia objects presentation that includes software metrics, testing, and maintenance phases based on software engineering. Section 6 gives the conclusions and future work.

## 2 Related Petri Nets Models

Petri net were named after Carl A. Petri who created in 1962 a net-like mathematical tool for the

study of communication with automata. Petri net can be used to model properties such as process synchronization, asynchronous events, concurrent operations, and conflicts or resources sharing. These properties characterize discrete-event systems, and computer-based systems. Petri net as graphical tools provide a communication medium between the user, typically requirements engineer and the customers. Complex requirements specifications, instead of using ambiguous textual descriptions or mathematical notations difficult to understand by the customer, can be represented graphically using Petri net. This combined with the existence of computer tools allowing for interactive graphical simulation of Petri net, puts in hands of the development engineers a powerful tool assisting in the development process of complex systems [1, 2, 3].

This paper focuses on the distributed multimedia presentation applications. The related Petri net models will be introduced in this section briefly.

## 2.1 Petri Net

Petri net is composed of four parts: a set of places  $P$ , a set of transitions  $T$ , an input function  $I$ , and an output function  $O$ . The input and output functions relate to transition and places. The input function  $I$  is a mapping from a transition  $t_j$  to a collection of places  $I(t_j)$ , known as the input places

of the transition. The output function  $O$  maps a transition  $t_j$  to a collection of places  $O(t_j)$  known as the output places of the transition. The structure of a Petri net is defined by its places, transitions, input functions, and output functions [2].

**Definition:** A Petri Net structure,  $C$ , is a four tuple,  $C = (P, T, I, O)$ .  $P = \{ p_1, p_2, \dots, p_n \}$  is a finite set of places,  $n \geq 0$ .  $T = \{ t_1, t_2, \dots, t_m \}$  is a finite set of transactions,  $m \geq 0$ . The set of places and the set of transaction are disjoint,  $P \cap T = \emptyset$ .  $I: T \rightarrow P^\infty$  is the input function, a mapping from transitions to bags of places.  $O: T \rightarrow P^\infty$  is the output function, a mapping from transaction to bags of places.

## 2.2 OCPN Model

Object Composition Petri Net (OCPN) have been proposed in [4, 14, 15] to depict the multimedia synchronization characteristics in an orchestrated presentation. Orchestrated multimedia presentation applications are composed of different media streams such as text, voice, video, image and animation ...etc. These characteristics demonstrated the time instants at which the presentation of objects in the media streams must synchronize. The OCPN augments the conventional Petri net model with durations, and resource utilization on the places in the net [4].

Definition:  $C_{OCPN} = \{T, P, A, D, R, M\}$  where, additionally,  $D: P \rightarrow R$  and  $R: P \rightarrow \{r_1, r_2, \dots, r_k\}$ .  $D$  and  $R$  are mappings from the set of places to the real numbers (durations), and from the set of places to a set of resources, respectively. Associated with the definition of the Petri net is a set of firing rules governing the semantics of the model.

Fire rules:

- (1). A transition  $t_i$  fires immediately when each of its input places contain an unlocked token.
- (2). Upon firing, the transition  $t_i$  removes a token from each of its input places and adds a token to each of its output places.
- (3). After receiving a token, a place  $p_j$  remains in the active state for the interval specified by the duration  $\delta_j$ . During this interval, the token is locked. When the place becomes inactive, or upon expiration of the duration  $\delta_j$ , the token becomes unlocked.

### 2.3 XOCPN Model

Considering the timing relationships among multimedia objects at the presentation period. Extended object composition Petri net (XOCPN) takes into account the demands of isochronous objects, requiring a rate-controlled transmission. XOCPN is a timed Petri-net based model which can

be used to specify the scheduling time of multimedia objects for their transportation, and to configure channels according to the required quality of service of objects [5].

Definition:  $C_{XOCPN} = \{T, P, A, D, R, M, Y, Z\}$  where  $T = \{t_1, t_2, \dots, t_m\}$  is a finite set of transactions,  $m \geq 0$ .  $P = \{p_1, p_2, \dots, p_n\}$  is a finite set of places,  $n \geq 0$ .  $A = \{T * P\} \cup \{P * T\} \rightarrow I$ ,  $I = \{1, 2, \dots\}$  is a set of directed arcs.  $D: P \rightarrow \{d_1, d_2\}$  where  $d_1, d_2$  are real numbers and represent delay before an action, respectively.  $R: P \rightarrow \{r_1, r_2, \dots, r_k\}$  is a mapping from the set of places to a set of object types, corresponding to video, audio, image or textual objects.  $M: P \rightarrow I$ ,  $I = \{1, 2, \dots\}$  is a mapping from the set of places to the integers, which represents the number of tokens in  $s$  specified place.  $Y: P \rightarrow \{\text{Resource-setup, Resource-release, SIU (Synchronization Interval Unit) payout, SIU-transmit, Interstream-synchronize}\}$  is a mapping from the set of places to a set of action types to be performed during communication.  $Z: P \rightarrow \{\text{address for Qos parameters, address for an Synchronization interval unit, address for synchronization requirements}\}$

### 2.4 DOCPN Model

J. S. Yang et al. proposed a prioritized Petri

nets model [13]. It extends traditional Petri nets with priority. The different of priority Petri nets from traditional Petri nets lies in the introduction of priority into functions. Input functions are treated unequally at some transitions. A priority input event arrival at a transition may force firing without waiting for the other arrival of non-priority events. By using these features, we can deal with the time schedule dominates an event transition in the real communication world. Even though some conditions are not yet, an event will occur when its time schedule is due. This will happen when real time constraint is concerned and when a downgraded service can be achieved with out some pre-specified resources. The priority Petri model can be applied to these cases by using a clock or time schedule and priority arc driving those time-sensitive transitions.

Definition: A priority Petri Net structure,  $C$ , is a five-tuple,  $C = (P, T, I, I_p, O)$ .  $P = \{ p_1, p_2, \dots, p_n \}$  is a finite set of places,  $n > 0$ .  $T = \{ t_1, t_2, \dots, t_m \}$  is a finite set of transactions,  $m > 0$ . The set of places and the set of transaction are disjoint,  $P \cap T = \emptyset$ .  $I: T \rightarrow P^\infty$  is the input function, a mapping from transitions to bags of places.  $I_p: T \rightarrow P^\infty$  is the priority input function, a mapping from transitions to bags of places.  $O: T \rightarrow P^\infty$  is the output function, a mapping from transaction to bags of places.

Fire rules: A transaction with non-priority input events would fire when all events are complete and ready. A transaction with a priority input event, without waiting for other non-priority events. For the same priority events concurring at a transaction, we apply the "AND" rule. A place with a token and several transaction enabled from this place will fire the transaction with a priority arc from this place.

The Distributed Object Composition Petri Net (DOPCN) model includes the following properties:

- (1). Inheriting the characteristics of Petri Net, that is, waiting at a transition until all input signals arrived, and then firing concurrently.
- (2). The arrival of priority input at a transition may cause firing of a transition without waiting for other non-priority input concurring at the same transition.
- (3). Using the synchronous methods inherited from OCPN and XOPCN to achieve synchronization among inter-media objects.
- (4). Extended OCPN to a distributed environment that handles asynchrony across platforms using a global clock.
- (5). Adding user interaction control into OCPN, thus user interaction can be a new important factor in synchronization.

### 3 Basic Definitions of Distributed Time Petri Nets and Multimedia Specification

We define multimedia objects representation based on the characteristics of the Petri net. As a graphical tool of Petri net, the followings are basic properties of a Petri net [1,2,8] and the description of multimedia objects:

**Definition 3.1.** A Petri net is a 5-tuple,  $PN = (P, T, F, W, M_0)$  where:

- $P = \{P_1, P_2, \dots, P_m\}$  is a finite set of places,
- $T = \{T_1, T_2, \dots, T_n\}$  is a finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$  is a set of arcs (flow relation),
- $W: F \rightarrow \{1, 2, 3, \dots\}$  is a *weight function*,
- $M_0: P \rightarrow \{0, 1, 2, \dots\}$  is the *initial marking*,
- $P \cap T = \emptyset$  and  $P \cup T \neq \emptyset$ .

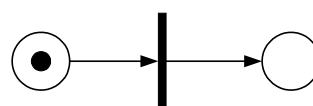
The structure of Petri net  $N = (P, T, F, W)$  without any specific initial marking is denoted by  $N$ . The generic components of Petri net include a finite set of places and a finite set of transitions. Petri net is a finite bipartite graph. Its places are linked with transitions in turn are connected to the output places. For a given place, there are input and output transitions defined.

**Definition 3.2.** A transition that has no input places is called a *source transition* and a transition that has no output places is called a *sink transition*. Note that a source transition is unconditionally enabled or fired and that the firing or enabling of a sink transition consumes tokens, but cannot produce any.

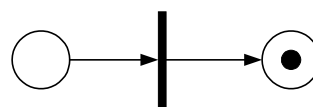
**Definition 3.3.** The distribution of tokens over places is called a marking of the net. A transition may enable or fire when each of its input places contains at least one token. The firing of a transition results in removing tokens from its input places and adding tokens to the output places.

**Definition 3.4.** If a place is both an input and output of a transition, a pair of a place and a transition is called a *self-loop*.

**Definition 3.5.** If a Petri net has no self-loops, it is said to be *pure*. If all of arc weights of a Petri net are 1's, it is said to be *ordinary*.



1(a) An initial marking before transition firing



1(b) A new marking after transition firing

Figure 1. State transfers by transition firing

A marking represents the state of a system, which changes when a transition fired to produce a new marking. An example of Petri net is given in figure 1.

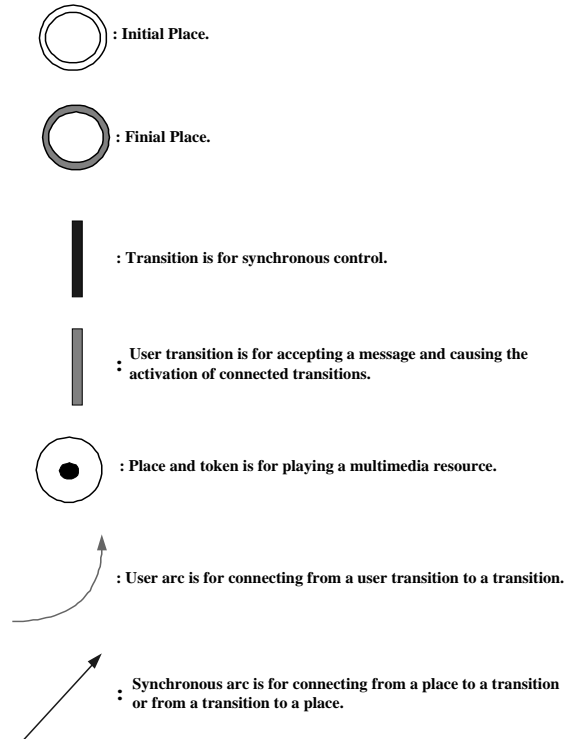
**Definition 3.6.** Multimedia objects representation specification :

- A multimedia object is as a place node including a unique token and a transition node displays an event enabling or conditional sufficiency in the multimedia Petri net.

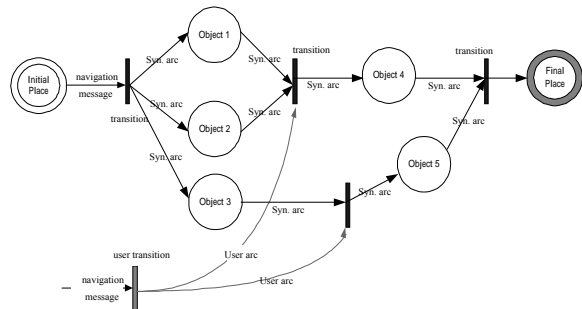
- A place node holds a token and time duration. It controls a multimedia resource to be played for the time duration. A transition node controls synchronization and it is fired only after each place node adjacent to the transition releases the token. The nodes of place and transition connect via synchronous arcs in a Petri net.

- We add *user transitions* and *user arcs* to our multimedia Petri net. A user transition receives a navigation message form the user before it is fired. A user transition is directly connected to some transitions. The activation of user transition can interrupt the demonstration of an arbitrary presentation window and cause the activation of the connected transitions simultaneously.

As illustrated in figure 2, the followings are components of multimedia objects in a Petri net and an example for multimedia objects based on our multimedia Petri net:



(a) The definitions of multimedia objects



(b) The overview of a multimedia web

Figure 2. The representation of a multimedia objects

#### 4 A Management Algorithm Based on Timed Petri Net

To design the algorithm based on the Petri net [7, 9, 10], we consider two points. The first is the algorithm independent to any operating system. The second is the realization and feasibility of our web system.

**Algorithm:** multimedia objects synchronous set on the Petri net at the run\_time

**Input:** A diagram of a Petri net

**Output:** multimedia objects synchronous set

##### Procedure

multimedia\_objects\_synchronous\_set

Max: integer;

Index: integer;

##### Begin

Sort the transition\_node to a transition\_list by topological sort algorithm;

Setup the first transition\_node to Index;

Computing the number of transition\_list;

Setup the number of transition\_list to Max;

Initial the node pointed to by the first navigation\_message to node\_time

= 0;

**For** index max **do** **Begin**

**For** each incoming edge **do** **Begin**  
     estimated\_time = node\_time +  
     edge\_duration;

    Setup node\_time of some node to the  
     maximum of all

    estimated\_time of incoming edges;

**End;**

Setup node\_time to transition\_node;

Index = Index + 1;

**End;**

**For** transition\_list list\_end **do** **Begin**

**Cobegin**

Process 1:

    Play the resource concurrently at some  
     transition\_node during

    node\_time;

Process 2:

**If** user interaction **Then** interrupt;

    Wait for instructions;

**Coend;**

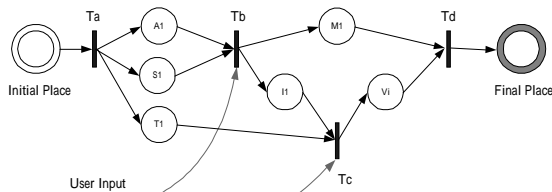
**End;**

**End;**

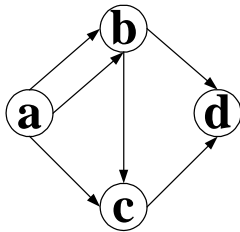


### 4.1 Implementation by Above Algorithm

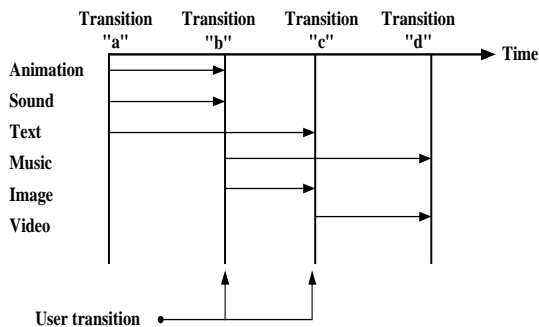
We give an example to implement the above algorithm in figure3. The example is a diagram including multimedia objects such as animation, sound, image, music, text, and video multimedia resources. Transitions “a”, ”b”, ”c”, and “d” control



3(a) An example by the above algorithm



3(b) Topological sort of the transition node



3(c) Time schedule of the example by above algorithm

Case1 : Without firing of user transition

**a** : {animation,sound,text}

**b** : {music,image,text}

**c** : {music,video}

**d** : End

Case2 : With firing of user transition

**b** **c** : {music,image,video}

**d** : End

(3d) The synchronous set order of firing of transitions

Figure 3. The description of an example by above algorithm

synchronous operations for the diagram. In addition, we will consider a situation including user transition in the diagram.

There are two cases discussed in the example.

Case 1: In figure3, under the condition that the user transition is not fired, the transition order is “a”, ”b”, ”c”, and “d” by topological sort of the algorithm. When transition “a” is fired, the synchronous set is {animation, sound, text}. When transition “b” is fired, multimedia objects animation and sound completed. The place named “text” is continuing to play and the places named “music” and “image” begins to play. So the synchronous set is {music, image, text}. When transition “c” is fired, the multimedia resources “text” and “image” complete, and the multimedia object “video” begins

to play. So the synchronous set is {music, video}. When transition “d” is fired, all operations are completed. The synchronous sets are given in figure (3d) with respect to some transition duration. Case 2: In figure (3d), If user transition is fired, transition “b” and “c” are fired, but transition “a” is not fired. So the synchronous set is {music, image, video}. The multimedia objects including “animation” and “sound” do not play because the transition “a” is not fired. The operation of the synchronous set is given in figure (3c) and figure (3d).

#### 4.2 Distributed Multimedia Presentation

##### Environment

The multimedia presentation system in distributed environment can be taken it into a communication tool for virtual conferencing or distance learning. The communication tools need to considerate the group communication and floor control mechanism. In order to achieve these objectives, the distributed multimedia presentation system (DMPS) needs to build a global clock first (as shown in figure 4). The global clock is a standard time in the present period of the client sides. A communication tool which be held “Synchronous” one is because of the bounded delay time. The global clock not only provides the global time frame facility but also control the higher priority of user interaction floor control.

For instance, in a case of group communication, user need to initiate the group first, then users can set their communication media needed via our DMPS tools (as shown in figure 4 DMPS communication window).

The DMPS server builds a communication group and initiates a global clock when the client side has initiated the communication configuration. The global clock admission control is under centralized mode. It has the highest priority to handle the transition enforced to fire immediately or not. If the clock in client side is faster than global clock, the current transition will not fire until global clock arrives. On the other hand, if the local clock in client side is slower than global clock, the transition will be fired without delay.

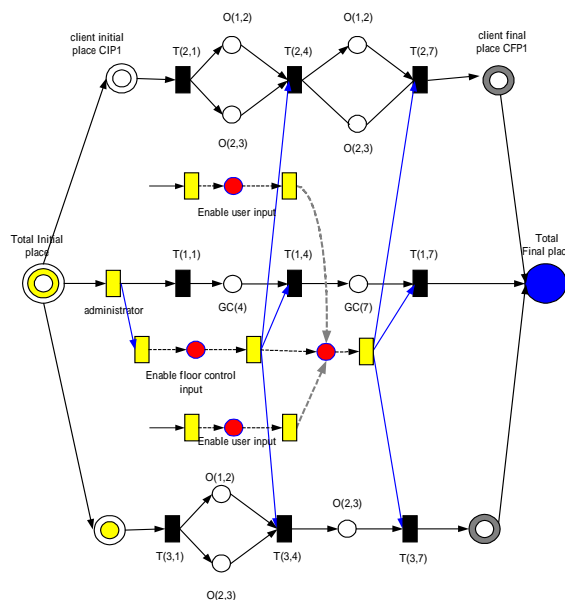


Figure 4 : An overview of distributed multimedia presentation Petri net

In the presenting period, user can request the floor control and change the presenting media. The floor control include four modes:

1. Free Access
2. Equal Control
3. Group Discussion
4. Direct Contact

Free access means everyone (e.g. including session chair and participants) can send the message to the message-window or whiteboard. This mode is like general discussion with no privacy and priority. We have a limitation of speak in equal control mode. In this mode, there is only one (session chair or participants) can deliver at the same time until the floor control token passed by the holder.

Participants are encouraged to propose their ideas in some time. So, another small group discussion mode is provided. The manner is that a user can create a new group to invite others. For example, if user A wants user B to receive his invitation, he can send an inviting message. User B can make a decision to accept or not. If yes, user B will be chosen as listen group of user A, and the user A will be the session chair in his small group. Everyone can choose one to receive the message actively. Therefore, all participants in the same group can send message to each other, we regard it as private communication group. The fourth floor control mode is direct contact. Actually, it is similar to the third mode. It means two people can communicate directly in a private window and communicate with others via free access, equal control, and direct contact at the same time.

The floor control model is managed by group administration of the DMPS server. All the users' floor control input requests are sent to the server, the server will take the messages with their rationality to handle the floor control in group



5(a) DMPS communication window for student



5(b) DMPS communication window for teacher

Figure 5 An example of the DMPS communication

communicating period. If the users' floor control requests are permitted, the requests will be combined with the global clock control and with the same highest priority.

## 5 Software Testing to Multimedia Presentation

After the user design the schedule and layout of the presentation via our Petri net tool and graphical user interface, our multimedia system generates the presentation automatically. Unlike the typical software development, no programs have to be written. This generation strategy is also used by most multimedia presentation systems available on the market [11, 12]. The advantage is that if the specification and design of a presentation is correct, our system guarantees the correct implementation.

It is possible to obtain the software metrics of a presentation. Our system is able to calculate software complexity of a presentation based on the following criteria:

- Number of presentation windows
- Number and sizes of multimedia resources
- Number of transitions
- Number of user transitions
- Number of dynamic mutations

- Number of synchronous arc messages
- Number of user arc messages
- Number of navigation messages

Software metrics of a presentation indicate the complexity and the amount of efforts to test the presentation. Complexity of the presentation also indicates the effort to collect presentation resources and the size of potential disk storage used. Presentation testing is essentially important to ensure a smooth demonstration without missing resources and error navigation sequences. It is hard to perform a complete testing due to the amount of navigation sequences. Fortunately, our system facilitates testing by means of an automatic tool. The tool traverses the presentation based on the following testing criteria:

- Every presentation window should be tested
- Every multimedia resource should be tested
- Every navigation message should be tested
- Every transition should be tested
- Every user transition should be tested
- Every synchronous arc should be tested
- Every user arc should be tested

The testing of a presentation window includes the Petri net itself, which may accept navigation messages sent to transitions or user transitions. Synchronization of multimedia resources used by places is tested as well. Moreover,

selections, assignments, and conditions are tested. Since there is no particular sequence that the user navigates the presentation, the path topology of testing is arbitrary. Our testing tool traverses a graph consists of nodes (i.e., selections, assignments, conditions, and starting transitions/user transitions) and links (i.e., navigation messages) based on a modified breadth first search (BFS) algorithm. Upon the activation of a presentation window, each selection, assignments, or conditions are kept in a queue for further traverse. When the internal state of the presentation is changed due to the change of state variables (which may cause navigation change, layout change, or resource change), some modified nodes are added to the queue used in the BFS algorithm for re-testing. The testing tool keeps a testing log file indicating a testing sequence.

Another feature of our system is the evaluation tool of the presentation design status. It is very likely that a user designs an incomplete or inconsistent presentation. The following is a list of potential mistakes that a user may create in the presentation:

- Non-used state variables
- Non-used multimedia resources
- Non-used assignments
- Non-used conditions
- Incomplete navigation messages
- Incomplete selections

- Incomplete assignments
- Incomplete conditions
- Incomplete presentation windows (i.e., if any component of the window is incomplete)
- Inconsistent presentation windows (i.e., unbalanced links)
- Undocumented presentation windows

Non-used items, such as state variables not used in any condition, multimedia resources not linked to places, or the tool marks assignments and conditions not triggered by navigation messages. Incomplete items, with one of their properties missing, are also marked. A presentation window is inconsistent if the window and its refinement have different links (i.e., the unbalanced link problem). An inconsistent or undocumented presentation window is also marked. The user is able to check the list of problematic items before releasing the presentation.

It is possible that the user will release different version of a presentation. Therefore, presentation maintenance is an important issue. A presentation specification in our system contains a number of presentation windows. Each presentation window has a documentation box. It is important to document the presentation script in the box for further reference. Moreover, how to keep track of different versions of a presentation window is

important. In a multimedia presentation database we developed, presentation windows as well as presentation resources are reusable objects. Each of these objects has several versions. A presentation is a composed object from these reusable objects. The database system serves as an underlying supporting tool for several multimedia presentation systems that we have developed, including the one we propose in this paper.

## 6 Conclusions

The main goal of our web system provides an interactive Petri net model. We suggest a method constructing a web structure on a distance learning system. We hope that the algorithm is independent to any operating system applied different platforms. In addition, we provide a friendly user interface to access multimedia resources on the system. The prototype system was developed on MS Windows 98 or MS Windows NT to justify our approach. We will focus on the performance of the system and improve the web multimedia presentation in the future research. We have some contributions in the paper. We provide a model for web multimedia presentation on distance learning by the extension of basic Petri net. With the system, we hope to bring a feasible method including structural design and analysis for the design of the distributed multimedia presentation. The model can be used for multimedia

presentation, in business, education, and others. Another contribution is to provide software metrics, testing and maintenance techniques for web multimedia presentation in software engineering field.

## References

- [1] Tadao Murata, "Petri Nets: Properties, Analysis and Applications," Proceedings of The IEEE Vol. 77, No. 4, 1989.
- [2] J. L. Peterson, "Petri Net Theory and the Modeling of Systems," Englewood Cliffs, NJ: Prentice-Hall, Inc. 1981.
- [3] Richard Zurawski and MengChu Zhou, "Petri Nets and Industrial Application: A Tutorial", IEEE Transactions on Industrial Electronics, Vol. 41, No. 6, Dec. 1994.
- [4] T. D. C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal on Selected Areas in Communications, pp.413-427, Apr. 1990.
- [5] M. Woo, N. U. Qazi and A. Ghafoor, "A Synchronous Framework for Communication of Pre-orchestrated Multimedia Information," IEEE Network, pp. 52-61, Jan./Feb. 1994.
- [6] S. Crespi-Reghezzi and D. Mandrioli, "Petri nets and szilard languages," Information and

Control, Vol. 33, No. 2, pp. 177-192, Feb. 1977.

[7] E. W. Mayr, "An algorithm for the general Petri net reachability problem," SIAM, J. Comput. Vol. 13, No. 3, pp. 441-460, Aug. 1984.

[8] Toshinori Suzuki and Sol. M. Shatz, "A Protocol Modeling and Verification Approach Based on a Specification Language and Petri Nets," IEEE Transactions on Software Engineering. Vol. 16, No. 5, May 1990.

[9] M. A. Holiday and Mary K. Venon, "A generalized timed Petri net model for performance analysis," IEEE Trans. Software Eng., vol. SE-13, no. 12, pp.1297-1310, Dec. 1987.

[10] J. L. Peterson, "Petri nets," ACM Computing Surveys, vol. 9, no. 3, pp. 223- 252, Sept. 1977.

[11] G. A. Schloss and M. J. Wynblatt, "Presentation Layer Primitives for the Layered Multimedia Data Model," In Proceedings of the IEEE 1995 International Conference on Multimedia Computing and Systems, May 15-18, Washington DC, pages 231-238, 1995.

[12] R. H. Thomas and e. a. Harry C. Forsdick, "Diamond: A Multimedia Message System Built on a Distributed Architecture," IEEE Computer, December: 65-78, 1985.

[13] Sheng-Uei Guan, Hsiao-Yen Yu and Jen-Shun Yang, "A Prioritized Petri Net Model and its Application in Distributed Multimedia Systems,"

IEEE Transaction on Computer, Vol. 47 No. 4, April 1998.

[14] T. D. C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," IEEE Journal on Selected Areas in Communications, pp1368-1382, Dec. 1991.

[15] T. D. C. Little, "Synchronization for Distributed Multimedia Database Systems," Ph. D. Dissertation, Syracuse University, Aug. 1991.



Dr. Shih is a Professor and the Chairman of Department of Computer Science and Information Engineering at Tamkang University, Taiwan, R.O.C. His research interests include Multimedia Computing and Networking, Software Engineering, and Formal Specification and Verification. He was a faculty of the Computer Engineering Department at Tamkang University in 1986. In 1993 and 1994, he was a part time faculty of the Computer Engineering Department at Santa Clara University. He was also a visiting professor at the University of Aizu, Japan in summer 1999. Dr. Shih received his BS and MS degrees in Computer Engineering from Tamkang University and California State University, Chico, in 1983 and 1985, respectively. He also received his Ph.D. in Computer Engineering from Santa Clara University in 1993. Dr. Shih has published over 200 papers and participated in many international academic activities. Dr. Shih has received many research awards, including Tamkang University research awards, NSC research awards (National Science Council of Taiwan), and IIAS research award of Germany. He also received many funded research grants from NSC, from the Institute of Information Industry, Taiwan, and from the University of Aizu,

Japan. Dr. Shih has been invited frequently to give tutorials and talks at domestic and international conferences and research organizations. The contact address of Dr. Shih is Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei Hsien, Taiwan E-mail: tshih@cs.tku.edu.tw, Fax: +886 2 26209749, Phone: +886 2 26215656 x2743, x2616, <http://www.mine.tku.edu.tw/chinese/teacher/tshih.htm>.



Dr. Keh is an associate professor in Information Engineering and Computer Science at Tamkang University since 1991. His research interests include Object-Oriented Methodology, Parallel Architectures, Database Systems, and Multimedia Computing. Dr. Keh received his BS in engineering and MS in Computer Science from National Chung-Hsing University, Taiwan and from Oregon State University, in 1980 and 1985, respectively. He also received his Ph.D. in Computer Science from Oregon State University in 1991.



Jiung-yao Huang is an Associate Professor in the Department of Computer Science and Information Engineering at the Tamkang University of Taiwan, ROC. He received his PhD degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst in 1993. Prior to receiving his MS degree in Computer Science from the Tsing-Hua University, Taiwan, in 1988, he was graduated from the Department of Applied Mathematics at the Chung-Hsing University, Taiwan, in 1983. His research interests include computer graphics, multi-user virtual reality, distributed real time system, and multimedia system. He has been a member of IEEE and ACM since 1990, and joined the Society for Computer Simulation in 1996. Currently, he is also the director of Cham-Pion Incubator Center at Tamkang

University, Taiwan. Reader can contact the author at [jhuang@mail.tku.edu.tw](mailto:jhuang@mail.tku.edu.tw).



Lawrence Y. Deng received the Bachelor and Master degrees in department of CSIE form Tamkang University, Taipei, Taiwan, in 1997 and 1999. He is currently a PHD Candidate in the department of CSIE at Tamkang University. His currently research interests include multimedia computing, communication protocol, mobile computing and content-based retrieval.



Teh-Sheng Huang received the Bachelor degree in Math from Soochow University and MS degree in Math from Tamkang University. Currently he is a PHD Candidate in the department of CSIE at Tamkang University. His research interests include Software Engineering, Distributed Object Computing and Multimedia Software Engineering.



Ding-Jung Chiang received the Bachelor and Master degrees in department of CSIE form Tamkang University, Taipei, Taiwan, in 1995 and 1998. He is currently a PHD student in the department of EE at National Taiwan University. His currently research interests include multimedia computing, communication protocol and mobile Agent.