A Multimedia Educational Blackboard System ---- Toward a Richer Multimedia Applications

Chen-Ning Hsi
Department of Electronic Engineering &
Institute of Computer, Communication, and Control
National Taipei University of Technology,
Taipei, Taiwan 10643
cnhsi@en.ntut.edu.tw

Abstract

This paper describes a multimedia educational blackboard system MEBS. The MEBS system is based on a structural network document model. It embeds mechanism to enhance interaction capability and to enable dynamic temporal and spatial layout. It adapts blackboard convention to deliver courseware contents composed of local media objects and remote media objects. It consists of tools to support multimedia-based courseware development and delivery. Instructors use commonly available programs to create collections of educational materials and use MEBS tools to integrate and synchronize the course materials into courseware modules. The MEBS offers rich interactivity and compositional convenience with hierarchical structure. Directions toward richer multimedia applications based on MEBS experience will also be listed and discussed.

Keywords: *courseware, high-level authoring, interaction, multimedia system, synchronization.*

1 Introduction

The number of multimedia applications is growing at a tremendous rate as new types of media, such as audio and video, become technically ready and available on various computer platforms. Application domains such as telecommunications, education and entertainment, have all begun to realize the usefulness of these new forms of information in their own areas. Most early and current multimedia research has placed emphasis on the technology, i.e. on how to make it possible and more efficient to store, retrieve, transport, manipulate and present new media using computers, e.g. [6, 9, 16, 23, 25]. As this thread of research has gained great success in recent years, the cost of supporting and utilizing these new types of media on desktop computers has become reasonable and affordable. Together with rapidly improvement of computation power, especially in personal computers, the capability to handle and apply various new media has been delivered to a great number of users. The use of multimedia to support different aspects of our daily information needs therefore begins to be realized and gains more and more momentum and popularity. However, application-related multimedia researches are still in its early stage. Directions such as how to efficiently utilize the multimedia capabilities in various domains and how to evaluate their performance and suitability, and how to offer a higher level tools, i.e. in the design level, to guide and aid the authors on their development of the multimedia productions.

A central challenge of composing a multimedia document results from the dynamic characteristics of new types of media, such as video and audio, whose contents change across time. The term *continuous media* (or *dynamic media*) is commonly used to distinguish this property of time variance, and the term *discrete media* (or *static media*) is used for more conventional media, such as text and still images. Since timing becomes significant when a document consists of continuous media, in addition to conventional spatial relationships and interactive relationships among media objects, temporal relationships also need to be specified to create a meaningful and logically coherent presentation. The difficulty resides in the design nature of the composition task which requires the authors to explore a large space of different temporal, spatial, and interactive arrangements among media objects without losing focus in the design or violating practical limitations such as available network bandwidth and disk space.

Because of the difficulty of the composition task, a number of techniques have been developed to aid multimedia production. At the simplest level (i.e. the user interface level) graphical tools, such as timeline editors[7] and data flow editors[17], have been developed to improve ease of use. These tools simplify the authoring task by employing simple graphical notations. To better support authoring aids, a number of techniques to analyze a multimedia document have also been developed. For example, a Petri net formalism has been employed by Furuta and Stotts[27, 28] as an underlying multimedia document structure, and by Little and Ghafoor[19, 20] to model multimedia composition and construct a database schema, respectively. Analysis techniques developed using Pert net formalisms can then be applied to verify a number of important properties of the multimedia documents, including the maximum number of windows required for reading a document, reachability of certain media material, and the existence of independent or concurrent browsing paths. Another example of analysis is Buchanan's scheduling algorithm [3, 4, 5]. Through the use of linear programming, this algorithm can generate a schedule to fulfill the temporal constraints specified by the author in a graphical environment.

As a trend, the structure of multimedia documents has tended to grow more and more complex to improve their expressive power. However, they also become more and more difficult to understand and debug. To balance expressive power with the ease of use, compositional approaches to temporal specification has been developed [2, 11, 12]. Under the compositional approach, a number of composition operators are provided to create multimedia documents. Sequential and parallel are two common composition operators used to combine various media materials to play sequentially and in parallel, respectively. To support better authoring aids, the compositional framework developed by the authors [15] enforces a strict hierarchical constraint, i.e. temporal relationships across hierarchical boundaries are not allowed. This restriction can be limiting in some cases. However, this loss of power is not a problem for most documents in practice. On the other hand, the hierarchical structure provides many advantages. In particular, sophisticated new analysis and synthesis techniques become possible which allow very high-level authoring support [14].

In this paper, we will present a multimedia educational blackboard system --- MEBS. This system is based on a hierarchical framework to embed temporal and interactive as well as spatial relationships among media materials. A common layout, i.e. blackboard in a conventional classroom, is used to match the educational application domains. The hierarchical nature of MEBS eases the composition tasks by establishing a natural courseware design structure. In terms of temporal constraints, MEBS offers timeline notation (static nature) and event-trigger model (dynamic nature) to relate media presentations. This enables a broad-range of temporal composition. In terms of interactivity, MEBS provides standard VCR-like presentation control as well as user-paced content controls. Besides, complex interaction, such as time-limit response, and conditional presentation based on user behavior, are also possible. Courseware learning sessions can therefore be instructor-paced as well as student-paced.

The rest of this paper is organized as follows. Section 2 illustrates the system by a working example. Section 3 overviews the MEBS document model. Section 4 presents MEBS system architecture and Section 5 details the common interface between system central control unit and various media presenters. Finally, Sections 6 lists directions toward richer multimedia applications. Concluding remarks are offered in Section 7.

2 System in Action

Figure 1 illustrates MEBS in action. The topic of each course session is displayed on the window top. The main area under the topic is to present course content based on the interactive, spatial and temporal specifications defined in the document. On the left side of the window, there is a list of buttons. By clicking these buttons, students can move to other related topics. In the bottom of the window, a control panel provides basic VCR-like functions to play, pause/resume, abort, and restart this course session. The system status is shown to the right of this panel. Figure 1(a) shows an initial display for a courseware titled ' Graphs and Graph Algorithms' and Figure 1(b) shows the display after student moves to another topic named 'Graph Basics'.

To provide better browsing capability, a root button and a back button are available on top of the button list. They can be used to return to the home position of a courseware and to previous course session. A list button in the control panel enables the students to jump to certain position in time. These time positions can be defined statically in the document, or dynamically during the learning process. To summarize, the buttons on the left provide structural browsing control while the bottom control panel, including the list button, provide the basic playback control. Complex interactions, such as hyperlinks in the courseware content, time-limit response, and conditional display based on user behavior, are also possible. This type of complex interaction will be further discussed in next section.

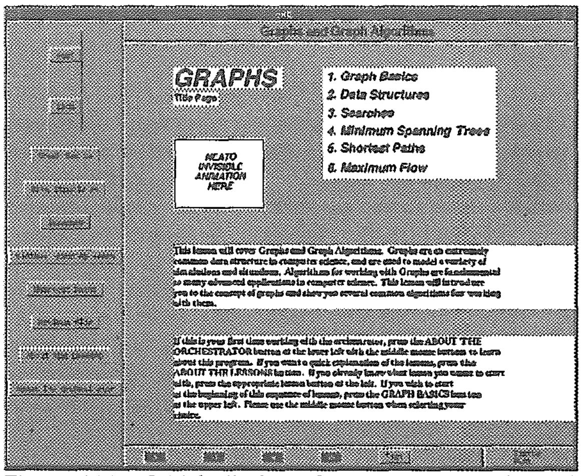

Figure 1(a) Initial display of courseware 'graphs and graph algorithm' under MEBS.

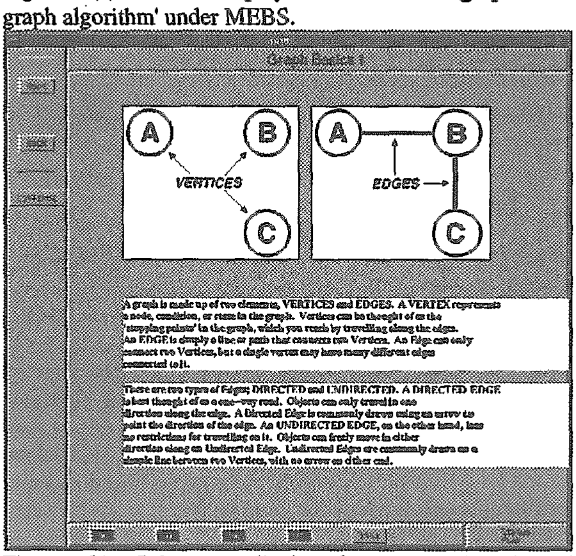

Figure 1(b) Subsequent display of courseware 'graphs and graph algorithm' under MEBS.

3 Document Model

MEBS documents or courseware possess a rigid organizational structure. The documents incorporate a standard hypertext data model based upon a directed graph of nodes and links called information network. The nodes of the network represent a logically related collection of information, the directed edges of the network indicate available transition routes to subsequent nodes. These arrangements are similar to HTML formalism. Unlike

HTML for web pages, however, a hierarchical structure is imposed on top of the information network. Each chunk of courseware is organized as a tree, with a node designated as the root. This tree construct is used to group semantically related pieces of information and can be viewed as a super node which in turn can be grouped in another tree structure. Children nodes of a specific node form subsequent nodes to visit naturally during browsing session and the boundary of a tree serves as a semantics separation. Figure 2 depicts the structure of a simplified document.

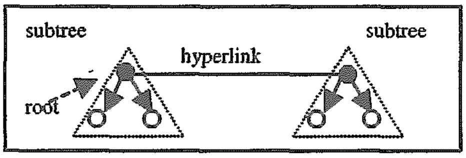

Figure 2 A simplified document structure

Appendix in the end of this paper lists the document formalism in the grammar form. Below is a sample document specification.

```
subtree sort_algorithm "Sorting" {
node sort_algorithm"Sorting Algorithms" {
timeline {audio intro_speech from 0 to 10
                        (intro_speech.au);}
hyperlinks{ save sort_algorithm2
                             "Heapsort"; }
}

node sort_algorithm2 "Sorting Algorithm2"
{
timeline { audio intro_speech from 0 to 10
                        (intro_speech.au); }
hyperlinks{save sort_algorithm3
                             "Heapsort"; }
}
root = sort_algorithms; }
```

Unlike a typical hypermedia document, not all of the information within a node is necessarily presented simultaneously. MEBS associates a synchronization object --- timeline object --- with each node to allow the author to relate various media presentations. A synchronization scenario such as "upon entering the node, a sound track will be played for 5 seconds, and then a image together with its text annotation will appear for 10 seconds" will become possible.

The timeline object only allows simple static, time-indexed score. To enable the association of temporally dynamic objects with fixed temporal objects, an event-triggered model is provided. By temporally dynamic objects, we refer to those objects with dynamical or undetermined temporal behavior. Two examples are program-driven animation and user interactions. Note here that we model user interactions as a media type. This will enable us to handle interaction with other media objects in an uniform manner. Based on the event-triggered model, each media object can be monitored for its internal event as wish. Hooked with this event, a list of media objects are recorded with various actions. When the awaiting event occurs, a trigger will be delivered to each of the object in the list with the corresponding actions. The internal events are not differential to avoid media-type related issues. However, event-trigger still model lots of interesting temporal relationship. The media actions triggered by media event are organized to form a uniform object interface. This object interface offers not only the trigger mechanism, but also serve as the base of presentation flow control.

There are various degree of interactions in the document model. This basic interaction is a VCR-like flow control. This interaction will keep the temporal composition among media content untouched. Temporal synchronization can be accomplished by rendering pre-fetch schedules with proper buffer management to fulfill inter-media and intra-media synchronizations [10, 18, 21, 22, 24]. The second type of interaction is in browsing control. From the presentation of a node, subsequent node can be visited by following links in the presentation content or links to the natural subsequent nodes. This interaction can be realized with the same approaches as the basic interaction. The third type of interaction is between interactors (or called gui, graphical user interface, widgets) and normal media objects. By interactors or gui widgets, we can elicit user input to trigger other media actions. This type of interaction is implemented by

1) treating interactors as a regular media type,
2) making event-trigger mechanism available to interactors, and
3) associating logic with the interactors' dispatch mechanism.

Through this arrangement, user interactions can be integrated into courseware and treated like normal media objects. Therefore, time-limit response (e.g. put a time limit upon user's reply on certain posted question) and conditional presentation based on user's interaction (e.g. display warning message if user gives wrong answer three times) become possible. The time-limit response can be implemented by scheduling the inquiring interactor(s) into time line like other media objects. The conditional presentation can be implemented by associating a routine with the responding interactor(s) to trigger the warning message (i.e. schedule the warning message object in the timeline) when the user gives the wrong answer three times.

Below is a sample document illustrating interesting interaction. In this one-node document, there are two text media objects (t1 and t2), one graphic object (g0), and two interactor objects (two buttons, butt0 and butt1, in this example). The t1 text object is scheduled to play at location (1, 1) as the course session begins, lasting for 1024 time units. The button butt0, plays at time 1 until time 100 (i.e. a time-limit interatction), and button butt1 plays at time 4 without any time limit. Both buttons have a procedure associated with them. When the butt0 button is clicked the first time, it will schedule another text object, t2, to play immediately. When butt0 is clicked the second time, it will schedule this t2 text object to remove from the display. The same behavior will repeat for consequent clicks. When button butt1 is clicked, it will schedule itself to disappear for 25 time units and redisplay in a different position.

Through this example, we can see how interaction can be integrated into a document by delicately treating interactors as a regular media type. This arrangement makes interaction possible to schedule like other media objects. By associating procedures with interactors, and making temporal control and spatial control available in a procedure, interactive, temporal , and spatial relationship can interact with each other.

```
component buttonex "Buttons" {
node sampl "Button Examples" {
declare {
integer i := 0; integer j := 0;
integer k := 100;
ltext t1 init{"text0.txt"};
ltext t2 init{"text1.txt"};
lgraphics g0 init{"algo1.ppm"};
lbutton butt0 init{"Display Text"}
proc {
if (usersignal1 = 1)  i := i + 1;
if (i = 1)
       schedule t2 from 1 play at 400 200 ;
if (i = 2) {
       schedule t2 from 1 stop  ;
       i := 0;
}
};
lbutton butt1 init{"Catch Me"}
proc {
if (usersignal1 = 1){// button pressed
       schedule butt3 from 1 stop;
       k := k + 30;
       schedule butt3 from 25 play
              at k 150 ;
}
};
}
timeline {
t1 from 0 until 2048 play at 1 1;
butt0 from 1 until 100 play at 10 200 ;
butt1 from 4 play at 10 400 ;
}
}
  root := sampl;
}
```

Synchronization issues are more complex since the media composition (including the media content as well as media temporal relationship). Most of the time, we need to perform pessimistic analysis for the worst case scenario to satisfy the synchronization constraints [13, 14].

4 System Architecture

The MEBS has three major components: orchestrator, presenters, and interactors. The orchestrator serves as the centralized controlling object. It maintains the internal run-time representation of the network, including the layout and the state of the current node's timeline. By making public its services, orchestrator can also be controlled by the external process. The synchronization and presentation flow control of the courseware delivery are also handled by the orchestrator using an internal clock grained based on the hosted hardware. Interaction control and event trigger model are provided by the orchetrator. These are all accomplished by a standard inter process communication interface (IPC).

Media content is rendered by presenters. There are two types of presenters within the system: remote and local. Both types of presenters render media content on the client machine through orchetrator control via IPC. The orchetrator controls remote presenters (separate processes) via a 2-way interprocess communication mechanism. These presenters encapsulate the individual media-specific code and data. Each presenter provides a common set of services to orchestrator (refer to Section 4). Specialized presenters also support a set of extended messages which are sent to the orchestrator.

Local presenters include the interface elements, e.g. the graphical user interface widgets/interactors (defined later) as well as locally resident media presenters. They include buttons, dialog objects, text presenter, image presenter, etc. Different from remote presenters, the local presenters are tightly integrated to the orchestrator (in the same process space as orchestrator). The following figure illustrates the MEBS system architecture.

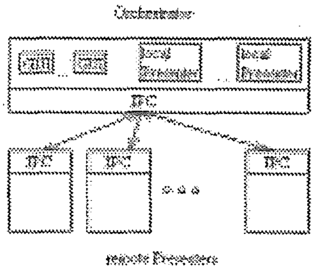

Figure 3 MEBS system architecture

To handle user interactions uniformly, we treat them a media type: interactor. The presentation of an interactor is the "display" (graphical, audio, or any user sensitive presentation) as the stimuli to user interactions. This "display" will end when the expected interaction has been received. The synchronization of user interaction with other media materials has two levels. The first level is the synchronization of interaction stimuli, i.e. the "display" of the interactor. Example for this level of synchronization is to post a dialog box at certain time for a specified duration. The second level of the synchronization of the interaction itself. Example for this level of synchronization is to specify an expired time for the received user input.

The current presenters support raster images, postscript documents, plain text, digitalized audio, UNIX executables, hypertext documents, digitalized video, and controllable Polka animations [26]. When a new media type is desired, a new presenter can be devised which complies the uniform IPC interface. A number of presenter templates have been developed to aid the system extension.

5 Uniform IPC Interface

The IPC mechanism is created as a class library. An instance of this class is created on both orchestrator and presenter side during run time. This allows the orchestrator and presenters to communicate. Mechanisms for intermedia and intramedia synchronization are built in both ends of each connection. To unify the IPC interface, every possible presentation functions are defined and made available. They include: load, play, stop, pause/resume, map, unmap, getindex, setindex, getval, sendval, status, kill, and generic_coded_function. For each type of media, some functions may have slightly different meanings. Refer to [Gall93] for the detailed media-related function semantics.

As we can see, the uniform IPC interface glues various system components. Under this arrangement, presenters can reside on networked machines to balance to load. To extend from a single user on a single workstation session to a controlled courseware broadcast will be possible by additional broadcasting control center which control individual orchestrator. This is possible based on the fact that the orchestrator itself are designed and implemented to comply the uniform IPC interface and posts and make available its service to any external process. In other words, orchestrator can be seen as a presenter itself which in turn can be controlled by another orchestrator. Figure 4 depicts this situation.

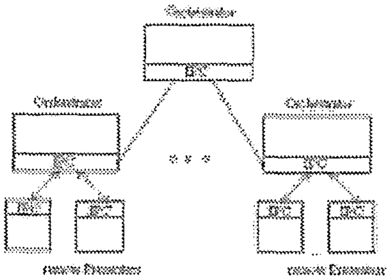

Figure 4 Orchestrators function as remote presenters

6 Toward Richer Multimedia Applications

A multimedia blackboard delivery system, MEBS, has been described. It has a number of advantages:

1) hierarchical structure on top of information network base,
2) capability of specifying static and dynamic temporal relationship among various media objects,
3) uniform manner to handle user interaction, including VCR-like presentation control, as well as interaction which alters presentation content and composition, and
4) capability of synchronizing user interaction with different media types.

This research is aimed at the education domains. Along the research and system development process, we emphasize more on the application-oriented considerations than those on the technology side. Examples of applications-oriented considerations are degree of interactivity of the multimedia presentation and suitability of the underlying document framework. This application-driven (instead of technology-driven) approach shed light on our research directions toward richer multimedia applications. As described earlier, multimedia technology make all multimedia applications possible. It forms the base of all multimedia productions in a cost-efficient manner. However, the benefits of the technology advance will not be realized without any application domains. Established technology without considering the application demands will fall short when adapted to solve real world problems. In this section, some application-oriented considerations are discussed in the hope to offer directions to multimedia researches to improve the multimedia usability as well as broaden multimedia application domains.

The considerations are based on the relationships among multimedia objects. As the multimedia composition contains various media materials, the relationships existing among those media materials are the key to turn/string those materials into a meaningful unit. In other words, relationships among multimedia materials in a production should be considered as a critical constituent to its expressiveness and semantics. There are three possible relationships to relate different media objects: spatial, temporal, and interactive relationship. The spatial relationship relates media contents based on the media geometry. By placing various media contents in a 2D or 3D space with proper sizing, multiple media materials can be laid out to convey certain meaning. The spatial arrangement usually plays a crucial role in its expressiveness. Common layout examples include side-by-side, or top-to-down, and four-squared comics. These layouts can be used to signify the associations/interactions among media materials: equally important display (side-by-side), causal-effect (top-down), and story-sequence (four-squared comics) correspondingly.

The temporal relationship imposes timing constraints among media materials to form consistent and meaningful presentation. There are 13 possible temporal relationships, described in [ALLE83], between two media objects. It is important to maintain the timing relationships in a multimedia presentation to deliver its meaning precisely and efficiently. Common temporal arrangements such as parallel and sequential presentation are usually used to signify cross-referencing, annotating, or cause-effect associations among media materials. Failure in maintaining the temporal relationship in a presentation might result in unpleasant experience, e.g. unsynchronized audio and video presentation, and unsuccessful communication/expression, e.g. lengthy delay in a cause-effect demonstration.

The interactive relationships planned in a multimedia production are targeted to improve the degree of user engagement. Lack of interaction will not only downsize a multimedia presentation into fancy slide show, but also restrict its usage. Educational applications require extensive user interactions to have the students involved as well as to evaluate the student's performance to direct the subsequent presentation. Simple VCR-like flow control will not be sufficient.

As a result, relationships among various media materials are critical to its expressiveness and usability in a multimedia production. Unnecessary limitations on those relationships will affect the effectiveness in delivering a multimedia presentation as well as impair its expressiveness. Based on these observations, it will be important to support flexible spatial, temporal and interactive relationships in order to realize richer multimedia applications. Therefore, a list of derived multimedia research directions to accelerate this cycle of information evolution might worth attention. They are:

1) richer interactions: more than a VCR-like flow control. This will complicate the synchronization/scheduling problem. Due to the uncertainty nature of the interaction, pessimistic analysis of system resources and dynamical scheduling might be necessary. It will also complicate the document structure if the interaction features are built to its document formalism.

2) flexible temporal combination: more than a simple fixed timeline scheduling. This will incorporate media with dynamic temporal nature, such as program-driven animation, in the multimedia production, as well as make various temporal arrangements available to deliver richer semantics, such as annotate a program-driven animation with an audio clip. This will complicate the synchronization problem as well as document structure.

3) high-level authoring tools: design-level other than tool-level. As the flexibility of multimedia productions grows, the complexity to author them will increase. Authoring tools need to be improved for their capabilities to cope the increasing difficulties. Design-level tools will be the key. It will be desired for tools to aid the authors in determining proper spatial layout, temporal arrangement and interactive planning, as well as help them to explore the huge possible design space to shape/converge their ideal designs.

4) media application guidelines and evaluation rules: guidance instead of trial-and-error based. We are still young and inexperienced in applying different media materials to express certain meanings. The guidelines and evaluation rules on using each type of media or composition of these media types effectively and efficiently are still missing. Principles like how to choose a proper color, fonts, spatial layout, or even an appropriate volume for a audio clip, window size, etc. during multimedia production are all on demand.

7 Conclusion

In this paper, a multimedia educational delivery system, MEBS, has been described. It has four advantages: the hierarchical structure, the capability of specifying static and dynamic temporal relationships, the uniform manner to handle user interaction, and the capability of synchronizing user interaction with various media objects. The document model, the system architecture, and its IPC interface are described accordingly. Together, they form a system to deliver multimedia courseware. Currently, there are different courseware modules available for topics like data structure/algorithm and clean room. The future direction of this research will be on improvement the authoring environment and combination of a tutoring model with MEBS. Based on the experience of this research, a numbers of application-driven considerations are also discussed. It is the author's perspective that multimedia technology advances will be on the practical track and useful when based on practical application considerations.

Appendix

The document formalism used in MEBS is listed in the grammar form.

```
network  ::=  subtree_spec

substree ::=  T_SUBTREE subtree_id display_name
              T_LEFTBRACE opt_subtree_spec
              node_specs root T_RIGHTBRACE

opt_subtree_spec  ::=  lambda | subtree_spec

subtree_id        ::=  ident

node_specs        ::=  lambda |
                       node_specs node_spec

node_spec ::= T_NODE node_id display_name
              T_LEFTBRACE timeline
              opt_events links T_RIGHTBRACE

node_id           ::=  ident

timeline ::= T_TIMELINE T_LEFTBRACE
             event_list T_RIGHTBRACE

opt_events ::= lambda |
               T_EVENT T_LEFTBRACE
               dyn_events T_RIGHTBRACE

dyn_events        ::=  lambda |
                       dyn_events dyn_event

dyn_event ::= event_type event_name T_WHEN message
              opt_duration T_LEFTPAREN filename
              T_RIGHTPAREN T_SEMICOLON

message ::= T_MESSAGE1 message |
            T_MESSAGE2 message |
            T_MESSAGE3 message |
            T_MESSAGE4 message |
            T_MESSAGE5 message

opt_duration ::= lambda | T_DURATION integer

links ::= T_HYPERLINKS T_LEFTBRACE
          link_list T_RIGHTBRACE

root ::= T_ROOT T_EQUALS root_node
         T_SEMICOLON

root_node ::= node_id

event_list::= lambda | event_list event

event  ::=  event_type event_name T_FROM
            start_time ope_end_time opt_loc
            T_LEFTPAREN filename
            T_RIGHTPAREN T_SEMICOLON

start_time        ::=  integer

opt_end_time      ::=  lambda | T_TO integer

opt_loc           ::=  lambda |
                       T_AT integer integer

event_name        ::=  ident

filename :        :=   ident

event_type        ::=  T_AUDIO | T_VIDEO |
                       T_GRAPHICS | T_TEXT|
                       T_PS | T_ANIM
```

link_list	::=	lambda | link_list link
link	::=	link_type link_dest display_name opt_index T_SEMICOLON
link_type ::=	lambda | T_SAVE | T_NOSAVE	
link_dest ::=	node_id	
opt_index	::=	lambda | T_LEFTPAREN start_index T_RIGHTPAREN
start_index	::=	integer
ident	::=	T_IDENTIFIER
display_name	::=	string
string	::=	T_STRING
integer	::=	T_INTEGER
lambda	::=	/* empty */

This work was funded by National Science Council under grand NSC 88-2213-E-027-009.

References

[1] J.F. Allen, "Maintaining Knowledge about Temporal Intervals", Comm. ACM, v.26, no.11, pp. 823-843, 1983.

[2] D.C.A. Bulterman, G.V. Rossum, V.R. Guido, and R.V. Liere, "A Structure for Transportable, Dynamic Multimedia Documents", USENIX, 1991.

[3] M.C. Buchanan, and P.Zellweger, "Specifying Temporal Behavior in Hypermedia Documents", Proc. of the European Conference on Hypertext 1992, Milan, Italy, December 1992.

[4] M.C. Buchanan, and P. Zellweger, "Scheduling Multimedia Documents Using Temporal Constraints", Proc. of the Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, CA., November 1992.

[5] M.C. Buchanan, and P.T. Zellweger, "Automatic Temporal Layout Mechanisms", Proceedings of the ACM MULTIMEDIA 93, pp.341-350, August 1993.

[6] M.S. Chen, D.D. Kandlur, and P.S. Yu, "Storage and Retrieval Methods to support fully interactive playout in a disk-array-based video server", ACM Multimedia Systems, v.3, no.3, pp.126-135, July 1995.

[7] G.D. Drapeau, and H. Greenfield, "MAEstro --- A Distributed Multimedia Authoring Environment", Proceedings of Summer Usenix Conference, 1991.

[8] E. Gallant, and G. Li, , "Multimedia Educational Delivery System (MMEDS)", User Documentation, Georgia Institute of Technology, October 6, 1993.

[9] C.F. Goldfarb, "HyTime: A Standard for Structured Hypermedia Interchange", IEEE Computer, pp. 81-84, August 1991.

[10] S. Gollapudi, and A. Zhang, "Buffer Model and Management in Distributed Multimedia Presentation Systems", ACM Multmiedia System, v.6, no.3, pp. 206-218, 1998.

[11] R. Hamakawa and J. Rekimoto, "Object Composition and Playback Model for Handling Multimedia Data", Proceedings of the 1st ACM International Conference on Multimedia, pp. 273-381, August 1993.

[12] L. Hardman, G.V. Rossum, and D.C.A. Bulterman, "Structured Multimedia Authoring", Proceedings of the 1st ACM International Conference on Multimedia, pp. 283-289, August 1993.

[13] C.N. Hsi, Ph.D. dissertation, Computer Science Department, University of Arizona, June 1994.

[14] S.E. Hudson, and C.N. Hsi, "Algorithms to Detect and Exhibit Resource Conflicts in Multimedia Documents", Proceedings of the 10th Workshop on Combinatorial Mathematics and Computation Theory, December 9, 1994.

[15] S.E. Hudson, and C.N. Hsi, "A Framework for Low Level Analysis and Synthesis to Support High Level Authoring of Multimedia Documents", IEEE Multimedia'94 Workshop, Kyoto, May 1994.

[16] K. Keeton, and R.H. Katz, "Evaluating Video Layout Strategies for a High-Performance Storage Server", ACM Multimedia Systems, v.3, no.2, pp.43-52, May 1995.

[17] J.F. Koegel, J.L. Rutledge, and J. Heines, "Visual Programming Abstractions For Interactive Multimedia Presentation Authoring", IEEE Proceedings of Visual Language, pp. 231-233, 1992.

[18] W. Liao, and O.K. Li, "Synchronization of distributed multimedia systems with user interactions", ACM Multmiedia System, v.6, no.3, pp. 198-205, 1998.

[19] T.D.C. Little, and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects", IEEE Journal on Selected Areas in Communications, v.9, no.3, pp. 413-427, April 1990.

[20] T.D.C. Little, and A. Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks", IEEE Computer, pp. 43-50, October 1991.

[21] T.D.C. Little, and A. Ghafoor, "Interval-based conceptual models for time-dependent multimedia

data", IEEE Transaction Knowledge Data Engineering, v. 5, no. 4, pp. 551-563, October 1991.

[22] B. Prabhakaran and S.V. Raghavan, "Synchronization Models for Multimedia Presentation with User Participation", ACM Multimedia System, v.2, no. 2, pp.53-62, August 1993.

[23] P.V. Rangan, S. Ramanathan, and T. Kaeppner "Performance of Intermedia Synchronization in Distributed and Heterogeneous Multimedia Systems", Computer Networks ISDN System, v.27, pp.549-565, 1995.

[24] C. Shahabi, and S. Ghandeharizadeh, "Continuous Display of Presentations Sharing Clips", ACM Multimedia Systems, v.3, no.2, pp.76-90, May 1995.

[25] C. Shahabi, and S. Ghandeharizadeh, "Continuous Display of Presentations Sharing Clips", ACM Multimedia Systems, v.3, no.2, pp.76-90, May 1995.

[26] J.T. Stasko, "Using Direct Manipulation to Build Algorithm Animations by Demonstration", Proceedijngs of the ACM SIGCHI'91, New Orleans, pp.307-314, May 1991.

[27] P.D. Stotts, and R. Furuta, "Temporal Hyperprogramming", Journal Visual Language and Computing, v.1, no.3, pp. 237-253, April, 1990.

[28] P.D. Stotts, and R. Furuta, "Dynamic Adaptation of Hypertext Structure", Hypertext'91 Proceedings, pp. 219-231, December 1991.