Message Passing in XML-Based Language for Creating Multimedia Presentations

Stanisław Polak[1], Renata Słota[1], and Jacek Kitowski[1,2]

[1] Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Cracow, Poland,
phone: (+48 12) 6173 964, fax: (+48 12) 6339 406,
{polak,rena,kito}@uci.agh.edu.pl, http://www.icsr.agh.edu.pl/
[2] ACC CYFRONET, ul. Nawojki 11, 30-950 Cracow, Poland

Abstract. The paper describes a new language and supporting tools for creation and display of multimedia presentations, specially for the Internet usage. The language implements the message passing paradigm to set up communication between elements of presentations. Model of the language and its flexibility are discussed.

1 Introduction

There are a few languages for creation Internet-enabled multimedia presentations. Simple presentations are implemented in HTML [1]. Tags like `<img>`, `<embed>` or `<bgsound>` are used for inclusion of multimedia data (e.g. graphics, movies, sounds) to the WWW pages. Only some WWW browsers recognize these tags correctly, furthermore the processes of presenting those data cannot be synchronized.

Complicated multimedia presentations can be built using Java [2] and Java Media Framework (JMF) [3]. Processes of presenting media data can be synchronized but knowledge of Java programming language is required.

Another possibility is to build a presentation with Synchronized Multimedia Integrated Language (SMIL) [4]. This language is based on eXtensible Markup Language (XML) [5]. SMIL enables one to create complicated multimedia presentations, including audio or video sequences, animations, graphics and texts. Media elements can be placed in any position on the screen, displayed according to user's computer capabilities and coordinated in three ways: sequential, parallel and exclusive. The sequence of presentation is determined by tags, therefore modification of the sequence requires changes in the SMIL document. In SMIL, semantically-related XML elements, attributes, and attribute values are grouped in modules. In the reported approach we propose a different method of synchronization of element presentation to have more flexibility in controlling the sequence of the presentation. The method of synchronization is based on events and messages generated on-line according to user's needs. Instead of modules we propose libraries which extend language capabilities without any additional modifications of the interpreter code.

In this paper first we shortly describe the MUltimedia Lecture description Language (MULL) [6,7] and two tools related to the language. Next, we discuss

B. Monien and R. Feldmann (Eds.): Euro-Par 2002, LNCS 2400, pp. 826–829.

the flexibility of the language that results from the message passing paradigm used for media synchronization as well as a few libraries for extending scope of the language. Conclusions summarize the paper.

2 Multimedia Lecture Description Language and MULLtools System

The set of MULL elements consists of objects, messages, buffers, events and libraries. The object is an abstract representation of reality. 'Audio', 'Video', 'Picture', 'Answer', 'Timer', 'Text', and 'Page' objects are defined. Their properties describe various object features, for example the name and the position on the screen. The object contains buffers and it can execute a predefined basic operation (e.g., "show", "hide", "start", "stop" and "display"). The objects communicate to others through Message Manager by message passing. The messages are placed in buffers, which are associated with events. Every object can send any kind of the message, while it can interpret particular messages only. There are four message types: operational, setting up, information and library. The set of operations for the objects and the set of messages can be extended by the libraries *olibraries* and *mlibraries* respectively. MULL is implemented in XML. Objects and libraries are represented by tags. Buffers and properties establish a set of tags attributes. The message is a value of the attribute representing the buffer.

Two programs – Creator and Browser, called MULLtools – are associated with the MULL. The Creator is used for design of lecture appearance, which is defined by the resulting MULL file. For example, the content of the file describes presentation modes of the individual elements, connection between the elements, etc. Since the MULL file is a plain text file, a similar result could be obtained using any text editor for the price of more programming effort and knowledge of the MULL language. To show the lecture, the Browser is executed. The MULLtools system is written in Java 1.1.

3 Flexibility Obtained from the Message Passing

Capabilities of the MULL language depend on the messages because the messages define the sequence of the presentation. Message passing is used to set up communication between parallel events, i.e., to synchronize, to initiate and to activate them. Flexibility obtained from the message passing is correlated with different kinds of messages but especially with: a) possibility of changing messages in buffers during presentation and b) possibility of extending a set of messages by specialized libraries. These two possibilities are shortly described below.

The messages are placed in buffers. Every buffer has an attribute equivalent in the MULL document. Changing values of the attributes results in automatic change of the content of the buffers, that influences the sequence of the presentation. So the sequence can be determined by the author of the presentation

in the MULL document. But if the buffers contain the setting up messages, the sequence of the presentation can be modified on-line, according to various circumstances, e.g, student's actions.

The MULL contains a basic set of messages, which can be extended with the libraries. Three examples of the libraries are described below – for sending a few messages at the same time, for exchanging messages among different MULL presentations and for creating simple conditions respectively.

The "Complex Message" Library. A few basic messages are placed in one buffer, therefore, the "complex message" is regarded as a single message. The *mlibrary* recognizes this kind of message and extracts the basic messages to be send in the order of their appearance in the "complex message". This library is useful if one has defined a sequence of events in advance.

The "Communication Message" Library. A single WWW page can contain several named applets, which communicate to others. Because the MULL Browser is an applet, so it can send messages to other MULL Browsers placed on the same WWW page thanks to the "communication message" library. Therefore two or more presentations could control each other.

This library provides direct and indirect modes of message sending. In the direct mode the messages sent by the objects placed at the sender side (i.e. by the first Browser instance) are received by the objects localized both at the sender and at the receiver sides (by the second Browser instance). In the indirect mode, the object localized at the sender side can send only selected messages to the objects placed at the receiver side. The message transmitted to the receiver is encapsulated in the "communication message".

The "Condition Message" Library. The purpose of the library is to create a simple conditional "instructions". Two types of the "condition message" are introduced: ordinary and special. Using the special "condition message", modifications of values of the variables previously defined are permitted. The value of these messages is tested by the ordinary "condition message", and according to the result, a corresponding message is sent.

One example of the library usage is to display a picture under condition that a process of reproducing a movie has been finished and at the same time a mouse cursor has been placed within a specified active area of an another picture.

4 Conclusions

The result of our work is the MULL language and the tools for composing and for presenting multimedia lectures. The tools consist of the MULL Creator and the MULL Browser. The language is useful for displaying text and pictures during playing audio/video file, building simple multiple-choice tests, synchronous starting, stopping and playing audio/video files, synchronous displaying pictures,

creating simple animations, building a course which depends on user preferences, actions, and state-of-the-art, etc.

The message passing is responsible for controlling the sequence of presentation and the appearance of its individual elements. The messages are placed in buffers and the content of them can change according to current circumstances, like user's actions. If the existing capabilities are not sufficient, the user can take advantage of the libraries. Own libraries can be developed as well. Capabilities of the Browser, the Creator and the language automatically follow the developed libraries – every library extends the set of available messages.

The Browser has been designed to share message libraries with other language interpreters, however these languages must be based on the presented paradigm. For example, we can create a new language similar to X3D [8] for which we can use the same libraries without any recompilation of the new language interpreter.

Acknowledgments

The work has been supported by Scientific and Technological Cooperation Joint Project between Poland and Austria: KBN-OEADD grant No. 3633/R01/R02 and by AGH grant.

References

1. World Wide Web Consortium: HyperText Markup Language. (2002) `http://www.w3.org/MarkUp/`
2. Sun Microsystems: The Java Tutorial. (2002) `http://java.sun.com/docs/books/tutorial/`
3. deCarmo, L.: Unlocking the secrets of the Java Media Framework. Java Developer's Journal, Feb.-Apr. (2001) `http://www.sys-con.com/java/`
4. World Wide Web Consortium: Synchronized Multimedia Integration Language (SMIL 2.0) Specification – W3C Recommendation. (2001) `http://www.w3.org/TR/smil20/`
5. Cagle, C., Gibbons, D., Hunter, D., Ozu, N., Pinnock, J.: Beginning XML. Amazon (2000).
6. Polak, S., Słota, R., Kitowski, J.: Creation and Presentation of Multimedia Courses on the Internet. In Domingo, M.E., Cebollada, J.C.G., Salvador, C.P., eds.: Proceedings of Six Annual Scientific Conference on Web Technology, New Media, Communications and Telematics Theory, Methods, Tools and Applications, SCS, Delft (The Netherlands) (2001) 222–226.
7. Polak, S., Słota, R., Kitowski, J., Otfinowski, J.: Xml-based tools for multimedia course prepararion. Archiwum Informatyki Teoretycznej i Stosowanej, Vol. 13, No. 1 (2001) 3–21.
8. The Web3D Consortium: X3D Specification. (2001) `http://www.web3d.org/TaskGroups/x3d/specification/`