

## DESIGN AND IMPLEMENTATION OF THE SMIL (SYNCHRONIZED MULTIMEDIA INTEGRATION LANGUAGE) PLAYER

Dongkyoo Shin and Dongil Shin  
Department of Computer Engineering, Sejong University, Seoul, Korea

### ABSTRACT

The Synchronized Multimedia Integration Language (SMIL) is recommended by the World Wide Web Consortium (W3C) because it offers an integrating format representing how various media objects are positioned spatially and combined temporally using a collection of XML elements and attributes for multimedia delivery over Web. This paper presents the design and implementation of a JAVA-based SMIL player, which processes different types of media objects using multiple threads. Its cache engine detects the media type and allocates the proper cache memory for the corresponding media object.

### 1. INTRODUCTION

With the full development of Web technologies, providers of synchronized multimedia contents such as television programs and online courses are very interested in using Web because they have easily changeable and up-to-date contents, and Web offers cheaper costs of distribution, and no need for a new end-user application. However, current HTML-based Web technology is clearly limited when it comes to creating continuous multimedia presentations. It lacks a format for authoring synchronized multimedia content, and support for network transmission of this type of content [1]. The Synchronized Multimedia Integration Language (SMIL) is recommended by the World Wide Web Consortium (W3C) and allows creation of time-based multimedia delivery over Web [2]. Based on XML, it allows developers to mix many types of media such as text, video, graphics, audio, and vector-based animation together, to synchronize them in a timeline, and to make authoring of TV-like multimedia presentations on the Web easier. Besides SMIL is adopted in enhanced TV applications in which broadcast and Web content are coordinated and in wireless devices for supporting lightweight multimedia functionality [3,4]. To facilitate this upcoming technology, a well built SMIL player is

necessary along with a well built SMIL authoring tool [5]. Currently several SMIL players are available around the world [6]: Grins, HPAS, S2M2, QuickTime, Internet Explorer and RealPlayer G2.

### 2. FEATURES OF SMIL

SMIL is an XML-based declarative language to present Web-based multimedia documents. It offers an integrating format representing how various media objects are coordinated and positioned spatially and combined temporally using a collection of XML elements and attributes [6,7]. A SMIL document consists of two containers <head> and <body> which can nest other containers. The head container defines spatial layout and can nest containers such as <layout>, <region> and <meta>, and the body container defines spatial and temporal relations of media objects and linkage and can nest containers such as <par>, <seq>, <img>, <video> and <audio>. SMIL describes four fundamental aspects of multimedia presentations. It provides layout specification primitives to support simple and easy use of document layouts. It also provides temporal specification primitives to encode the temporal structure of the application and the refinement of the start and end times of media events. Alternative behavior specification primitives are provided to express the various optional encodings within a document based on system or user requirements. Finally, hypermedia support primitives are used to link various media objects. In a SMIL document, the linking problem is much more complicated than normal HTML.

SMIL was first released in mid-1998 and has been installed on desktops worldwide because it is adopted in RealPlayer G2, Quicktime and Internet Explorer. A significant update was made with SMIL 2.0 in August 2001 [3,4]. SMIL 2.0 defines 10 major function groupings of elements and attributes: animation, content control, layout, linking, integration of media objects, meta-information, structure, timing and synchronization, time manipulations, and transition effects [6]. Of these,

the timing and synchronization grouping is the core part of the SMIL specification. The SMIL profiles are provided to combine relevant groups of modules for player manufacturing. The SMIL 2.0 profiles include SMIL 2.0 language, SMIL basic, SMIL and XHTML+SMIL. The SMIL 2.0 language profile is adopted in players and tools including Oratrix and RealNetworks. The Third Generation Partnership Project (3GPP) has used the SMIL basic profile as the basis for wireless multimedia devices [6].

SMIL is combined with MPEG-4 in the development of Extensible MPEG-4 Textual (XMT) format specification, in which MPEG-4 is used for low-level object encoding and SMIL is used for high level XML-based object composition [4,8]. SMIL is also used in applications such as electronic commerce and distance education [9,10].

**3. SSP: SEJONG SMIL PLAYER**

We design and implement a SMIL player, the Sejong SMIL Player (SSP), which is a JAVA application utilizing the JDK(Java Development Kit) and the JMF (Java Media Framework) [11] APIs to support various media types. The SSP is comprised of three modules: SMIL main, parsing and rendering. Figure 1. shows the overall structure of the SSP.

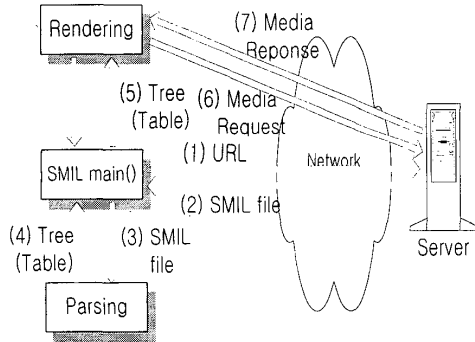


Figure 1. Overall Structure of the SSP

SMIL main module controls and manages use interface, and accesses a remote server designated by URL which retrieves a SMIL file and sends it to parsing module. It also receives parsed data from the parsing module and transfers it to the rendering module and controls various components used in the interface of the player. Parsing module uses SAXDOM (Simple API for XML DOM) to modularize the SMIL DTD (Document Type Definition) parsing [12]. SAXDOM basically separates the underlying parsing functions into a set of callback routines from the SAXDOM driver. In the parsing stage, each element and its attributes in the SMIL file are stored in a hash table and saved in a node in the

tree structure, as shown in Figure 2. That is, parsing module generates a set of hash tables about spatial information while parsing <head> parts of the SMIL document. Parsing <body> parts generates the corresponding media type information and its timing synchronization in a set of hash tables. In implementation, a vector is used to store tree node position and its corresponding hash table. Rendering module defines all SMIL media objects through MIME type, since SMIL supports various types of media object such as audio, video, and text streams. Media objects are synchronized in a time line and multi-threading is used for spatial synchronization. JMF class library is used to render synchronized media objects using the stored media information.

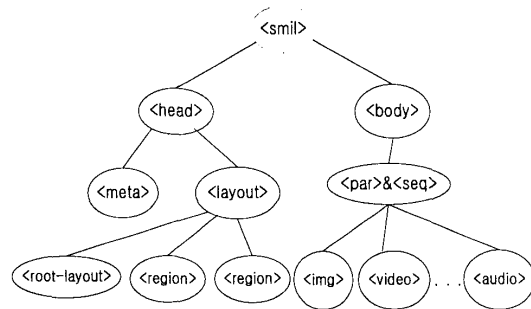


Figure 2. Tree Structure after Parsing

Figure 3 shows interface of the SSP,

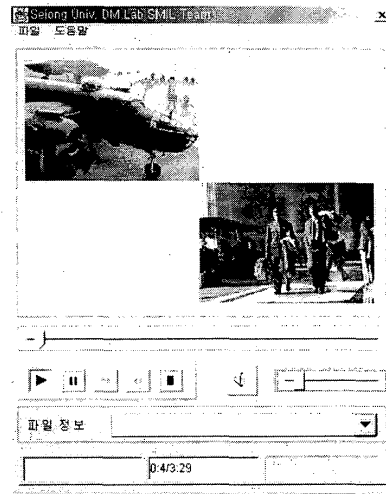


Figure 3. Interface of the SSP.

**4. CONSIDERATION FOR PERFORMANCE**

The syntax of SMIL is relatively easy to understand. However, it can employ many different types of media such as text, graphic, audio and video. To handle different types of media objects at the same time, the SSP is designed to be multi-threaded. When parsing each part of a SMIL documents, a new thread is created to handle it. At the end of parsing the resulting objects are stored in hash tables on the nodes of the tree structure. When execution of the SMIL document starts, corresponding threads are concurrently running to render and show the different media objects. Figure 4 shows the multi-threaded processing of head and body containers in which nested containers and media objects are also multi-threaded according to temporal and spatial synchronization in the stored data.

Compared with traditional textual objects, multimedia objects usually require much higher bandwidth and real-time transfer. Moreover, multimedia stream is usually bursty. Just increasing the bandwidth will not solve the burstiness problem. To achieve smooth playback of the presentation, the player must be able to prefetch and buffer media objects and dynamically utilize different versions of media objects.

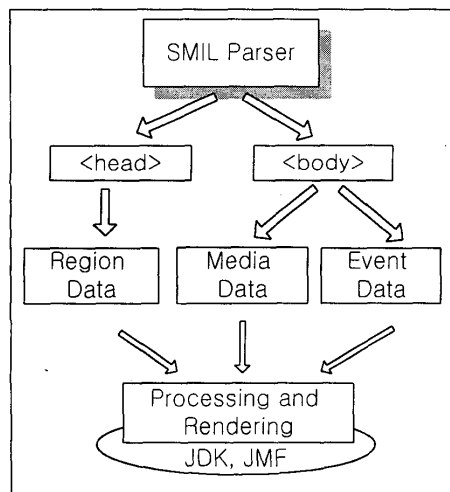


Figure 4. Internal Structure of the SSP.

The SSP incorporates an intelligent cache engine which detects the media type and allocates the proper cache memory for the corresponding media object. While caching research traditionally focused on replacement policies for optimal storage allocation [13], service differentiation technique is applied in the cache engine which utilizes biased replacement policies [14]. The basic idea is that if more storage space is assigned to the selected media class based on its importance in the playback representation, it will enhance the overall presentation. If the future access pattern is known, the amount of space that needs to be allocated to each media

class can be decided ahead of time to achieve optimal presentation quality. Importance of media classes can be decided by the user or application. In the meantime, the cache engine stores the played media objects into a JAR (Java Archive) file for future uses. Currently all cached media data is stored into several JAR files of a single client machine on which the SSP runs. Each JAR file keeps track of single media type. For instance, if the SSP plays SMIL documents including 3 videos, 2 audio clips, and 5 text streams, the client machine keeps 3 different JAR files.

## 5. CONCLUSION

A JAVA application-based SMIL player that employs a caching engine is presented. Current version of the SSP supports a SMIL 2.0 language profile. XHTML+SMIL and SVG(Scalable Vector Graphic) should be integrated in the future. The SSP also utilizes HTTP for transferring SMIL documents. However, the next version of it should support RTP(Real-time Transfer Protocol) [15] and RTSP(Real-time Streaming Protocol) [16]. In future research, a more attractive user interface and more robust cache design should be incorporated into the SSP. Research on more effective caching methods that work with the SSP should follow.

## REFERENCES

- [1] F. Rousseau and A. Duda, "Synchronized Multimedia for the WWW", Computer Networks and ISDN Systems, Vol 30, No. 11-7, April 1998, pp.417-429.
- [2] P. Hoschka, "An Introduction to the Synchronized Multimedia Integration Language", IEEE Multimedia, Vol. 5, Oct.-Dec. 1998, pp. 84-88.
- [3] D. Buttermann, "SMIL 2.0 Part1: Overview, Concepts, and structure", IEEE Multimedia, Vol 8, Oct.-Dec. 2001, pp. 82-88.
- [4] D. Buttermann, "SMIL 2.0 Part2: Examples and Comparisons", IEEE Multimedia, Vol 9, Jan.-March 2002, pp. 74-84.
- [5] D. Bulterman *et al.*, "GRiNS: A Graphical Interface for creating and playing SMIL documents", Computer Networks and ISDN Systems, Vol 30, No. 1-7, April 1998, pp. 519-529.
- [6] W3C Synchronized Multimedia Working Group, <http://www.w3.org/AudioVideo>
- [7] L. Rutledge, "SMIL 2.0: XML for Web Multimedia", IEEE Internet Computing, Vol 5, Sep.-Oct. 2001, pp. 78-84.
- [8] M. Kim *et al.*, "eXtensible MPEG-4 Texture Format", Contribution ISO-IEC JTC1/SC29/Wg11 MPEG00/6110, May 2000, Geneva, Switzerland.
- [9] S. Shim, J. Gao and W. Wang, "Multimedia Presentation Components in E-commerce", Second International Workshop on Advances Issues of E-Commerce and Web-based Information Systems, Milpitas, CA, June 2000, pp 158-165.
- [10] H. Chen, G. Chen and J. Hong, "Design of a Web-based Synchronized Multimedia Lecture System for Distance Education", International Conference on Multimedia Computing and Systems, Vol 2, 1999, pp. 887-891.
- [11] Javasoft, Java Media Framework API, <http://java.sun.com/products/java-media/jmf>
- [12] SAXDOM: The Simple API for XML for DOM,

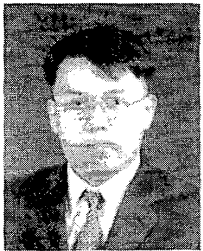
<http://www.megginson.com/SAX>

- [13] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich and T. Jin, "Evaluating content management techniques for web proxy caches", In 2<sup>nd</sup> Workshop on Internet Server Performance, Atlanta, GA, May 1999.
- [14] T. Kelly, Y. Chan, S. Jamin and J. MacKie-Mason, "Biased replacement policies for web caches: Differential quality-of-service and aggregate user value", In 4<sup>th</sup> International Web Caching Workshop, San Diego, CA, March 1999.
- [15] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.
- [16] H. Schulzrinne, A. Rao and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.

## BIOGRAPHIES



**Dongkyoo Shin** received a B.S. in Computer Science from Seoul National University, Korea, in 1986, an M.S. in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 1992, and a Ph.D. in Computer Science from Texas A&M University, College Station, Texas, in 1997. He is currently an Assistant Professor in the Department of Computer Engineering at Sejong University in Korea. From 1986 to 1991, he worked in Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he worked in the Multimedia Research Institute of Hyundai Electronics Co., Korea as a Principal Researcher. His research interests include Web-based multimedia application, XML application, mobile Internet and distributed systems.



**Dongil Shin** received a B.S. in Computer Science from Yonsei University, Seoul, Korea, in 1988. He received an M.S. in Computer Science from Washington State University, Pullman, Washington, U.S.A., in 1993, and a Ph.D. from University of North Texas, Denton Texas, U.S.A., in 1997. He was a senior researcher at System Engineering Research Institute, Deajun, Korea, in 1997. Since 1998, he has been with the Department of Computer Engineering at Sejong University in Korea where he is currently an Assistant Professor. His research interests include Mobile Internet, Computer Supported Cooperative Work, Object-Oriented Database, Distributed Database, Data Mining and Machine Learning.