자바기반의 SMIL 동작기의 설계 및 구현

김상국, 신화종, 신동일, 신동규, 송상훈

Design and Implementation of a JAVA-based Synchronized Multimedia Integration Language Player

Sangkuk Kim, WhaJong Shin, Dongil Shin, Dongkyoo Shin, and Sanghoon Song
Departmenf of Computer Engineering, Sejong University

The Synchronized Multimedia Integration Language (SMIL) is a recommendation from the W3C that makes authoring of TV-like multimedia presentations on the Web easier. It uses the eXtensible Markup Language (XML) to define a set of markup tags to synchronize the timing and positioning relationships between multimedia objects We present the design and implementation of JAVA-based SMIL player, SSP (Sejong SMIL Player). SSP processes different types of media objects using multiple threads Moreover, its cache engine detects the mediatype and allocates the proper cache memory for the corresponding media object.

1 Introduction

Web is expected to be turned into a distribution system for both interactive and synchronized multimedia content. Moreover, many of today's television programs are multimedia presentations and could be produced by using techniques learned in creating multimedia web pages Providers of synchronized multimedia content are very interested in using the Web for several reasons: easily changeable and more up-to-date content, cheaper cost 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. The Synchronized Multimedia Integration Language (SMIL) is a recommendation from the World Wide Web Consortium (W3C) that allows for the creation of time-based multimedia delivery over the web [8,9]. Based on XML, it allows developers to mix many types of media, text, video, graphics, audio, and vector-based animation together, to synchronize them to a timeline, and to make authoring of TV-like multimedia presentations on the Web easier.

This paper presents a new JAVA-based SMIL player called SSP: Sejong SMIL Player. SSP has been designed and developed as a stand-alone JAVA application. Section 2 presents an overview of SMIL. Section 3 discusses the oveall structure and main features of SSP: Sejong SMIL Player. Section 4 concludes the paper.

2 Synchronized Multimedia on Web

The new SMIL language is a powerful tool for creating synchronized multimedia presentations on the web over low bandwidth connections. SMIL offers a new way to

assemble and deliver streaming multimedia presentation [4,7]. To facilitate this great upcoming technology, well-built SMIL player is must-to-have along with a well-built SMIL authoring tool [1]. Currently several SMIL players are available around the world [9]: CWI Grins, Helio Barbizon, HPAS, NIST S2M2, Productivity Works Lp player, and RealNetworks G2.

SMIL is a XML-based declarative language to present Web-based multimedia documents. Current version of SMIL includes primitives which allow for easy implementation and early experimentation.

SMIL is an integrating format only representing how various media objects are to be positioned spatially and combined temporally. That is, it takes information objects encoded using their own formats (such as MPEG and WAV) and combines them into a Web-based multimedia presentation.

3 SSP: Sejong SMIL Player

We present the design goals, main features, and implementation details of JAVA application-based SMIL player. SSP utilizes up-to-date JMF (Java Media Framework) [3] APIs to support various media types.

3.1 Design Goals and Overall Structure of SPP

SSP consists of the SMIL parser and the rendering engine. The SMIL parser uses SAXDOM (Simple API for XML DOM) to modularize the SMIL DTD (Document Type Definition) parsing [10]. SAXDOM basically separates the underlying paring functions into a set of callback routines from the SAXDOM driver.

Since SMIL supports various types of media object such as audio, videos, and textstreams, all SMIL media objects are defined through MIME types. The parser generates a set of hash tables while parsing <head> parts of the SMIL document. Parsing <body> parts generates the corresponding media type information and stores the proper JMF class method. Finally, these stored media information is used to render by using the JMF class library. Figure 1. shows the overall structure of SPP.

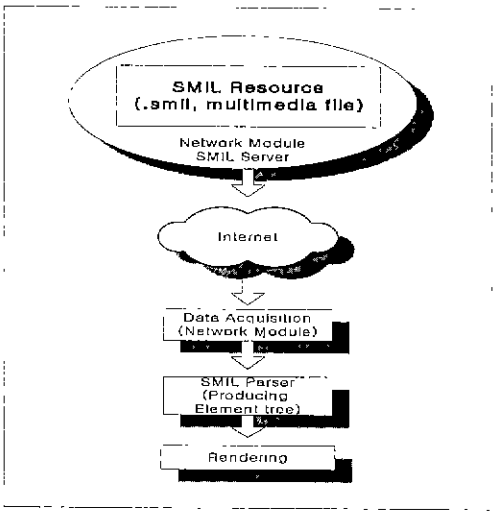

Figure 1. Overall Structure of SPP

3.2 Considerations for Performance Improvements

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, 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 a hash table. When execution of the SMIL document starts, corresponding threads are concurrently running to render and show the different media objects. Figure 2 shows the multith-readed processing of SMIL header and body elements.

Compared with traditional textual objects, multimedia

objects usually require much higher bandwidth. A typical piece of 25 second 320x240 QuickTime movie could take 2.3MB, which is equivalent to about 1000 screens of textual data. This is unimaginable in the old days when only textual data is transmitted on the net. Most multimedia objects require the real-time transfer.

Moreover, multimedia data stream is usually bursty. Just increasing the bandwidth will not solve the burstiness problem. To overcome these difficulties, SSP incorporates intelligent cache engine. The cache engine detects the media type and allocates the proper cache memory for the corresponding media object. It even 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 single client machine on which the SSP runs. Each JAR file keeps tracks of single media type. For instance, if the SPP played SMIL documents including 3 videos, 2 audio clips, and 5 textstreams, the client machines keeps 3 different JAR files.

4 Conclusion

Currently, HTML only supports static layouts and restricted multimedia presentations. The Synchronized Multimedia Integration Language (SMIL) is a recommendation from the W3C that makes authoring of TV-like multimedia presentations on the Web easier. We presented the design and implementation of JAVA application-based SMIL player.

Current version of SSP utilizes HTTP for transferring SMIL documents. However, next version of it should support RTP (Real-time Transfer Protocol) [5] and RTSP(Real-time Streaming Protocol) [6]. These standard streaming protocols for multimedia content have been developed within the IETF (Internet Engineering Task Force).

In future research, more attractive user interface and robust cache design should be incorporated into SSP. A research on a SMIL authoring tool that works with any play engines should follow.

References

[1] Bulterman, D., et. al., "GRiNS: A Graphical INterface for creating and playing SMIL documents", Computer Networks and ISDN Systems, Vol 30, No. 1-7, April 1998.

[2] Hoschka, P., "Toward Synchronized Multimedia On the Web", http://www.w3j,com/6/s2.hoschka.html

[3] Java media API, http://java.sun.com/products/java-media/jmf

[4] Rousseau, F., and Duda, A., "Synchronized Multimedia for the WWW", Computer Networks and ISDN Systems, Vol 30, No. 11-7, April, 1998.

[5] Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V., "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[6] Schulzrinne, H., Rao, A., and Lanphier, R., "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.,

[7] Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, http://www.w3.org/ TR/REC-smil/

[8] W3C Issues SMIL as a Proposed Recommendation, http://www.w3.org/Press/1998/SMIL-PR

[9] W3C Synchronized Multimedia Working Group, http://www.w3.org/AudioVideo

[10] SAXDOM: The Simple API for XML for DOM, http://www.megginson.com/SAX.