Authoring Tool for Mobile Contents based on LASeR

(LASeR 기반 모바일 콘텐츠 저작 도구)

Sun Kyung Kim*, Hee Sun Kim**
(김선경, 김희선)

Abstract MPEG 4 Part 20 LASeR (ISO/IEC 14496 20) is a specification designed to deliver rich media services in a mobile environment. The specification is an emerging standard that can replace the MPEG 4 BIFS specification designed to deliver PC based heavyweight media contents. The specification describes the representation of scene information in a resource constrained mobile environment. Unlike the BIFS standard designed to deliver heavyweight rich media, the LASeR specification has a restricted description that conforms to the SVG Tiny 1.2 specification. Also, the specification has an advantage of allowing for the efficient conversion of one graphics format to another. In this paper, we present the design and the implementation of a LASeR authoring system that allows for fast and efficient creation of interactive rich media contents in a mobile environment. The GUI interface of the authoring system presented in this paper allows users, who do not have prior knowledge of the scene description language, to conveniently create contents and store the produced scenes using the internal list data structure. The system allows users to navigate scene objects internally stored and to create LASeR XML files in the structured XML format.

Key Words : MPEG 4, LASeR, Authoring Tool, Mobile Contents

1. Introduction

There have been research studies on the scene description language designed to support user interaction in multimedia contents. Also, users have created a huge demand for interactive rich media services in the mobile multimedia environment. MPEG 4 BIFS[1] is a major standard for representing the scene description language designed to deliver rich media services in the current PC multimedia environment. BIFS has been adopted as a standard for graphics data broadcasting in DMB. However, because BIFS is a standard designed to deliver heavyweight media contents, the BIFS specification is not ideal for the mobile environment that is limited to lightweight applications. Thus, the MPEG group has proposed the MPEG-4 Part 20 Lightweight Application Scene Representation (LASeR) specification[2 5] that represents media contents for lightweight applications.

Ikivo has introduced the Ikivo Animator as an

* School of Computer & Information Technology Engineering, Daegu University
** School of Electronic & Information Industrial Engineering, Andong National University

SVG Tiny 1.2[6 7] compliant authoring tool that supports the conversion of multimedia contents developed in MPEG 4 BIFS to the LASeR graphics format. Still, research on graphics data processing and interactive rich media services in LASeR based mobile terminals is in initial phase. A user friendly authoring tool that allows the fast and efficient creation of interactive rich media contents in a mobile environment is greatly needed.

Thus, in this paper, we present an overview of the MPEG 4 LASeR specification and the design and the implementation of an authoring tool that allows users to easily create rich media contents and save the contents as structured LASeR XML files. The MPEG 4 LASeR authoring tool that this paper presents consists of the following major components: the user interface that allows users to intuitively create geometric data objects and place media objects, the attribute manager that manages the properties of two dimensional geometric data objects and media objects created by users via the GUI interface, and the LASeR XML generator that creates LASeR XML file in the binary XML format.

Chapter 2 presents an overview of the LASeR specification, Chapter 3 presents the structure of the LASeR authoring tool and the steps for the production of LASeR XML files, and Chapter 4 shows a demonstration of the system. Chapter 5 presents the study's conclusion.

2. LASeR

In the MPEG 4 system, there exist concepts such as scenes that represent the spatio temporal presentation of various media objects. The BIFS standard has been proposed as a scene descriptor language specification that represents media objects in temporal relation and the spatio temporal presentation of media objects.

However, BIFS, the MPEG 4 scene description language, has limits as it was developed under the existing PC environment which is based on the limited operation resources and the use of a battery. Thus, as an alternative to BIFS, the contents production and transmission method standardizations to provide more suitable and various multi media services under the mobile environment was started in MPEG, and the standardized measure is MPEG 4 Part20: Lightweight Application Scene Representation.

LASeR has a limited description determined in the standards to determine the scene description information for rich media services to be provided in binary form, and the scene description can be easily converted into other graphic formats.

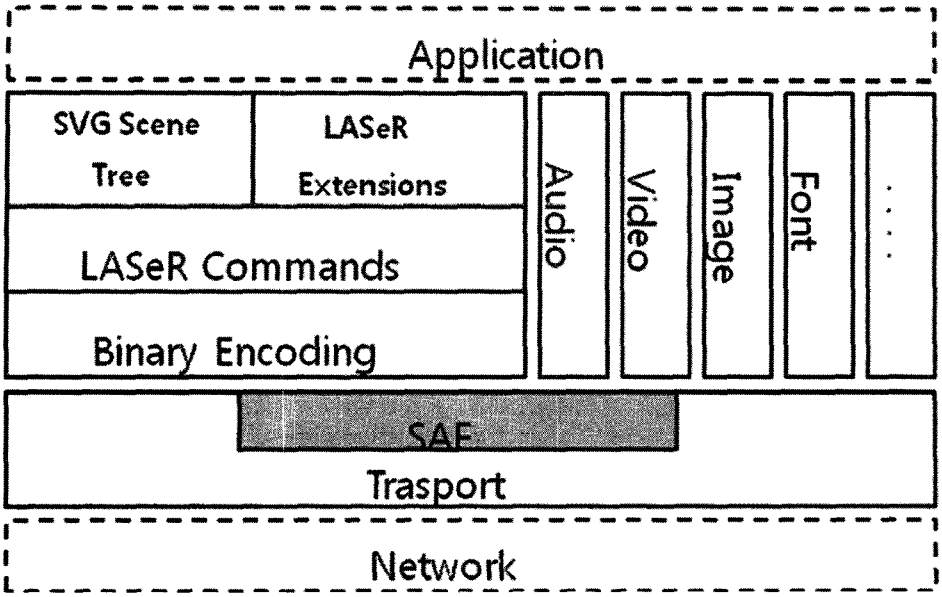
Also, as it does not download entire documents but makes the scene update and conversion in binary scale by content unit, it includes the technology to perform transmissions in streaming form, express the video and audio freely with graphic data, and multiply and transmit the data in logical stream.

Through such capabilities, the realization of dynamic rich media that supports rich screen and compressive efficiency under the mobile environment becomes possible. The architecture of the LASeR standard consists of parts as shown in the following figure 1.

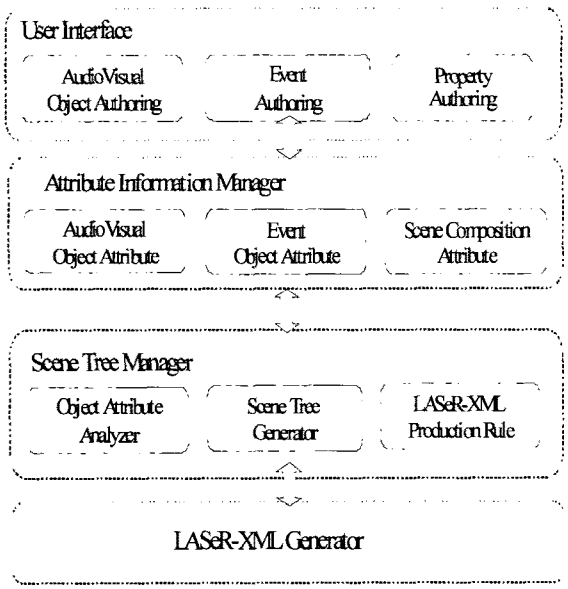
3. LASeR Authoring Tool

3.1 Structure of LASeR Authoring Tool

As shown in figure 2, the LASeR authoring tool implemented in this paper has a user Interface, an attribute manager, and a LASeR XML generator.



(Fig. 1) Architecture of LASeR[2]



(Fig. 2) Architecture of LASeR Authoring Tool

The user interface facilitates the management of LASeR scenes by providing users with various audio visual icons, property setting windows and dialog boxes. The WYSIWYG interface allows users to author audio visual objects, object properties, and events and commands.

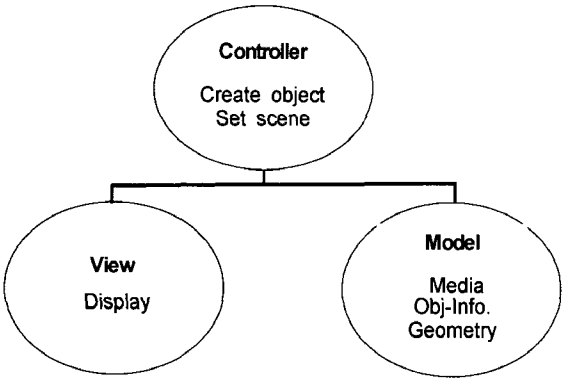
The attribute information manager creates object properties when creating two dimensional audio visual and media objects and set events for them. The attribute information manager analyzes the properties required for each object and defines properties when creating LASeR scenes. The scene tree manager creates scene trees in order to convert audio visual objects, events, and properties to XML formatted data. The LASeR XML generator produces LASeR XML files in the binary XML format for scene trees created by the scene tree manager.

3.2 User Interface

The user interface generates 2D geometrical objects like the ellipse, rectangle and polyline, as well as objects like video, audio, and images as it interacts with object units presented in icons and graphics that are shown to the general user. When a scene is composed, the scene command sets the attribute and time information of each object and composes LASeR contents with the setting of the event

information.

As shown in figure 3, the user interface was designed in the MVC pattern to support the non synchronized action of the general user. The types of objects and events for the user to be used in the composition in the user interface are shown in the following table 1.



(Fig. 3) Editor's MVC Pattern

3.3 Attribute information manager

The attribute information manager interacts with the user interface. When the general user creates a 2D object and sets the event and composes the scene, the attribute information manager produces the attribute of each object and the scene composition information and event attribute information. Also, when they are changed, the changed values are renewed and, as they interact with the input of the user on the user interface, it consistently manages the attribute information.

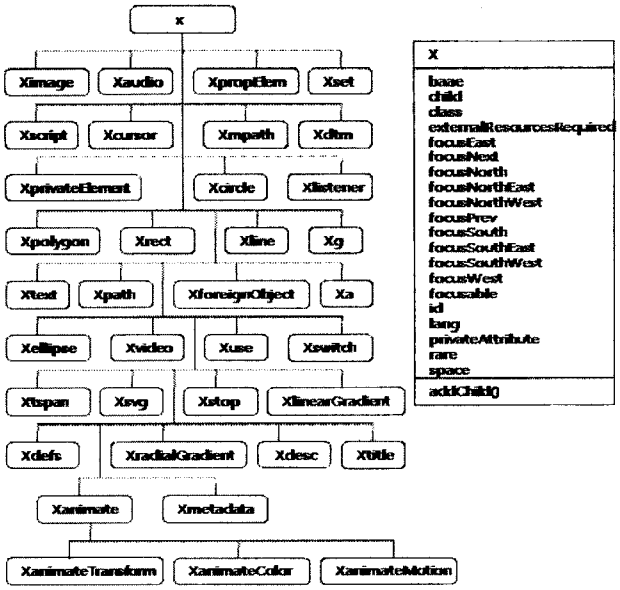
3.4 Creating Scene Tree and LASeR files

Authored scenes are internally stored in the proprietary object list data structure format. Objects and properties are inserted into the list data structure in their order of creation. The attribute information manager creates and maintains theobject list. The LASeR XML tree structure is created with the scene property information from the object list. The scene tree generator utilizes the information stored in the attribute information manager to analyze objects and encode objects into XML trees. See figure 4 for class hierarchy for creating scene trees.

Users can use the XML generator to store scene trees as LASeR XML files in the binary format, or can pass scene trees through the binary encoder and SAF generator to create SAF formatted data that can be transmitted to mobile terminals.

<Table 1> The types of objects and events for authoring LASeR contents

Objects	Events
2D geometry objects (circle, ellipse, poly gon, polyline)	scale, color, fill, opacity, transition, rotation, linearGradient, radialGradient, animate, animateColor, animateTransform
Text/Cursor	size, color, transition, rotation, animateTransform
Audio	transition, start, stop, volume, repeat
Video	transition, scale,start, stop, repeat, rotation
Image	transition, rotation, animateTransform
Switch	whichChoice

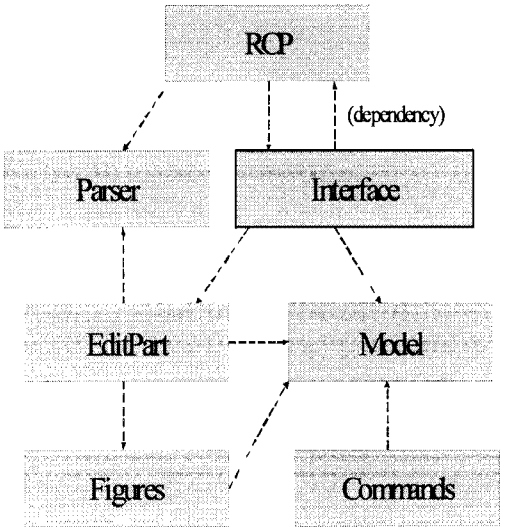


(Fig. 4) The class hierarchy for creating scene trees.

4. Implementation and results

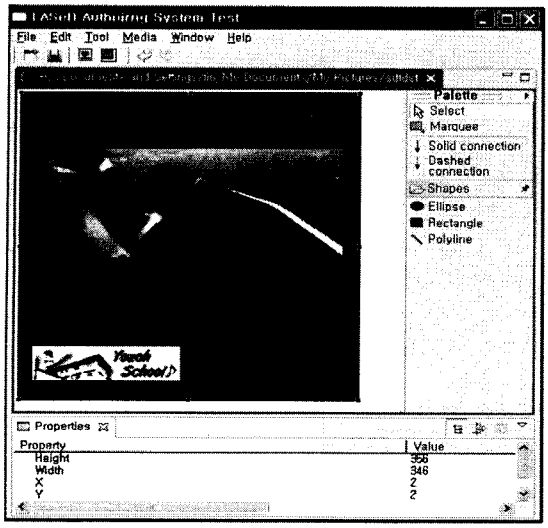
The LASeR authoring tool was realized with the development of RCP using IBM's Graphical Editor Framework (GEF) in the Eclipse 3.1 development environment. We used the GEF's MVC pattern to design and realize the LASeR authoring tool. The View expresses the 2D geometrical object and media object that the general user wants using Draw2d library provided by GEF. Model and View interacts with the user's input on the screen at every moment upon detection of changes between the two at the class that succeeds the EditPart object. The Model renews the model attributes by EditPart, the controller, whenever there is user input.

LASeR authoring tool was designed in an object-oriented way with seven packages, including the RCP package for application operation, LASeR Graphical Editor package for performing editing functions by the user's input, the Model package for changing and maintaining the model's attribute, EditPart package for playing as the controller, the Commands package for performingthe order function to create and remove the model, the Figure package for container role to show the media file, andthe View and Parser package for creating and extracting the LASeR-XML file, as shown in the following figure 5.



(Fig. 5) Structure of the LASeR Authoring Tool Package

Fig. 6 shows the interface of the authoring tool. The user interface in figure 6 shows the canvas, which represents two-dimensional geometric objects and text, the drag-and-drop enabled palette, the outline used to add media objects, and the property window that shows the properties information of the selected object. Automatically storing LASeR scenes triggers the creation of LASeR XML files. We authored simple geometric objects and media objects and then stored the objects as LASeR-XML files. Table 2presents the LASeR XML file created for the scenes created in Figure 6.



(Fig. 6) GUI of LASeR Authoring Tool

5. Conclusion

In this paper, we have presented an authoring tool that conforms to the MPEG 4 LASeR specification that enables the interactive multimedia services in mobile environments. This tool provides an intuitive GUI interface that allows users who do not have prior knowledge about LASeR language to quickly and easily produce LASeR contents. With the user friendly GUI interface, users can author audio and visual objects, properties, and events. The authoring tool can automatically store the authored contents as a LASeR XML file.

<Table 2> An example of generated LASeR-XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<saf:SAFSession xmlns:ev="http://www.w3.org/2001/xml-events" xmlns:saf="urn:mpeg:mpeg4:SAF:2005"
                xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.w3.org/2000/svg"
xmlns:lsr="urn:mpeg:mpeg4:LASeR:2005">
    <saf:sceneHeader>
        <lsr:LASeRHeader profile="full" />
    </saf:sceneHeader>
    <saf:RemoteStreamHeader streamID="Video0" objectTypeIndication="32"
                                        streamType="4" url="MV001.MPG"/>
    <saf:RemoteStreamHeader streamID="Image0" objectTypeIndication="32"
                                        streamType="4" url="timgae.jpg"/>
    <saf:sceneUnit rap=true>
        <lsr:NewScene>
            <svg id="root-node" width="350" height="360"
                                viewport-fill="rgb(0,255,255)">
                <g>
                    <rect x="2" y="2" width="346" height="356"
                                                fill="green" stroke="black"/>
                    <rect x="15" y="248" width="320" height="45"
                                                fill="blue" stroke="black"/>
                    <circle cx="36" cy="268" r="16"
                        fill="red" stroke="black" stroke-width="1" />
                        <text font-size="10" x="90" y="270" fill="rgb(0,0,0)" font-family="Arial" font-style="normal" font-
weight="bold">
                        LASeR-AuthoringTool Testing..
                    </text>
                    <video xlink:href="#Video0" begin="2" transformBehavior="pinned" repeatCount="3"
                                x="15" y="8" width="320" height="240"
                                                fill="blue" stroke="black"/>
                    <image xlink:href="#Image0" begin="2" x="16" y="295" width="150" height="40"/>
                </g>
            </svg>
        </lsr:NewScene>
    </saf:sceneUnit>
    <saf:endOfSAFSession/>
</saf:SAFSession>
```

Researching LASeR specification, objects, properties, events and scenes has enabled us to develop the user friendly GUI interface. The produced objects are internally maintained using the proprietary list data structure. When creating LASeR XML files, the authoring tool navigates the list of objects and calls the LASeR XML creation module to create Scene trees and LASeR XML files.

Our further research plan is to enhance the event processing feature, integrate an SAF encoder that enables transmission in mobile terminals, and ultimately implement the framework on which users can author various types of multimedia objects that can be played in mobile terminals.

Acknowledgment

This work is supported by Daegu University Grant.

Reference

[1] ISO/IEC 14496 1:2000 MPEG 4 Systems October 2000.

[2] ISO/IEC 14496 20:2006(E) Information technology Coding of audio visual objects Part 20: Lightweight Application Scene Representation(LASeR) and Simple Aggregation Format(SAF)

[3] Jean Claude Duford, Olivier Avaro, Cyril Concolate "LASeR : the MPEG Standard for Rich Media Servicse", IEEE Multimedia, http://www.mpeg LASeR.org/documents /LASeRWhitePaper.pdf

[4] LASeR web site, http://www.mpeg laser.org /html/standProcess_history

[5] 박상현, 김병철, 김규헌 "휴대 단말용 대화형 데이터 서비스 기술 LASeR(Lightweight Application Scene Representation)", 전자공학회지 제 34권 제 8호, 2007. 8

[6] W3C, Scalable Vector Graphics(SVG) 1.1 Specification[Recommendation], http://www.w3.org/TR/2003REC SVG11 20030114

[7] W3C, Scalable Vector Graphics (SVG) Tiny 1.2 Specification [Last Call]

김 선 경 (Sun Kyung Kim)

- 1979년 이화여자대학교 수학과 학사
- 1982년 한국과학기술원 전산학과 석사
- 1991년 미국 Minnesota대학교 전산학과 박사
- 1992년~현재 대구대학교 컴퓨터·IT공학부 교수
- 관심분야 : 과학계산, 병렬처리, 알고리즘, 멀티미디어시스템

김 희 선 (Hee Sun Kim)

- 1996년 2월 : 대구대학교 전산정보학과(공학사)
- 1998년 2월 : 경북대학교 컴퓨터과학과(이학석사)
- 2001년 8월 : 경북대학교 컴퓨터과학과(이학박사)
- 2005년 3월~현재 : 안동대학교 전자정보산업학부 조교수
- 관심분야 : 멀티미디어 시스템, 모바일 콘텐츠 응용, 인간과 컴퓨터 상호작용