EXTENDING SMIL WITH 3D AUDIO

Kari Pihkala and Tapio Lokki

Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory
P.O.Box 5400, FIN-02015 HUT, Finland
kari.pihkala@hut.fi, tapio.lokki@hut.fi

ABSTRACT

This paper describes how SMIL can be extended to support 3D audio in a similar fashion than AABIFS does it for MPEG-4. The SMIL 2D layout is extended with an extra dimension to support a 3D space. New audio elements are positioned in the 3D space, whilst a listener element defines a listening point. Similarly to AABIFS perceptual modeling approach, an environment element describes environmental parameters for audio elements. These extensions enable interactive 3D audio capabilities in SMIL. In addition, any XML based rendering language could be extended with 3D audio capabilities by using a similar approach.

1. INTRODUCTION

Stereo sound systems have been around for a long time. The recent technical development is to use additional channels to enhance the listener's experience. These 5.1-channel home theaters are supported by the latest DVD players and desktop computer sound cards. The advantage of added channels is a more immerse audio world in games and multimedia presentations. However, some of the recent multimedia standards have not kept up with this technical advancement.

Synchronized Multimedia Integration Language 2.0 (SMIL) is a new multimedia format for the World Wide Web [1]. It allows integration of media objects spatially and temporally. It is an open standard, targeted to replace proprietary multimedia standards currently used in the Web. In addition, SMIL has been adopted as the content description language for the Multimedia Messaging Service (MMS) used in mobile phones.

Goose et al. [2] have proposed a framework for three dimensional (3D) audio presentations by extending SMIL. They describe a complete system architecture to deliver 3D audio into mobile devices by preprocessing a SMIL presentation on a server and delivering a resulting stereo audio stream to mobile devices. However, their approach lacks interactivity and works only for purely audio-based presentations, disregarding visual presentations all together.

MPEG-4 is a comprehensive multimedia standard including audio and video compression, scene description, multiplexing, and synchronization [3]. The scene description, called binary format for scenes (BIFS) is similar to SMIL, and it defines how MPEG-4 objects are composed together for presentation. MPEG-4 Advanced AudioBIFS (AABIFS) describes a 3D audio scene description, taking full advantage of audio spatialization.

XMT-Ω, the textual format of MPEG-4, is based on SMIL 2.0. It includes SMIL Animation, Content Control, Media, Metainformation, Timing and Synchronization, Time Manipulations, and Transition Modules, but also extends SMIL with its own set of elements. These include means for 3D spatialization. XMT-Ω is only meant to ease the authoring of MPEG-4 scenes and must be compiled into binary MPEG-4 content before playback. Thus, it is not a presentation language and cannot be used to play back 3D audio presentations as such [4].

This paper describes how SMIL can be extended to support 3D audio. The AABIFS scene description is taken as a reference. First, SMIL and AABIFS are briefly introduced to unveil their differences. Then, the 3D audio extension for SMIL is described. After that, the system architecture for the extension is given, while the last section draws conclusions.

2. BACKGROUND

This section describes the currently available audio related features in the SMIL 2.0 standard and briefs the 3D audio features in MPEG-4 AABIFS.

2.1. SMIL

One of the main design goals developing SMIL 2.0 was modularity. The SMIL 2.0 specification defines 10 functional areas, which are further divided into 45 modules. The modules define small pieces of functionality, as their names imply: Media Objects, Basic Layout, Prefetch Control, Basic Linking, and Spline Animation. The modules are combined to create language profiles, e.g., the SMIL Host Language profile contains almost all of the SMIL modules.

SMIL Media Object Modules can include external media objects in a presentation. The media objects are referenced with a Uniform Resource Identifier (URI), which is an extension of the commonly used URL. The SMIL specification does not define which format the external media files should use. It is up to the SMIL player to support any media format it desires, however it is recommended to support at least the following MIME types: audio/basic, image/png, and image/jpeg. Typically, mono or stereo audio objects are used, but it is possible to reference a 5.1 channel audio object, too. However, this means that the 3D position information is saved in the external object and cannot be interactively modified with SMIL.

The SMIL Layout Modules define spatial placement with so called regions. The position of a region is defined with absolute coordinates, which describe the top-left and bottom-right corners in two dimensions. Media objects are then placed in these regions. Various visual effects can be applied to regions and media objects, e.g., animation and transition effects. SMIL also allows an audio object to be assigned to a region, causing the region to alter the volume of the audio object. However, SMIL does not provide audio spatialization or other audio effects, such as environmental effects or filtering.

2.2. AABIFS

MPEG-4 AudioBIFS defines basic audio nodes for MPEG-4. These are Sound and AudioClip nodes, which define the location, direction, intensity, and a reference to an audio stream. Others, AudioSource, AudioMix, AudioSwitch, AudioFX, AudioDelay, AudioBuffer, ListeningPoint, and Sound2D provide streaming, mixing, effects processing, listening point, and interactive playing. The AudioFX node can be used to add sound filtering effects, expressed in Structured audio orchestra language (SAOL), which is a digital signal processing language defined in the MPEG-4 standard. The Sound2D node offers a way to place a sound in a 2D plane specifically designed 2D applications in mind, where a viewing point is assumed to be 1 meter outwards from the center of a 2D plane. Usually, a listener is located at the same spot as the viewing point, however it is possible to position it elsewhere using a ListeningPoint node.

The Advanced AudioBIFS nodes add features for virtual acoustic rendering with either physical or perceptual approach. The physical approach mimics real physical acoustics by simulating sound propagation in a 3D scene made of surfaces. An AcousticScene node can be used to describe generic acoustic parameters of the scene. These parameters include rendering region of the sound, late reverberation, and grouping of acoustic surfaces. An AcousticMaterial node defines acoustic properties of surfaces in a 3D scene.

A PerceptualParameters node is used in the perceptual approach to mimic audio rendering instead of trying to simulate physical properties of the scene. This node mainly affects the reverberation properties of sound nodes, dividing reverberation into four sections: direct sound, directional early reflections, diffuse early reflections, and late reverberation. Also, three frequency bands can be controlled: low-, mid-, and high-frequency bands. In addition to these, the modal density of a room response can be controlled, the effect of distance can be altered, and parameters to simulate occlusion, where a sound source is behind a source-obstructing object, can be specified.

A DirectiveSound node is similar to the Sound and Sound2D node with a better control over directivity of the sound source. It is possible to specify attenuation of the sound as a function of its frequency. It also specifies control over sound propagation delay with speed of sound, while attenuation can further be controlled with distance factor, and air absorption.

3. ADVANCED AUDIO IN SMIL

SMIL is based on XML, and thus it is extendable. A new module called Advanced Audio Markup Language (AAML) is created to support 3D audio.

Unfortunately, AABIFS nodes converted into XML elements or XMT-Ω elements cannot be used as such, because they do not take into account SMIL specific layout or timing. Therefore, slightly different elements have been designed along with an extension to SMIL layout.

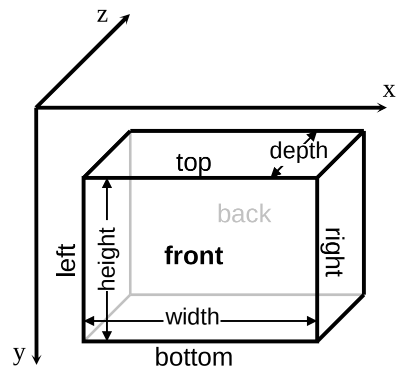

Figure 1. *Standard SMIL coordinates extended with front, depth, and back coordinates.*

3.1. Extended SMIL Layout

The standard SMIL 2.0 Language only supports 2D coordinate system designed for two-dimensional presentations. In 3D audio, as the name implies, there is also third dimension. The SMIL Layout Modules define a spatial position with left, width, right, top, height, and bottom attributes. These are extended with a similar system for third dimension by adding front, back, and depth attributes, as depicted in Figure 1. The new attributes do not affect the standard SMIL layout or media objects, they only affect the 3D audio objects. It must be noted that the new attributes are specifically designed for this purpose and most likely cannot be used as a general solution for 3D graphics in SMIL. In a 5.1-channel audio system, the third dimension is audible as distance dependent attenuation, and position between front and back loudspeakers.

The proposed new attributes comply with all SMIL Layout features. If a depth attribute is also added to a root-layout element to specify the overall depth of a presentation, then percentage values (e.g., position at 35% of the presentation depth) can be used instead of absolute values, and any missing values can be resolved (e.g., front value can be calculated based on back and depth values). SMIL Hierarchical Layout Module defines hierarchical spatial placement inside regions with subregions. In this case, the new attributes are treated in a similar fashion as the standard coordinates, i.e., they are translated and clipped by a parent region.

The new attributes can be animated with the SMIL Animation modules. This allows moving a region along a z-axis, too. The standard SMIL animateMotion element can move elements along a two-dimensional path. It is possible to create a similar animation element for 3D paths.

The units used in coordinates are pixels. They are converted into meters to make audio spatialization meaningful. If the application requires other metrics, it is possible to scale, rotate, and translate the coordinates.

As a remark, the described layout extension is not necessary, if two-dimensional positioning is considered to be adequate. In most cases, listeners will only be able to recognize the x-axis position due to limitations in sound reproduction systems, e.g., stereo systems.

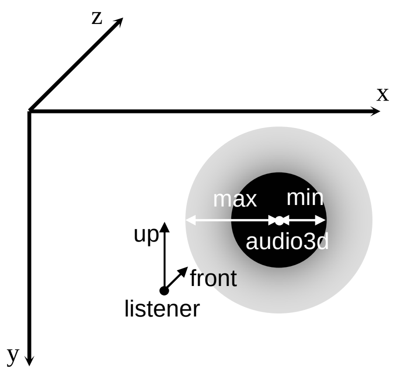

Figure 2. *Audio and listener elements.*

3.2. Audio and Listener Elements

The only controllable parameter in the standard SMIL audio object is the sound level, which is given in percentages of the original sound volume. To support spatial positioning, a new audio element is created. It places itself in the position given by its region. For realism, the sound attenuates in correlation with distance. The attenuation is controlled by two attributes. Audio is played at a given sound level within minimum distance (minDistance), while it is not played at all outside maximum distance (maxDistance). The sound level changes gradually between these two distances, as shown in Figure 2. Generally, media objects in a SMIL presentation do not have any orientation compared to objects in BIFS scenes, and thus audio objects are represented without orientation. Ambient sounds can be produced by setting the minimum distance value to the size of a presentation.

Now, as audio elements have a position, there has to be a listening point. For this, a listener element is defined in the head section of the document. By default, the listener is positioned in the middle of the window, a short distance backwards from it, oriented along the z-axis, as depicted in Figure 2, to mimic the position of a real user behind a screen. Thus, audio objects on the left side of the screen will be heard from the left side, while audio objects on the right are heard from right. It is possible to modify the default position of a listener with the left, top, and front attributes associated with the listener element.

In addition to the previously mentioned attributes, it is also possible to enable Doppler shift effects. Then, if an audio or listener is moved, its velocity will affect the pitch of the sound. The distance attenuation between minimum and maximum distances and Doppler shift effect can be increased or decreased with distanceFactor and dopplerFactor attributes. Altering these will sometimes create more pleasant effects.

3.3. Audio Environment

The audio environment can be modeled with perceptual parameters, defined in the head part of the SMIL document with an environment element. The parameters that can be modified are similar to those in the EAX 2.0 API [5] because of the implementation, cf. Section 4. One of the 25 preset environments (e.g., generic, bathroom, cave, concert hall, and underwater) can be selected with a preset attribute, while the rest of the attributes can be used to override the preset parameters (e.g., decayTime, reflectionsDelay, reverbDelay, and airAbsorption).

3.4. Example

The example in Figure 3 shows how AAML is used in SMIL. The second line defines the namespace prefix for AAML. All AAML elements and attributes are prefixed with this prefix to distinguish them from SMIL elements. Line 4 defines that a bathroom environment with overridden decay time is used as the environment. Line 5 describes the position of a listener. The region element in line 7 is extended with a third dimension. The same region will be used to position an image and an audio object. An image in line 11 is rendered in the region. Audio will be played in line 12, in the position given by the region. The animation element in line 13 will move the image and audio from left to right, when the image is clicked.

4. IMPLEMENTATION

This section describes an overall architecture for a client-side SMIL player with an AAML extension.

4.1. The SMIL Player

The proposed new audio module was added to a SMIL player [6] developed for the X-Smiles browser [7]. The X-Smiles browser

```
 1: <smil xmlns=”http://www.w3.org/2001/SMIL20/Language”
 2:       xmlns:aa=”http://www.x-smiles.org/2002/aaml”>
 3:    <head>
 4:       <aa:environment environment=”bathroom” decayTime=”5.0”/>
 5:       <aa:listener left=”200” top=”300” front=”30”/>
 6:       <layout>
 7:          <region id=”pic” left=”10” top=”20” width=”100”
 8:                  height=”200” aa:front=”10” aa:depth=”10”/>
 9:       </layout>
10:    </head><body><par>
11:          <img id=”i” region=”pic” src=”hello.gif” dur=”60s”>
12:          <aa:audio3d region=”pic” src=”music.wav”/>
13:          <animate targetElement=”pic” attributeName=”left”
14:                   from=”10” to=”400” begin=”i.click” dur=”10s”/>
15:     </par></body>
16: </smil>
```

Figure 3. *Example SMIL with 3D audio.*

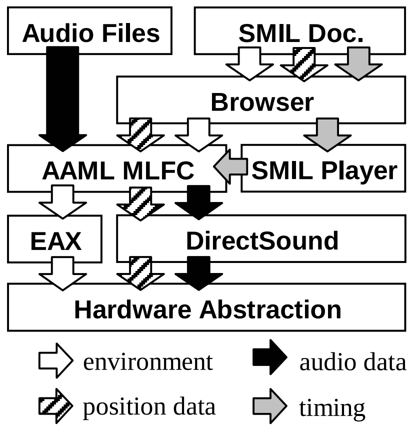

Figure 4. *Overall Architecture.*

allows adding new markup languages and linking them to the existing languages. This extendibility made it easy to add the new AAML module.

4.2. Rendering Architecture

The audio rendering architecture is similar to that suggested by Trivi et al. [8], as shown in Figure 4. The SMIL document contains position information for audio and listener elements, environment data, and URI references to audio files. These are read in by the browser, which forwards all but standard SMIL timing information to the AAML module. The AAML module renders audio and listener objects through Microsoft's DirectSound API, while environmental effects are achieved with Creative's EAX 2.0 API [5]. The timing for all this is given by the SMIL player.

The browser, SMIL player, and AAML module are implemented in Java, while DirectSound and EAX APIs are only accessible in low-level languages, such as C++. Therefore, Java Native Interface (JNI) is used in Java to access a piece of C++ code, which then calls the APIs.

The implemented AAML module recognizes run-time changes in the attributes of AAML elements, i.e., animation of coordinates or changes in environmental parameters are recognized. The implementation samples these attributes every 50 ms and forwards changes to the DirectSound and EAX APIs through the JNI interface. The Hardware Abstraction layer then alters its audio output accordingly.

A possible alternative rendering approach of the proposed extension is similar to the one Goose et al. [2] used. A SMIL presentation is preprocessed on a server by resolving timing and synchronization of media elements, which are then mixed into a stereo audio stream. The stream is then delivered to a client, which can play it out with a standard audio player. Of course, interactivity is then compromised.

5. CONCLUSIONS

The current SMIL 2.0 standard supports only simple audio output without any spatial control. The MPEG-4 AABIFS takes a more comprehensive approach, describing spatialized interactive audio objects with effects, such as room acoustics.

A new XML based language for 3D audio was extended to the SMIL language. The new language allows spatial positioning of audio objects in three dimensions. Also, a listening point can be positioned in the same space. Perceptual environment acoustics can be altered to mimic real world acoustics.

The new language can also be used with any other rendered XML language, such as XHTML or SVG. Their layout can be extended using the proposed approach, if two-dimensional positioning is not adequate. The other proposed elements can be used as such without any complications.

In the future, better control over directivity of a sound source could be provided. Also, the elements and attributes could be made more similar to those in AABIFS for interoperatibility.

6. ACKNOWLEDGEMENTS

The author Kari Pihkala would like to thank Nokia Foundation for financial support during the research work.

7. REFERENCES

[1] L. Rutledge, "SMIL 2.0: XML for Web multimedia," *IEEE Internet Computing*, vol. 5, no. 5, pp. 78-84, 2002.

[2] S. Goose, S. Kodlahalli, W. Pechter, and R. Hjelsvold, "Streaming Speech: A Framework for Generating and Streaming 3D Text-To-Speech and Audio Presentations to Wireless PDAs as Specified Using Extensions to SMIL," in *Proc. of the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, May 7-11, 2002, pp. 37-44.

[3] F. Pereira and T. Ebrahimi, *The MPEG-4 Book.* Prentice Hall, Upper Saddle River, NJ, USA, 2002.

[4] M. Kim, S. Wood, and L. Cheok, "Extensible MPEG-4 Textual Format (XMT)," in *Proc. of the 2000 ACM Workshops on Multimedia*, LA, California, USA, October 30 – November 4, 2000, pp. 71-74.

[5] Creative Technology Ltd., "Environmental Audio Extensions: EAX 2.0," EAX 2.0 Extensions SDK, available at http://developer.creative.com/.

[6] K. Pihkala and P. Vuorimaa, "Design of a Dynamic SMIL Player," in *Proc. of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 26-29, 2002, pp. 189-192.

[7] P. Vuorimaa, T. Ropponen, N. von Knorring, and M. Honkala, "A Java based XML browser for consumer devices," in *The 17th ACM Symposium on Applied Computing*, Madrid, Spain, March 10-13, 2002, pp. 1094-1099.

[8] J.-M. Trivi and J.-M. Jot, "Rendering MPEG-4 AABIFS Content Through A Low-Level Cross-Platform 3D Audio API," in *Proc. of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 26-29, 2002, pp. 513-516.