

SMIL to MPEG-4 BIFS conversion

Beilu Shao, Laureano Moro Velázquez, Nicolas Scaringella, Neha Singh, Marco Mattavelli
Signal Processing Institute (ITS-LTS-3), Swiss Federal Institute of Technology (EPFL),
Lausanne, CH-1015, Switzerland

beilu.shao@epfl.ch, laureano.moro-velazquez@epfl.ch, nicolas.scaringella@epfl.ch,
neha_ns@cse.iit.ac.in, marco.mattavelli@epfl.ch

Abstract

Traditional media, such as text, image, audio and video, have long been the main media resources and granted full support of standard desktop tools and applications. Interactive rich multimedia documents, adding resources such as video or synthetic animations and relying on complex synchronization among objects, are now making their entrance into the world as new multimedia formats emerge. In this context, the Synchronized Multimedia Integration Language (SMIL) is receiving more and more attention from content authors due to its fine property of multimedia synchronization and authoring interactivity for the content production. At the same time, MPEG-4 is designed to address the requirement of new generation of highly interactive multimedia applications, while simultaneously maintaining the support of traditional applications. MPEG-4 provides facilities (XMT and BIFS) to integrate and synchronize, spatially and temporally, many different media objects together. However, these facilities lack appropriate authoring tools to widen its audience and subsequently limit the application.

In this paper, we present a comparative analysis between SMIL and XMT, the textual description of MPEG-4, to illustrate the pros and cons of these two major interactive media. And then we propose a conversion scheme from SMIL to the Binary Format for Scenes (BIFS) of MPEG-4 to take advantage of both formats. According to this scheme, we design the real implementation method using the current available tools and discuss the purpose and significance of such conversion.

1. Introduction

The promise of multimedia computing grown with the development of other relevant information technologies brings a revolution of electronic media into the human world. The traditional ways of conveying information among people are now being

substituted day by day. While the introduction of audio/video media objects into computer-based presentations has been technically possible since the beginning of the nineties of last century, there is still a long distance to fully standardizing them as commonly used desktop tools, such as word processors for text and slide-based presentations for demonstrations. The underlying problem is that interactive multimedia authoring is such a complicated process that it requires the author to specify many different kinds of information at different levels [1]. For instance, to add some media items into another media object, the author needs to concern the spatial layout, where to place these items, temporal relationships, when to display the items as well as other functionalities such as interaction with the author, hyperlinking behaviors between other media objects. The scenarios of transmitting of multimedia presentations via a network infrastructure, e.g. multimedia streaming, increase the complexities of the production process. It is necessary to specify additional information (e.g. metadata information) to facilitate adaptation to the delivery environment (dependent on the available bandwidth or device capabilities) and the needs of a diverse user community.

Traditional media, such as text, image, audio and video, have long been the main media resources and granted full support of standard desktop tools and applications. Yet, as new media formats emerge, *rich* multimedia presentations are making their entrance into the world and are being increasingly used for newscasts, educational material, entertainment etc.

By *rich* media, we refer here to a broad range of digital media choreographing audio, video, text, graphics and synthetic animations in real time. Another key feature of such media is interactivity. Rich media such as SMIL and MPEG-4 indeed respond directly to user interactions allowing for new ways of consuming media.

In the recent years, a large number of new multimedia formats have been created independently from each other including among others QuickTime, RealVideo, Advanced Streaming Format, Shockwave,

SMIL and MPEG-4. In this paper we focus on the last two of these formats. They are both non-proprietary and support a variety of application scenarios [2]. Also SMIL and MPEG-4 are both specifically designed to address the issue of integration and synchronization of multimedia.

Though SMIL and BIFS have very similar functionalities and target applications, the former appears to be much easier to author while the latter seems more suitable for broadcasting and video-on-demand applications. Additionally, BIFS offers mechanisms for copy protection and intellectual property management [3]. Moreover, there is a broad potential for BIFS in the PDA sector since these appliances are based on chip sets and not downloadable software. However, BIFS is difficult to edit and subsequently hard to use because the binary coded representation often cannot be "reverse-engineered" in a consistent manner to represent the content author's original intentions.

This paper is organized as follows: the next two sections will describe the SMIL format and the MPEG-4 BIFS format in terms of structure and layout, timing and synchronization, supported object formats, user interaction function, typical application tools. Based on these comparative analyses, Section 4 will emphasize the similarities between both formats and discuss about the meaning of taking the advantage of both media formats. And then a relevant conversion scheme from SMIL to BIFS is introduced and the method for real implementation using the current available tools is given. The last section will reach some conclusions and draw some perspective of the future work.

2. The SMIL Format

SMIL (Synchronized Multimedia Integration Language) is developed by the W3C SYMM working group in March 1997. Since then more and more features and relevant tools have been provided to improve its functionalities and applications. The last recommendation is SMIL 2.1 published in December 13th 2005[4]. SMIL is an XML-based language which is self-describing and easy to use like HTML. It allows integrating a set of independent media objects into a synchronized multimedia presentation.

2.1 Main Applications

The SMIL language finds most of its applications in multimedia presentations. It allows arranging different media together (video, audio, text, 2D animations...) with total synchronization and possible user interaction.

SMIL's target audience is mainly community since it allows authoring of multimedia presentations using different source servers with total synchronization.

2.2 SMIL Modules and Profiles

Since SMIL2.0, The strategy of modularization and profiling has been adopted for the concern of integrating respective functionality with other XML-based languages like HTML. Modularization is an approach in which markup functionality is specified as a set of modules that contain semantically-related XML elements, attributes, and attribute values. Profiling is the creation of an XML-based language through combining these modules, in order to provide the functionality required by a particular application [4]. The strategy of modularization and profiling partitions the SMIL features into a set of ten markup modules, and provides a framework for adding these modules into other XML document formats, including the Scalable Vector Graphics (SVG) [5].

These markup modules contain semantically-related XML elements, attributes, and attribute values. Each module is uniquely identified by an XML Namespace Identifier. The modules are summarized as follows: (for more details, please refer to SMIL 2.1 specification[4]):(1) the Animation Module: including animation onto a time line, and a mechanism for composing the effects of multiple animations; (2) the Content Control Modules: providing runtime content choices and optimized content delivery; (3) the Layout Module: allowing allow positioning of media elements on visual and audio rendering surfaces and to control of audio volume; (4) the Linking Module: enabling navigation through the SMIL document which can be triggered by user interaction or other triggering events, such as temporal events. Most of SMIL linking constructs are similar to that from XLink[6]; (5) the Media Object Module: describing SMIL media objects such as image, audio, video, text, textstream, animation and ref and object attributes; (6) the Meta Information Module: describing the properties of the SMIL document and media objects with meta. Supporting the recommendation of Resource Description Framework (RDF)[7]; (7) the Structure Module: containing elements for structuring SMIL content; (8) the Timing and Synchronization Module: specifying the properties of begin, end and duration of an element, etc.; (9) the Time Manipulations Module: controlling the speed of playback of a media object, etc.; (10) the Transition effects Module: describing transition effects such as Edge, Iris, Clock and Matrix wipes.

In the succeeding chapters, we will focus on some of these above modules: the structure and layout, timing and synchronization and linking module. These

are crucial and key properties where interactive and synchronized composite multimedia stands out.

2.3 Structure and Layout

A SMIL file is divided into a header and a body section. The header contains information on the positioning of visual objects into different regions within a given layout and defines how the media linked to the surfaces should be rendered as well as the sound configuration. Transition effects are also defined in this section. The body contains the references to the media sources. It defines their temporal relationships, interactive behaviors between objects, timing manipulations and the possible interactions with users. Transition effects defined in the header sections can also work coordinately with the corresponding media object in the content body. Besides, SMIL also supports alternative layout scheme, such as Cascading Style Sheet (CSS).

2.3 Timing and Synchronization

SMIL media objects can be temporally related in sequence, in parallel, or exclusively. Such functions are implemented using the time containers <seq>, <par> and <excl> respectively. SMIL timing also provides attributes to specify an element's timing behavior. Elements have a begin, a simple duration or a end attributes. The begin attribute can be defined in various ways. For example, an element can begin at a given time, or based upon when another element begins, or when some user interaction event (such as a mouse click) happens. Elements can also be defined to repeat the simple duration, a number of times or for an amount of time. The simple duration and any effects of repeat are combined to define the active duration. When an element's active duration has ended, the element can either be removed from the presentation or frozen (held in its final state), e.g. to fill any gaps in the presentation.

The media element will be activated when it begins its active duration, and becomes inactive when it exceeds the active period. The following is an example of a container with parallel relation of inside media elements (for simplicity, we omit the src and region attributes):

```
<par>
  <video id="vid" begin="3s", dur="50">
  <audio begin="3s", dur="50">
  <img begin="vid.begin +3s">
</par>
```

Both the audio and video are children of the <par> container and hence are in parallel synchronization. Both start at 3s while the image appears 3 seconds after the video begins playing and remains for as long as the presentation is active. This is just a very basic example to demonstrate the flexibility of SMIL timing. For more details of the timing and synchronization, the reader can refer to the SMIL2.1 specification [4].

2.3 Supported Formats

SMIL does not specify the availability of supported multimedia formats. The rendering of SMIL presentations entirely depends on the formats and codecs supported by the relevant SMIL player in use. If the player cannot render a kind of media format included in the SMIL presentation, this media resource will be skipped by default. This property turns the extensibility of SMIL supported formats into the upgrade of relevant SMIL player. We will discuss such advantage of the flexibility by showing the structure of the SMIL Ambulant Player in section 2.6.

Moreover, SMIL guarantees the synchronization between all sources. If the streaming of one of the source is delayed, the presentation will be stopped or the source will be ignored.

2.4 User Interaction

Interaction is achieved using hyperlinks in SMIL. SMIL allows lots of interactions between the user and the player through mouse and keyboard commands. For example, different audio channels can be selected during a presentation, multiple videos can be chosen from a list to be reproduced or a document can be viewed by clicking on a hyperlink. In SMIL, <href> and <area> have been defined for the entire and portion visual display of media objects respectively.

2.5 Playing and Streaming SMIL

A number of commercial and free players can play SMIL presentations. Some internet browsers also support rendering of SMIL presentations. Also, the third Generation Partnership Program (3GPP)¹ has defined a SMIL profile that allows the reproduction of SMIL in mobile devices, centering the scope on multimedia messaging (MMS) and the enhanced packed switched streaming (e-PSS).

There are two ways of reading the data from the sources in a SMIL player. The first one is by downloading the complete file that is included into the

¹ <http://www.3gpp.org>

presentation and to play it from the rendering device. The other solution is streaming the media sources. The streaming of SMIL is very player and codec dependent. The SMIL document must first be sent to the user and the player will then locate the sources from the corresponding streaming servers.

2.6 SMIL Tools

There are many applications and systems developed for SMIL especially the authoring tools to aid the user for multimedia production. Among them, GRiNS[8] and LimSee[9] are two of the most powerful SMIL editors with friendly graphical user interface designed to ease the manipulation of time-based multimedia scenarios and increase the efficiency of production.

In terms of players, several media players support SMIL format such as QuickTime, Real Player, etc. Among them, Ambulant SMIL player [10] is the few players with an open-source license. It contains a basic player UI framework and implements nearly the entire SMIL 2.1 Language profile. It also implements the SMIL 2.1 Mobile and Extended Mobile profiles. For more information about Ambulant SMIL player, please refer to its website [10]. Figure 1 demonstrates the composition of Ambulant player. From this diagram, it is beneficial for researchers and developers to use it because we can support a given format by introducing the corresponding format player.

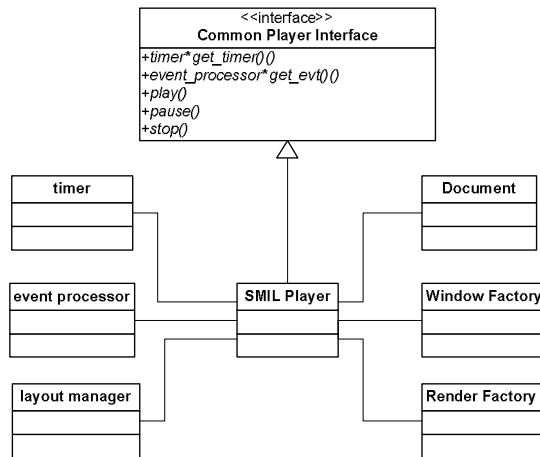


Figure 1: Ambulant SMIL Player Structure

3. MPEG-4 Standard

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group), the committee that also developed the Emmy Award

winning standards known as MPEG-1[11] and MPEG-2[12]. These standards made interactive video on CD-ROM, DVD and Digital Television possible.

MPEG-4 builds on the proven success of three fields: 1) Digital television; 2) Interactive graphics applications; 3) Interactive multimedia (World Wide Web, distribution of and access to content). MPEG-4 provides the standardized technological elements enabling the integration of the production, distribution and content access paradigms of the three fields [13].

3.1 Main Applications

MPEG-4 is designed to address the requirement of new generation of highly interactive multimedia applications, while simultaneously maintaining the support of traditional applications. In addition to efficient coding, MPEG-4 applications also enable more advanced functionalities, such as interactivity with individual media objects, scalability of contents and a high degree of error resilience. It is designed to facilitate the interoperability of multimedia terminals, products or services [13].

In MPEG-4 system, the individually coded audiovisual objects are multiplexed from storage or transmission medium, together with scene description information that describes how these objects should be combined in space and time. MPEG-4's Binary Format for Scenes (BIFS), extended from the Virtual Reality Modeling Language (VRML) [14], specifies the scene compositions and multiplexed media audiovisual objects by a compressor based on arithmetic coders.

3.2 BIFS-Binary Format for Scenes of MPEG4

BIFS stands for Binary Format for Scenes. It provides a complete framework for the presentation engine of MPEG-4 terminals. It has been described in the MPEG-4 standard part 11 [15] and at first has been designed as an extension of the VRML 2.0 specification. BIFS enables to mix various audiovisual media together with 2D and 3D animations and synthetic sounds and deals with the local or remote changes of the scene over time [15]. In contrast to the previous MPEG audiovisual standards 1 and 2 ([11], [12]), within BIFS, a multi-source scene can be rendered at real time on a player, while the audiovisual objects are generated, encoded and transmitted separately.

3.3 XMT-Textual Description of MPEG4

BIFS is a binary format, which means that it requires a compiler to translate a textual description of

the scene into its binary representation. Such textual description of MPEG-4, XMT, illustrates a high level view of MPEG-4. XMT has been proposed to describe an MPEG-4 scene using a textual syntax. Like SMIL, XMT is XML-based, relatively easy for authoring and provides high level constructs. XMT is designed with a two-tier architecture: XMT-O and XMT-A. XMT-O provides a high-level abstraction of MPEG-4 functionality, while XMT-A, provides a one-to-one deterministic mapping to MPEG-4 binary representation (BIFS).

XMT-O is the high-level version of XMT, orientated towards an easy creation of scenes and interchange between authors and authoring tools by using abstractions about the timing, media and animation manipulations. It has been developed between W3C and the MPEG Consortium and it is close to SMIL 2.0 (it uses most of the SMIL 2.0 modules) but it is dedicated to the MPEG-4 reproduction. It makes easy the interchange between SMIL players, VRML players and, of course, MPEG-4 players with simple transformations or compilations to binary code.

XMT-A is the low-level version of XMT, intended to comprise the full scope of BIFS. XMT-A is an exact transcription of the BIFS bit stream. As XMT-O is an abstraction and does not have a direct mapping to BIFS, it is not possible to control all the existing elements and functionalities of MPEG-4, but it is allowed to include XMT-A descriptions into an XMT-O document in order to manage the full scope of MPEG-4.

3.4 Structure and Layout

An MPEG-4 textual scene description consists of three parts: (1) VRML hierarchy structure which defines the media objects, the combinations and the timing and spatial relationships among them; (2) Interactive behaviors, animation and transition effects; (3) BIFS update commands. Metadata and elementary stream level information (e.g. buffer sizes etc.) are stored in object descriptors and elementary stream descriptors. Unlike SMIL, the dimension of an MPEG-4 window is defined in the BIFS configuration which is carried in one of the object descriptors, separate from the textual scene description. This allows the window size to be changed with the Update Commands. A drawback is that the content authors will have to get used to defining the window's size not in the scene description, but in a separate object descriptors file [16].

3.5 Timing and Synchronization

The timing model in MPEG-4 requires transmitted data streams to contain temporal information such as Decoding Time Stamp (DTS) and Composition Time Stamp (CTS) which determine when data should be available to be decoded and composite respectively. These timing stamps are measured with respect to the Object Time Base (OTB) which can be reconstructed either from Object Clock Reference (OCR) inserted in the stream or by an indication that it is slaved to a time base conveyed with another stream[13].

This is different from SMIL, since SMIL does not rely on the temporal information carried on the stream. Therefore, the flexible timing model like SMIL has been proposed to incorporate to the current MPEG-4 standard.

BIFS provides some facilities to schedule the similar SMIL playback in MPEG-4. BIFS Update Commands enable to insert the MPEG-4 nodes corresponding to the objects into the scene at the specified begin times, and delete them from the scene after the indicated time duration.

```
<par>
  
  <audio id="test audio", src="...", begin="5s", dur="20">
  <seq>
    <video id="video1", src="...",begin="40s", dur="20">
    <video id="video2", src="...",begin="10s", dur="30">
  </seq>
</par>
```

For instance, the above simple sample of SMIL could be simulated by BIFS update protocols as follows:

```
At 5s, insert image node "image" and audio node "test audio";
At 15s, delete image node "image";
At 25s, delete image node "test audio";
At 40s, insert video node "video1" and insert Media Control node;
At 60s, the ROUTE command route the output of Media Control
node to a conditional node which contains the INSERT command to
insert the video node "video2";
At 100s, delete video node "video2".
```

3.6 Supported Formats

To create an MPEG-4 media file, audio/video formats covered by the MPEG-4 Standard should be preferably used. For video, this includes among others formats such as MPEG-4 SP/ASP, MPEG-4 AVC/H.264, MPEG-2, MPEG-1, DivX5, DivX4/OpenDivX etc. For audio, we can cite AAC, MP3, MP2, MP1, CELP (for speech), SAOL (midi) etc. For subtitle and text MPEG-4 Timed Text should be used while JPEG or PNG are the main supported picture formats.

3.7 User Interaction

Two kinds of interactions between the user and the scene are allowed by BIFS: client-side interaction and server-side interactions. The content developers limit both of these interactions.

The client-side interaction is related to the changes in the presentation made in the receiver. For example, one can drag scene objects or read extra information about them, choose among different sounds, change their positions or change the point of view of the user.

The server-side interactions need a back channel since the content modifications take place at the transmitting side.

Interactions between the user and the presentation are controlled via the mouse and the keyboard.

3.8 Playing and Streaming MPEG-4

MPEG-4 is a versatile and flexible format. Due to its double nature (static file or stream), it can be used in various applications such as Broadcast television (cable and satellite), video streaming to wireless devices, Internet, ATM, CD and DVD.

A number of MPEG-4 players exist and since MPEG-4 can be streamed into MPEG-2, it can be compatible with MPEG-2 environments.

A BIFS document and all its sources may be included in a single .mp4 file or it can be streamed from a server; in this case it is not needed to download the complete BIFS document to render the scene but an entire part of a scene has to be downloaded before being displayed.

BIFS is mainly dedicated to MPEG-4 broadcast applications such as TV since it can be streamed. A very low bit rate can be obtained for BIFS presentations since it includes an efficient compression scheme of the scene description, which is usually smaller than other media (the scene description is only text data) while audio and video are compressed as well.

The MPEG-4 stream can be transmitted over many different transport protocols. For example, it can be streamed over the MPEG-2 transport layer to be used into existing MPEG-2 environments or, of course, over IP including RTP, RTSP, and HTTP, among others. MPEG has also defined DMIF (Delivery Multimedia Integration Framework) [17], a session protocol similar to FTP for the management of multimedia streaming over generic delivery technologies.

3.9 MPEG-4 Tools

There are many kinds of applications developed to support MPEG-4 standard extending from MP4 encoders and players to streaming systems [18]. However, there are few powerful MPEG-4 authoring tools, especially compared with the versatility and quality for SMIL.

4. SMIL to BIFS conversion

Based on the above comparative analysis, we propose SMIL to BIFS conversion to maximize the advantage of both formats. The main benefit of such conversion is incorporating the current SMIL applications to MPEG-4 applications and vice versa.

As a binary-coded format, BIFS is difficult to edit and subsequently hard to use because the binary coded representation often cannot be "reverse-engineered" in a consistent manner to represent the content author's original intentions. At the same time, friendly graphical environment for multimedia authoring applications are inherently difficult to build. This is mainly due to the fact that graphical user interface (GUI) software is often large, complex and time-consuming to develop and modify. One survey demonstrated that around 48% of the applications' code is devoted to the user interface, and that about 50% of the implementation time is devoted to the user interface portion [19]. To date, there are few powerful MPEG-4 authoring tools available. The Composite Media Group from IBM Company develops an XMT authoring tool XMTEdit [20] for MPEG-4 authoring.

Meanwhile, there are several powerful GUI SMIL authoring tools developed to simplify the authoring process. For instance, SMIL authoring tools GRiNS [8], and LimSee [9] are good delegates among them. Consequently, to take advantage of SMIL content in a larger range of applications or simply to benefit from existing SMIL authoring tools to develop BIFS content, it seems interesting to convert SMIL presentations to BIFS format of MPEG-4. We can efficiently utilize the SMIL authoring tools to commit the authoring functions for the MPEG-4 media objects.

Then we could use a SMIL player to directly preview the result for the MPEG-4 authoring instead of transcoding with at least three steps from XMT to BIFS and then to MP4. This is the advantage over any XMT authoring tool. From the Figure 1 in section 2, we also show the extensibility of the open source tool Ambulant SMIL player by diagram. This player is extensible to support more media format if the source code of such given media player is available.

As it has been stated earlier, XMT-O is based on SMIL 2.0 and consequently most features from SMIL 2.0 are included into XMT-O. Those SMIL features

that are not supported by XMT-O have a very similar structure anyway. To convert SMIL documents into BIFS scenes, it is reasonable to first translate SMIL into XMT-O due to the proximity of these formats. Once this first step is completed, the resulting XMT-O document can be converted into its XMT-A equivalently before a final compilation of the XMT-A file into BIFS. Figure 2 shows a graphical representation of the SMIL to BIFS conversion process.

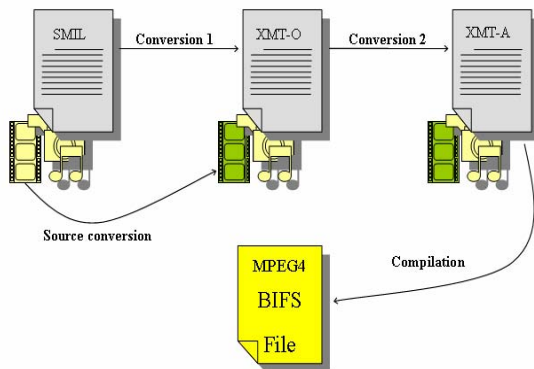


Figure 2: SMIL to BIFS conversion steps

4.1 Conversion from SMIL to XMT-O

As described in section 2, SMIL functionalities are divided into ten areas further partitioned into modules. Each module has a set of tags, properties and attributes. Some of them must be used independently while others may be combined.

Some SMIL 2.0 modules are syntactically and semantically the same as the XMT-O modules and can be directly used in the XMT-O file (namely the **Animation, Content Control, Media Object, Metainformation, and Time Manipulations** modules). In the **Structure** module, the `<smil>` tag must be replaced by the `<xmt-o>` tag.

The **Timing, Layout, Linking and Transition** features are only partially compatible with XMT-O. For those SMIL behaviours that have no correspondances with the XMT-O utilities, XMT-A commands must be used and included into the XMT-O document. Indeed, XMT-O is only an abstraction of BIFS and does not allow controlling of all the functionalities of MPEG-4. On the contrary, XMT-A manages the full scope of MPEG-4 and can be included in XMT-O documents so that using this mechanism most of the SMIL subtilities can be accurately translated to MPEG-4.

As it has been stated earlier, XMT-O is based on SMIL 2.0 while the current specification of SMIL is

2.1. Though some of the newer tags of SMIL 2.1 can be mapped to corresponding SMIL 2.0 tags, the new functionalities of SMIL 2.1 should also be handled by directly producing XMT-A commands.

This is the only stage for which there is no software already available. We build an application with xerces-c++ library [21] to implement this stage for the conversion. We read the SMIL with a DOM parser into a type of DOMDocument and analyze each node based on the similarities and difference between SMIL and XMT. For the same things, such as the layout and object name, we keep them unchanged. In case of difference, we utilize the correspondent parts of XMT-O. We test with the basic SMIL modules which include the main functionalities of SMIL.

4.2 Source Conversion

Since MPEG-4 puts some restrictions on formats for sources (see section 3.6), the transcoding of some of the sources used by the SMIL file will be needed. Moreover, SMIL supports links to web pages and monitoring of some non-audiovisual documents while this is not the case for BIFS. Such links should be skipped when converting towards BIFS.

4.3 Conversion from XMT-O to XMT-A

The mapping between XMT-O and XMT-A has not a single way to be performed. In other words, a high-level (abstract) instruction can be mapped to several different groups of low-level instructions so that the mapping from the high-level to the low-level language can be done according to the possibilities of the player or the rendering device. For more details about the conversion from XMT-O to XMT-A, please refer to the MPEG reference software.

4.4 Compilation: XMT-A to BIFS

Once a valid XMT-A file is produced, the binarization into BIFS is performed thanks to a compiler. The software mp4box claims to do the conversion. However, after testing, we find it does not give the proper result. Alternately, we use the style sheets developed by Lyndon J. B. Nixon. The relevant files can be found at the MPEG reference software website.

4.5 BIFS to MPEG-4

At this final stage, we could use bSoft or mp4box to generate mp4 files out of BIFS. Then we use any

MP4 player, such as OSMO MP4 player to review the result.

5. Conclusion and Prospective

In this paper, we have provided a comparative analysis between SMIL and MPEG-4 and have described a conversion scheme from the SMIL to the BIFS.

Converting SMIL to BIFS allows taking advantage of SMIL authoring tools and existing SMIL content in applications such as broadcasting and video on demand in which BIFS seems more appropriate. SMIL content is indeed mainly dedicated to Web applications while, at the time of publication of this paper, the first set-top box supporting MPEG-4 has been built by Tandberg TV while other companies are preparing the commercialization of their own solutions. Moreover, MPEG-4 content can be used in the already existing MPEG-2 networks by achieving some firmware and hardware changes.

Based on the scheme, we build an application to convert SMIL to XMT-O with basic functions. However, this part of work is still under development. At the same time, though it is possible to convert accurately a SMIL presentation into a BIFS file, there is no unique mapping from BIFS to SMIL, therefore converting a BIFS file into a SMIL presentation is much more difficult than its reciprocal. How to makeup the gap between SMIL to MPEG-4, two interactive media, still needs investigation.

6. Acknowledgements

The authors would like to thank the EC IST FP6 for the partial funding of the AXMEDIS project (<http://www.axmedis.org>) and to express gratitude to all AXMEDIS project partners internal and external to the project including the Expert User Group and all affiliated members for their interests, support and collaboration.

10. References

- [1] Dick C.A. Bulterman and Lynda Hardman. "Structured Multimedia Authoring" In ACM Trans. on Multimedia Computing, Communications and Applications, 1(1) 2005.
- [2] B. Rogge, J. Bekaert, R. Van de Walle, "Timing issues in Multimedia Formats: Review of the Principles and Comparison of Existing Formats", in IEEE Transaction on Multimedia, vol. 6, no. 6, December 2004

- [3] ISO/IEC 14496-13: Intellectual Property Management and Protection (IPMP) Extensions.
- [4] Synchronized Multimedia Integration Language (SMIL 2.1) W3C Recommendation, 13 December 2005. <http://www.w3.org/TR/2005/REC-SMIL2-20051213/>
- [5] W3C Recommendation, <http://www.w3.org/TR/SVG>. "Scalable Vector Graphics (SVG) 1.0 Specification", September 2001.
- [6] S.DeRose, E.Maler, D.Orchard and B.Traford. "XML Linking Language (XLink)" W3C Recommendation, <http://www.w3.org/TR/xlink>, June 2001.
- [7] O.Lassila and Ralph R. Swick. "Resource Description Framework (RDF) Model and Syntax Specification". <http://www.w3.org/TR/REC-rdf-syntax>, February 1999.
- [8] <http://www.oratrix.com/GRI/NS/>
- [9] <http://wam.inrialpes.fr/software/limsee2/>
- [10] <http://www.cwi.nl/projects/Ambulant/distPlayer>
- [11] ISO/IEC 11172: MPEG-1 standard.
- [12] ISO/IEC 13818: MPEG-2 standard.
- [13] R. Koenen, editor. MPEG-4 Overview, ISO/IEC JTC1/SC29/WG11 N2564, Rome, December 1998.
- [14] ISO/IEC WD 14772: "The Virtual Reality Modeling Language Specification, Version 2.0", August 4, 1996.
- [15] ISO/IEC 14496-11: "Scene description and Application engine", December 2005.
- [16] Lai-Tee Cheok, Alexandros Eleftheriadis, "SMIL VS MPEG-4 BIFS" technical report, CU/EE/TR 2002-02-111, Columbia University.
- [17] ISO/IEC 14496-6: "Delivery Multimedia Integration Framework (DMIF)", December 2006.
- [18] B. Lehane, N. O'Connor, N. Murphy, "MPEG-4 Tools and Applications: An Overview", 2003.
- [19] B.A. Myers, M.B. Rosson. "Survey on User Interface Programming." In Proceedings SIGCHI'92: Human Factors in Computing Systems. 1992. Monterey, CA: pp. 195-202.
- [20] <http://www.research.ibm.com/mpeg4/Projects/>
- [21] <http://xml.apache.org/xerces-c/>