THE MPEG-4 STRUCTURED AUDIO STANDARD

Eric D. Scheirer

E15-401D MIT Media Laboratory
Cambridge MA 02139
eds@media.mit.edu

ABSTRACT

The MPEG-4 standard defines numerous tools that represent the state-of-the-art in representation, transmission, and decoding of multimedia data. Among these is a new type of audio standard, termed "Structured Audio". The MPEG-4 standard for structured audio allows for the efficient, flexible description of synthetic music and sound effects, and the use of synthetic sound in synchronization with natural sound in interactive multimedia scenes. A discussion of the capabilities, technological underpinnings, and application of MPEG-4 Structured Audio is presented.

1. INTRODUCTION

ISO/IEC SC29/WG11—better known as the Moving Pictures Expert Group or "MPEG"—began a new work item in 1995 to standardize low-bit-rate coding tools for the Internet and other bandwidth-restricted delivery channels. This work item, now known as MPEG-4, will become an international standard in January 1999 as ISO 14496. However, since its inception, the focus of MPEG-4 has expanded; it now includes not only traditional coding methods optimized for low-bit-rate transmission, but also highly novel coders which allow the description of synthetic content, audiovisual scenes, and the synchronization and spatiotemporal combination of synthetic and natural content.

Among these new tools is an audio coder called "Structured Audio" (strictly, ISO 14496-3 subpart 5 [4]). Structured coding schemes have been described as data formats which make use of low-dimensional, semantic, and/or model-based representations of multimedia content [10]. The MPEG-4 Structured Audio standard has been created by MPEG to allow for standardized description of synthetic sounds, and guaranteed reproduction performance on all types of terminals.

Using the methods standardized in MPEG-4, soundtracks may be created, transmitted using very low bandwidth, and yet rendered on a terminal with very high quality—in many cases, higher quality than would be possible using other coding schemes, even at much higher bandwidth. Such synthetic soundtracks have application in arenas as varied as Internet karaoke, virtual gaming, multimedia presentation, and other types of content yet to be invented.

1.1. Why standardize synthetic sound?

There are a number of requirements which are satisfied by the MPEG-4 Structured Audio toolset. Within the MPEG-4 workplan, synthetic sound coding represents a extremely flexible tool, supporting interactive functionality that is not possible with other coders. Also, in many cases, structured audio bitstreams are much more compact than equivalent naturally-coded versions of the same sounds.

Going outside the specific goals of MPEG, the technology created in the MPEG-4 working process potentially represents a large step forward in the sophistication of PC-based sound synthesis devices. Other researchers, not just those participating in MPEG, feel that certain aspects of the commercial computer music industry have reached a bottleneck:

> MIDI was designed to mechanize performance on a keyboard-controlled synthesizer. It was not designed to serve as an interchange format for computer music...Commercial synthesizers and MIDI have sent us back to the "Note Age" by placing a wall between the instrument and the note that is played on it...MIDI synthesizers offer only a tiny subset of the synthesis techniques possible in software. (J. O. Smith, [8])

Injecting advanced technology into the marketplace in the form of an open standard with sufficiently broad application may help to promote a continuing evolution of commercial music-synthesis devices.

It is important to realize that there is no "inherent" sound quality created with synthetic sound. Much, perhaps most, of the music we hear today in movies, television, and interactive media is already created synthetically; in many cases, the quality of the synthesis is high enough that we are unaware or unconcerned about its origin. By ensuring through the standard that normative, high-quality synthesis can be performed on every MPEG-4 terminal, on-the-fly synthetic sound will become a viable alternative to recorded audio in multimedia applications.

In this paper, details of the MPEG-4 Structured Audio standard will be discussed, including the following topics: the components of the toolset, the capabilities provided by these components, the technical bases of the tools, the "profiles" or restricted subsets of the tools for use in certain applications, and current perspectives on implementing the standard.

It is important to note that, as of the time of this writing, the standard has just reached "Committee Draft" form. Although the major technical components are fixed and well-evolved, editorial or minor technical improvements may be made which invalidate particular statements herein. This paper does not represent a finished standard, but the viewpoints of one researcher involved in the ongoing standards process. Comments and critiques which help to improve the standard before finalization are encouraged at all times.

2. THE STRUCTURED AUDIO TOOLSET

The MPEG-4 Structured Audio toolset is based around a software synthesizer-description language. The technical basis of such a description is familiar to computer musicians, as it is similar to previous music languages such as Music V and Csound [9]. In fact, a previous (non-standardized) structured audio experiment called "NetSound" [2] made use of Csound to perform the synthesis. For MPEG-4 certain changes have been made to this framework. As with all MPEG standards, a sophisticated bitstream compression method is standardized which allows more efficient transmission; also, like other components of MPEG-4, the Structured Audio toolset allows streaming transmission of data, a feature not typically utilized in computer-music composition.

There are five major elements to the Structured Audio toolset. Each of them will be described briefly (the brief descriptions are taken, in slightly modified form from [4]), and a discussion of the manner in which they are unified in the overall decoding framework follows.

2.1. Components

The *Structured Audio Orchestra Language*, or *SAOL*, is the synthesis-description language at the heart of the standard. SAOL is a digital-signal-processing language which allows for the transmission of arbitrary synthesis and effects algorithms as part of the content bitstream. The syntax and semantics of SAOL are standardized as part of MPEG-4.

The *Structured Audio Score Language*, or *SASL*, is a simple score and control language used to describe the manner in which the sound-generation algorithms transmitted in SAOL are used to produce sound.

The *Structured Audio Sample Bank Format*, or *SASBF*, allows for the transmission of banks of audio samples to be used in wavetable synthesis, and the description of simple processing algorithms to be used with them.

A normative scheduler, the run-time element of the Structured Audio decoding process, is described. It maps structured sound control, specified in SASL or MIDI, into the dispatch of real-time events, with processing specified using the normative sound-generation algorithms described in SAOL.

Normative reference is made to several MIDI standards, standardized externally by the MIDI Manufacturers Association. MIDI is an alternate means of structural control which can be used in addition to or instead of SASL. Although less powerful and flexible than SASL, MIDI support in the standard provides important backward-compatibility with existing content and authoring tools. For some MIDI commands, additional semantics are defined in MPEG-4 to more logically integrate MIDI with the rest of the Structured Audio tools.

2.2. Decoding framework

As mentioned above, the technologies used in MPEG-4 Structured Audio are similar to those in existing music-synthesis languages. Thus, except for the bitstream-related requirements, readers familiar with computer-music languages such as Csound will quickly grasp the concepts here. The Structured Audio toolset does not standardize a "synthesis method", but a special language (SAOL) for describing synthesis methods. Any current known sound-synthesis method may be described in SAOL, which, may be thought of as a language for concisely describing signal-flow networks. Thus, any digital-signal-processing process which can be expressed as a signal flow network can be expressed in SAOL.

Although the concepts in SAOL are similar to those in Csound, SAOL is a completely new language developed by MPEG. Compared to Csound, it features improved syntax, a smaller set of core functions, and a number of additional syntactic features which make the authoring of certain synthesis algorithms much easier.

Decoding of the Structured Audio bitstream makes use of a reconfigurable synthesis engine, see Figure 1. This synthesis engine may be implemented in hardware or software, although it is difficult to envision solutions that make use only of fixed hardware without programmability. The exact capabilities required of this engine are fixed in the standard.

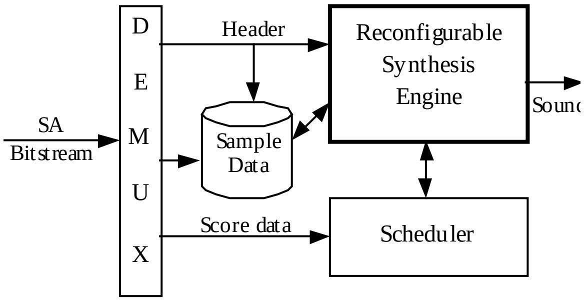

Figure 1. Overview of the decoding process. See text for details.

When a session is started, the MPEG-4 bitstream is demultiplexed into several component content streams (video, audio, graphics, synthetic sound, etc; this step is not shown in Fig. 1). The Structured Audio bitstream is further demultiplexed into several pieces: the *bitstream header*, *sample data*, and *score data* (which takes either MIDI or SASL form).

The bitstream header contains a description of several synthesizers, as well as effects-processors (for example, reverberators), control algorithms, and routing instructions to describe audio flow-of-control during the synthesis process. This description is tokenized for efficient transmission. When this header is decoded, the synthesis engine is reconfigured as described by the SAOL code, thus preparing it for use in the run-time decoding process. The reconfiguration step is in many ways similar to parsing and compiling a high-level computer language; similar skills in software engineering and computer science are required to implement a SAOL system as needed to implement a high-level language compiler.

SAOL is standardized in terms of approximately 100 primitive processing instructions, signal generators, and operators which fill wavetables with data. Thus, a typical header decoding process might also include a step that resembles linking a high-level language with a fixed library of abstract functions. Since most of the complexity in SAOL is encapsulated in these primitive functions, they may be carefully optimized for each particular implementation.

The bitstream header may also contain audio samples (blocks of floating-point data). This data is typically used in *wavetable synthesis* [1], but may be used for any other purpose in SAOL in addition or instead. For example, a block of sample data might contain an impulse response that creates a particular reverberation effect when it is convolved with target sounds (FIR filtering is one of the standard fundamental operations of SAOL). Blocks of sample data, carefully configured for optimum wavetable synthesis, may be transmitted in the MPEG-4 Structured Audio Sample Bank Format, or SASBF. A SASBF bitstream element contains several samples, simple parameters for algorithmic modification, and directions showing how they are to be configured for a particular synthesis session.

Once the header has been received, understood, and used to reconfigure the synthesis engine, the streaming bitstream data is used to control the synthesis process. After the reconfiguration step, the synthesis engine acts roughly like a set of fixed synthesizers in a traditional sound-studio. That is, it receives commands and turns them into sound.

The commands for controlling the synthesizers are created by the *scheduler*, which is the main control process of the Structured Audio system. The scheduler parses and understands the control information in the bitstream header and streaming data, determines at what time commands must be dispatched to the synthesizers, and communicates with the MPEG-4 Systems layer. Like a traditional synthesizer, the Structured Audio system can process commands in the MIDI protocol; they are internally converted into native events by the scheduler. The syntax of these commands is the same as that standardized in the relevant MIDI documentation [5], but certain semantics are changed to enable tight normative control over the synthesis process. In addition, a second control language, SASL, is standardized within MPEG-4. SASL is more flexible than MIDI and allows more complex control functions and mappings to be specified than are possible with MIDI.

In some computer-music systems today, scoring languages have become quite complex, featuring commands enabling subtle control over phrasing, sectional structures of music, and even mixing and programmatic control. The philosophy in the MPEG-4 design process has been the opposite. Since most content authors use special tools (sequencers, patch design interfaces) to create content, the desirable trait is for the control language to be simple and easily created automatically. Thus, although SASL allows for a great deal of flexibility in authoring, it is very simple and does not provide high-level control functions. This also allows easier transcoding of the score data between SASL and other formats.

2.3. Interaction with Natural Audio

Besides the Structured Audio toolset (and the other synthetic audio component, the Text-to-Speech interface), MPEG-4 standardizes state-of-the-art versions of several traditional audio formats, called “natural” content coders in the MPEG framework [6]. A great deal of the flexibility in MPEG-4 comes from the juxtaposition and interface of these coders with the synthetic coders; *Synthetic/Natural Hybrid Coding*, or SNHC, is an important concept in both the audio and visual toolsets of MPEG-4. For example, a soundtrack may be composed of two separate *audio objects*; the voice track is coded using the CELP low-bit-rate speech coder, and the background music is coded synthetically, using structured audio. At the decoding terminal, the two components are decoded and mixed together. This mixing process is described and carried out to sub-sample accuracy.

The mixing process is specified using the MPEG-4 Binary Format for Scene Description, or BIFS. In general, BIFS is similar in concept to the virtual-reality description language VRML, but the audio component in particular has been expanded in functionality. BIFS is standardized as part of the MPEG-4 Systems [3] toolset; audio experts have been involved in developing the audio aspects of this tool. Using audio BIFS, sound sources may be mixed, grouped, and delayed, processed with 3-D virtual spatialization, and post-processed using signal-manipulation functions transmitted in SAOL as part of the content bitstream

This last feature is a novel aspect of MPEG-4. Rather than standardize a fixed set of post-production techniques—reverberators, chorusers, etc—with a fixed set of parameters, in MPEG-4 the content author may freely design post-processing algorithms for inclusion in the bitstream. This has two major advantages: the content author has complete control over the exact characteristics of the algorithms used, and the algorithm set is expandable as needed for a particular piece of content.

The synthetic post-production capability applies not only to synthetic sound coded with the Structured Audio toolset, but to natural sound coded with other tools. It is easy to see how this leads to both bandwidth reduction and improved content quality. Very-low-bit-rate CELP, for example, makes use of a fixed speech model which works well for clean speech sound, but degrades quickly in the presence of noise, reverberation, or other artifacts. However, transmission of a reverberation algorithm written in SAOL is very inexpensive, perhaps no more than 200 bytes in the header. Thus, reverberated speech, which often sounds more natural in a soundtrack than flat speech, may be transmitted by coding flat speech with the low-bit-rate CELP coder, and post-processing it on the client side with a synthetic reverb. Such a sound only requires as much bandwidth as is needed for the flat speech, plus a tiny overhead for the reverberator.

These concepts, of juxtaposing real and synthetic sounds and mixing and combining them with synthetic post-production, are explored in more depth in [7].

3. PROFILES AND APPLICATIONS

As in previous MPEG standards, MPEG-4 describes the definition of several profiles which allow the implementation of terminals compliant to a subset of the full standard. There are three restricted profiles of the full MPEG-4 Structured Audio standard, each targeted at certain applications. Within each profile, only a certain subset of bitstream data items may appear.

Profile 1 bitstreams contain only MIDI data, and the semantics are as standardized by the MMA. This profile does not specify the exact sound quality to be produced; and so, in many terminals, will not produce high-quality sound. This profile is highly interoperable with existing hardware. It is suited to applications, such as internet karaoke, where MPEG-4 functionality such as synchronization of graphics, text, and voice is needed, but the quality of the music synthesis is not crucial.

Profile 2 bitstreams may not contain SAOL or SASL data; only the MIDI and SASBF data are used to generate sound. This profile is appropriate for musical applications where great synthesis flexibility is not needed, and where sufficient bandwidth is available for transfer of samples. Normative sound quality is produced in this profile. The main advantage of this profile is that the synthesis computation may utilize fixed, rather than programmable, hardware.

Profile 3 bitstreams may not contain SASBF or other sample data. This profile is appropriate when the bandwidth is very restricted, particularly in broadcast applications which require continuous re-broadcast of bitstream headers. All synthesis is purely algorithmic, and the sound quality is strictly normative.

Profile 4 is the full, default profile of MPEG-4 Structured Audio. All bitstream components may occur; the sound synthesis is strictly normative.

The advanced sound post-production capability described in the previous section is only available when Structured Audio Profile 3 or Profile 4 is supported by the terminal.

Using the various profiles of MPEG-4 Structured Audio, a number of applications are enabled, including the following. Low-bit-rate transmission of new music, especially modern dance music which relies heavily on electronic sounds, is perfectly suited to MPEG-4. Internet karaoke, including the synchronization of moving pictures, text, and natural voices with the musical background, is a core application. Virtual-reality environments, such as interactive music applications or "fly-through compositions", are easily described using Structured Audio and Audio BIFS.

In other content domains, the structured soundtrack capability (using BIFS) makes possible the description of interactive content, from interaction as simple as selecting the language of dialogue in a film, to complex re-mixing and co-authoring environments. The SAOL language is also well-suited to the needs of academic and avant-garde composers, and will provide a excellent platform for this genre of music.

Finally, as the implications of MPEG-4 as a whole are explored more fully by content producers, creative artists, and implementation designers, new forms of content may well arise which take advantage of this unique description format.

4. CONCLUSION

The MPEG-4 Structured Audio toolset has been described, and technical details about its decoding model, component formats, and applications presented. A free, public implementation and current details on the progress of the standard are available from the Structured Audio homepage, found on the World Wide Web at `http://sound.media.mit.edu/~eds/mpeg4/`.

Many new capabilities are ushered into the standards arena with the synthetic content and scene-description capabilities in MPEG-4. MPEG-4 Audio, in particular, represents the first explicit connection between the well-developed but independent bodies of work in the computer-music, synthetic speech, and audio-coding communities. All of the MPEG contributors and participants hope and feel that MPEG-4 will represent a landmark in the history of standardization activity.

Acknowledgements

The work described here had many contributors, including but not limited to: Itaru Kaneko, Ron Burns, Don Mead, Brian Link, Lee Ray, Luke Dahl, Adam Lindsay, Shigeki Fujii, Giorgio Zoia, and Jyri Huopaniemi. Thanks to the convener of MPEG, Leonardo Chiariglione, and the Audio Chair, Peter Schreiner, for their expert supervision. This work bears a strong debt to the ealier "NetSound" system, built by Michael Casey, Barry Vercoe, Paris Smaragdis, Adam Lindsay, and the author. The Machine Listening Group at the MIT Media Lab, under the direction of Barry Vercoe, was (as always) essential in the support of this paper through comments and criticism. Finally, thanks to Joern Ostermann for organizing the special session on new multimedia standards.

5. REFERENCES

[1] R. Bristow-Johnson. "Wavetable synthesis 101: a fundamental perspective", *Proc AES 101st*, 1996 (AES Reprint #4400).

[2] M. A. Casey and P. Smaragdis, "Netsound". *Proc Int. Computer Music Conf.,* Hong Kong, 1996.

[3] A. Eleftheriadis, C. Herpel, G. Rajan, L. Ward (Eds). *ISO 14496-1 (MPEG-4 Systems) Committee Draft.* MPEG document N1901, Fribourg CH, 1997.

[4] B. Grill, B. Edler, I. Kaneko, Y. Lee, M. Nishiguchi, E. D. Scheirer, and M. Väänänen (Eds). *ISO 14496-3 (MPEG-4 Audio) Committee Draft.* MPEG document N1903, Fribourg CH, 1997.

[5] MIDI Manufacturers Association. *The Complete MIDI 1.0 Detailed Specification* v. 96.2, 1996.

[6] S. Quackenbush, "Coding of natural audio in MPEG-4". *Proc IEEE ICASSP*, Seattle, 1998 (this volume).

[7] E. D. Scheirer, "Structured audio and effects processing in the MPEG-4 multimedia standard", *ACM Multimedia Systems*, in press.

[8] J. O. Smith III, "Viewpoints on the history of digital synthesis", Keynote paper, *Proc Int. Computer Music Conf*, Montreal, 1991.

[9] B. L. Vercoe. *Csound: a manual for the audio-processing system*. MIT Media Lab, 1995.

[10] B. L. Vercoe, W. G. Gardner and E. D. Scheirer. "Structured Audio: Creation, transmission, and rendering of parametric sound representations," *Proc. IEEE*, in press.