Symbolic Music Representation in MPEG

Pierfrancesco Bellini and Paolo Nesi
University of Florence

Giorgio Zoia
École Polytechnique Fédérale de Lausanne

With the spread of computer technology into the artistic fields, new application scenarios for computer-based applications of symbolic music representation (SMR) have been identified. The integration of SMR in a versatile multimedia framework such as MPEG will enable the development of a huge number of new applications in the entertainment, education, and information delivery domains.

Music is a complex piece of information. Although it's usually considered in its audible representation, many notations have attempted to represent visually or by other means the information needed to produce music as the author composed it. Music notations aren't intended exclusively for performers, however; much like the music they convey—and especially 20th-century music—musical scores are often considered art masterpieces in their own right.

A symbolic music representation (SMR) is a logical structure based on symbolic elements representing audiovisual events, the relationship between those events, and aspects related to how those events can be rendered. Many symbolic representations of music exist, including different styles of chant, classic, jazz, and 20th-century styles; percussion notation; and simplified notations for children and vision-impaired readers.

Information technology's evolution has more recently changed the practical use of music representation and notation, transforming them from a simple visual coding model for sheet music into a tool for modeling music in computer programs and electronic devices in general. Consequently, we currently use SMR or music notation for purposes other than producing sheet music and teaching—for example, for audio rendering, entertainment, music analysis, database querying, and performance coding.

The integration of SMR in versatile multimedia frameworks such as the Moving Pictures Expert Group's MPEG-4 standard, with technologies ranging from video and audio to 3D scenes, interactivity, and digital rights management, will enable the development of many new applications. This will let users load and receive music notation integrated with multimedia in consumer electronic devices, so as to manipulate music representation without infringing on copyright laws.

Multimedia music notation

Not only is music representation changing with respect to its purely music-related use, but associated applications that integrate multimedia and interactivity are rapidly evolving as computer technology spreads into the arts. New scenarios for computer-based applications of SMR (see also the "Current Multimedia Interactive Music Applications" sidebar) include:

- multimedia music for music tutoring systems such as Imutus (http://www.exodus.gr/imutus) and Musicalis (http://www.musicalis.fr);
- multimedia music for "edutainment" and "infotainment" (information and entertainment such as news on new music events that includes some entertainment aspects)—for example, for archives such as Wedelmusic's (http://www.wedelmusic.org) and for theaters such as OpenDrama (http://www.iua.upf.es/mtg/opendrama); and
- cooperative music editing in orchestras and music schools, such as the Moods[2,3] project.

Users have discovered the multimedia experience, and more suitable multimedia representations of music have in many cases replaced the traditional music notation model. The music community and industry need a comprehensive representation of music information integrated with other media. To realize multimedia and interactive applications, we must account for aspects ranging from information modeling to integration of the music representation with other data types. The main problems center around acceptably organizing music semantics and symbols to cope with the more general concepts of music elements and their relationship with audiovisual content.

The capability to represent nonwestern music notations, such as those from Asian countries, the Middle East, and Northern Africa, is also

1070-986X/05/$20.00 © 2005 IEEE

Current Multimedia Interactive Music Applications

New devices, products, and several innovative research and design projects supported by the European Commission, such as the MusicNetwork Center of Excellence (http://www.interactivemusicnetwork.org), are helping to popularize multimedia interactive music. These new applications exploit media integration and distribution via the Internet, satellite data broadcast, and other means to reach a large number of users. Innovative features are beginning to inundate the market, especially in the area of music education. Products include Freehand (see http://www.freehandsystems.com), Yamaha tools (http://www.digitalmusicnotebook.com/home), Sibelius Music Educational Tools (http://www.sibelius.com), and Finale CODA solutions (http://www.finalemusic.com/).

One of the best-known computer-based music applications—notation editing for professional publishing and visualization—is often too focused on the visual rendering of music symbols.[1-3] Similar applications include Sibelius (http://www.sibelius.com), Finale CODA (http://www.finalemusic.com), Capella (http://www.whc.de/capella.cfm), and MidiNotate, (http://www.notation.com).

Music publishers must produce music scores that are high quality in terms of the number of symbols and their placement on the staff. Several Extensible Markup Language (XML)-compliant markup languages for music modeling are available. They include the Musical Notation Markup Language (MNML), MusicML, Music Markup Language (MML), MusicXML,[4] Wedelmusic,[5] and CapXML.[6]

Most focus on modeling the music elements to represent and exchange them among applications. Only a few of the formats can cope with the emerging needs of innovative multimedia music interactive applications, which mostly use proprietary and incompatible technologies that reshape the music content for each tool and have difficulty exchanging (mostly notational) information among tools.

With no standardized formats for the symbolic representation of music (SMR), developers must implement their own solutions, which are highly variable in efficiency, scope, quality, and complexity. Previous attempts to standardize music notation include the Standard Music Description Language (SMDL)[7] and Notation Interchange File Format (NIFF).[8]

References

1. D. Blostein and L. Haken, "Justification of Printed Music," *Comm. ACM*, vol. 34, no. 3, Mar. 1991, pp. 88-99.
2. J.S. Gourlay, "A Language for Music Printing," *Comm. ACM*, vol. 29, no. 5, May 1986, pp. 388-401.
3. B.W. Pennycook, "Computer-Music Interfaces: A Survey," *ACM Computing Surveys*, vol. 17, no. 2, June 1985, pp. 267-289.
4. M. Good, "MusicXML for Notation and Analysis," *The Virtual Score Representation, Retrieval, Restoration*, W.B. Hewlett and E. Selfridge-Field, eds., MIT Press, 2001, pp. 113-124.
5. P. Bellini and P. Nesi, "Wedelmusic Format: An XML Music Notation Format for Emerging Applications," *Proc. 1st Int'l Conf. Web Delivery of Music*, IEEE Press, 2001, pp. 79-86.
6. P. Bellini, R. Della Santa, and P. Nesi, "Automatic Formatting of Music Sheet," *Proc. 1st Int'l Conf. Web Delivering of Music* (Wedelmusic), IEEE Press, 2001, pp. 170-177.
7. *Standard Music Description Language*, ISO/IEC DIS 10743, Int'l Organization for Standardization/Int'l Electrotechnical Commission, 1995.
8. *NIFF 6a.3: Notation Interchange File Format*, NIFF Consortium, 1998; http://neume.sourceforge.net/niff/.

becoming of paramount importance. To support the inclusion of these music representations, the model must be sufficiently open and general. Furthermore, educational aspects strongly influence requirements for effective music representation models, because they must integrate semantics information for rendering or playing and performance evaluation at the music element level. Many researchers have addressed the notation-modeling problem with its related rendering through computer systems.[4-9]

Symbolic music representation

Current tools for applications integrating multimedia with music notation are based on proprietary formats. The introduction of an MPEG SMR will let users move some relevant complexity from software tool development to the content model, so that standard tools can reenter the same content, including SMR, on more tools and devices. This will increase the market potential for new music applications. It was on these grounds that the MusicNetwork proposed a new work item for integrating SMR into MPEG standard formats (primarily MPEG-4[10]).

MPEG SMR will allow the synchronization of symbolic music elements with audiovisual events that existing standardized MPEG technology can represent and render. The breadth of MPEG standards for multimedia representation, coding, and playback, when integrated with SMR through an efficient collaborative development, will provide increased content interoperability in all MPEG SMR-compatible products. It will also help with exchanging and rendering SMR codes in a secure, efficient, and scalable manner.

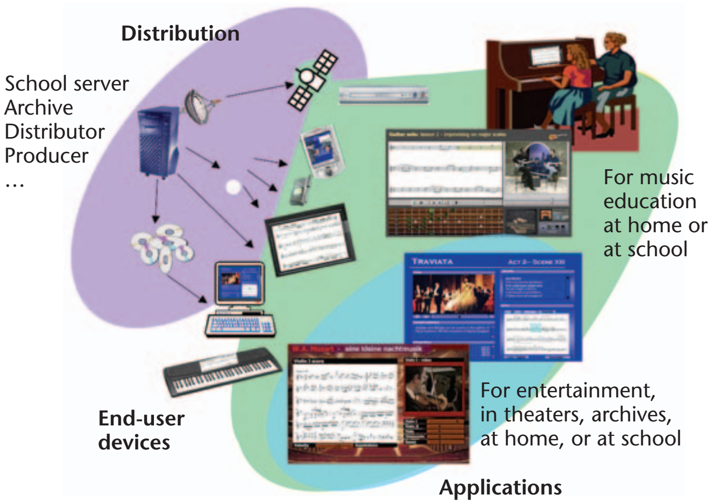


Figure 1. General scenarios of MPEG symbolic music representation (SMR) in entertainmentand education. MPEG-4 with SMR supports distribution through satellite data broadcast, the Internet, and wireless or traditional communication to theaters, homes, archives, and devices such as interactive TVs and PDAs.

Figure 1 summarizes the new applications that MPEG SMR makes possible. The many music-related software and hardware products currently available can greatly benefit from MPEG SMR, because it will foster new tool development by allowing increased functionality at a reduced cost. This increased functionality will quickly affect such applications as interactive music tutorials, composition and theory training, and multimedia music publication.

Information about the ongoing MPEG SMR activity—including a large collection of documents containing MPEG SMR requirements, scenarios, examples, and links—is available from the MPEG ad hoc group on SMR Web pages at http://www.interactivemusicnetwork.org/mpeg-ahg.

Application scenarios

Currently available products presenting some form of integration of SMR and multimedia include tools in the following areas:

- music education (for example, notation tools integrating multimedia, multimedia electronic lecterns for classrooms and lecture halls, and music education via interactive TV),
- music management in libraries (such as music tools integrating multimedia for navigation and synchronization),
- interactive entertainment (for example, karaoke-like synchronization between sound, text, and symbolic information),
- piano keyboards with symbolic music representation and audiovisual capabilities, and
- multimedia content integrated with music notation on musical instruments, PDAs, and so on.

Some innovative applications integrating synchronization of music notation and other media and 3D virtual reality are already in use in commercial tools such as Voyetra, SmartScore, PlayPro, and PianoTutor, and in prototypes from research and design projects (such as OpenDrama, Wedelmusic,[1] Moods,[3] and Imutus). All of these applications exploit MPEG technology for standard multimedia integration and the possibility of distributing various forms of content in a completely integrated manner. Organizations can use the distribution of multimedia music content to reach the mass market on interactive TV, mobile devices, media centers, and PCs for education, entertainment, and other applications.

Entertainment

One of the most interesting applications is the publication of multimedia music for entertainment, interactive music tutorials, multimedia music management in libraries, archives, and theaters. These applications integrate SMR with audio, lyrics, annotations, and visual and audible renderings (such as notations, parts and main scores, ancient and common western notation), and synchronize with audiovisual content.

For example, a multimedia music application for lyric operas, such as OpenDrama, synchronizes the music score with a video of the opera or with a virtual reality version (see Figure 2), letting the user

- follow the music by reading the instruments' or singers' score and selecting the lyric in a different language if desired;
- read the libretto and its translation;
- read and understand the plot (which can be intricate); and
- personally annotate the music to mark a particular passage.

With such an application, a user inside a theater could follow the actions on stage using a palmtop installed in the stalls.

Education

Another application is the exploitation of

MPEG SMR to support education (Musicalis, Imutus, and Wedelmusic are examples). As Figure 3 illustrates, these applications integrate SMR with audio, lyrics, annotations, and semantics for visual and audible renderings, and synchronize with audiovisual content.. They also provide 3D virtual renderings of hand and body positions (posture and gesture) or of the scene.

Music education and edutainment are music representation's largest markets. Just as we can synchronize a text to images and sounds in commercial DVDs, we can also synchronize audio, video, and music representation for various instruments and voices, and with various models or rendering aspects (tablature, Neumes, Braille, and so on). The capability to distribute content in different forms but related to the same piece of music is therefore desirable.

For music education and courseware construction, music notation must be integrated with multimedia capabilities. In a large part of Europe about 80 percent of young students study music in schools; thus, many of them can read music representation and play an instrument to some extent. Lower but still relevant percentages can be observed in other countries.

Music courseware must integrate music representation with other elements, such as video and audio files, storic, biographic, and theory documents, and animation. A music course can also offer exercises requiring special music notation symbols (assigned by an instructor or user-defined, for example) or audio processing (such as play or theory training assessment). Thus, music education and courseware production requires that both the client (music content usage) and server (content generation) have a visual representation of the musical model with full access to logical aspects and additional semantics information for visual and audio rendering, as well as the capability to manipulate music, support synchronization, and establish relationships with other media. These systems and models must therefore let users do the following:

- navigate through music representation features and multimedia content by following links associated with music symbols or entities (such as a video explaining a difficult passage's semantics or the composer's goal);
- edit music, which implies dynamically altering the symbolic music's logical, audio, and visual domains;
- transpose music notation—that is, dynamically altering details in the symbolic music's logical, audio, and visual domains;
- play music by interpreting the symbolic music representation to produce, for example, a visual rendering with some scrolling and a synchronized audio;
- format music notation—that is, dynamically interpret the symbolic music representation to produce a visual rendering that fits the screen or window size;
- reduce a music score with one or several parts to a single piano part, which involves dynamically altering aspects in symbolic music's logical, audio, and visual domains;
- select one lyric from a collection of multilingual

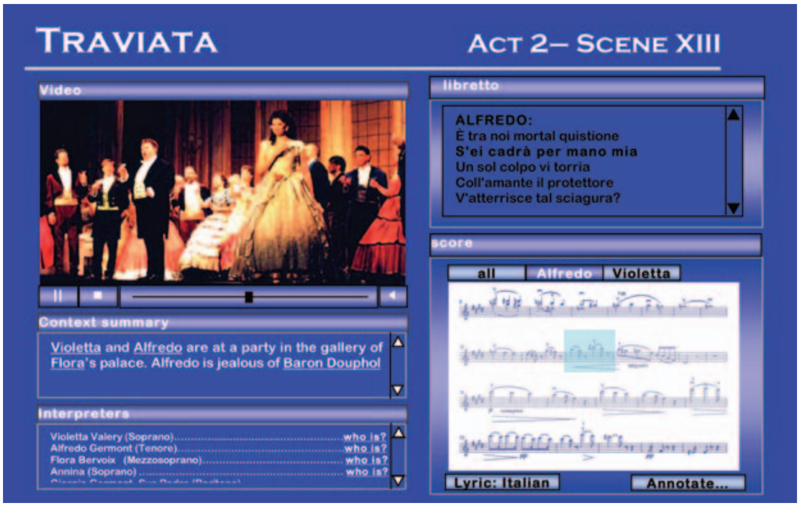


Figure 2. Mock up of a tool exploiting MPEG SMR for entertainment for theaters, interactive TV, archives, PCs/Tablet PC.

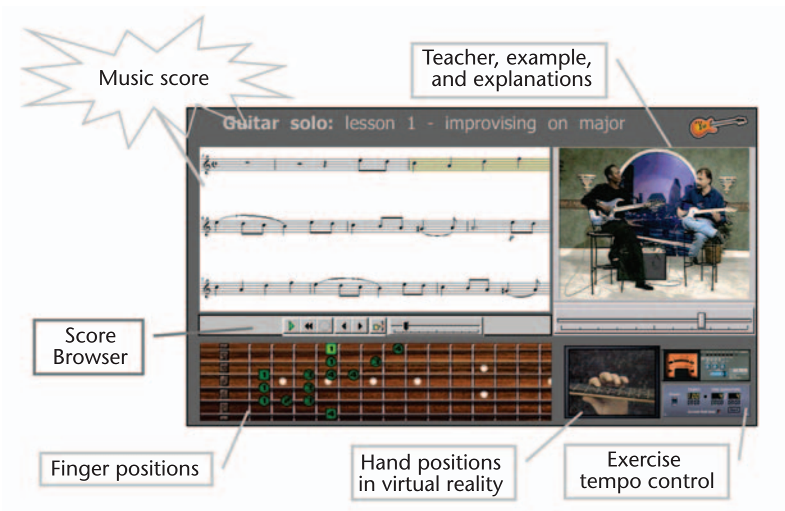


Figure 3. Mock up of a tool for exploiting MPEG SMR for education that's suitable for assisted or self-learning on interactive TV, Tablet PC, and mobile devices.

lyrics for the same music representation, or use lyrics in which music notation is synchronized with video and/or other audio content;

- synchronize audiovisual events with music representation elements to show visually how a piece of music should be played or a score followed (this can also involve changing performance velocity, adjusting playing parameters, and so on);

- show video of the instructor playing an instrument synchronously with the music notation or show the 3D rendering of correct hand gestures;

- play along with the computer, which could play selected parts using real audio, musical instrument digital interface (MIDI, http://www.midi.org) files, or a more sophisticated music notation rendering, or could comment on and correct the user's work; and

- format music for rendering on different devices, in different formats, or with different resolutions (for example, spoken, or talked, music is a verbal description of a music context useful for hearing impaired, dyslexic, or young students, or for nonexperts editing music notation; see http://projects.fnb.nl/Talking%20Music/default.htm).

More interactive and complex players integrate additional capabilities to support the following:

- textual, notational, or audiovisual annotation, including capabilities for versioning and attaching complex information and rendering hints to SMR elements;

- recognition of notes sung or played by an instrument using pitch recognition or other technologies—that is, the audio events must be in line with the notation conventions at the logical or symbolic level, creating an appropriate reconstruction and maybe rendering;

- assessment with respect to semantic annotations—for example, how to execute a certain symbol or assess a student's execution of a music notation piece (as noted earlier, a good level of assessment requires the precise modeling of pitch, rhythm, and sound quality, as well as detailed descriptions of physical gestures);

- cooperative music notation for classes, for orchestral work on music lecterns (for example, using tablet PCs), for rehearsal management, and so on; and

- customization of SMR rendering according to the individual's needs (for example, selecting symbols to be verbally described, printed, or played, or selecting a simpler model for the visualization or playing).

Integration

Current standardized MPEG tools don't systematically or satisfactorily support symbolic forms of music representation. Indeed, MPEG-4 covers a huge media domain. In particular, the standard's audio part lets an author synchronize standard MIDI content with other forms of coding.

Furthermore, structured audio (SA) allows structured descriptions of audio content through a normative algorithmic description language associated with a score language that's more flexible than the MIDI protocol. Although it's possible to extract symbolic representation from the information carried by SA tools (including MIDI), this information can't sufficiently guarantee correct notation coding because they lack, for example, information about visual and graphic aspects, many symbolic details, a thorough music notation modeling, and many necessary details for a correct human-machine interaction through the SMR decoder (for example, visual and concise respresentation of sequences of events are represented by ornaments and not by the detailed notes).

MPEG-7 also provides some symbolic music-related descriptors, but they aren't intended for coding SMR as a form of content. On the other hand, SMR content can be a complete representation in itself and can be synchronized with other audiovisual elements.

The integration of SMR in MPEG satisfies the application requirements described earlier, as well as many more, letting such tools integrate with the powerful MPEG model for multimedia representation, coding, and playback. At the same time, compliance with the MPEG SMR standard won't require compliance with all MPEG-4 parts and tools. MPEG mechanisms can cope with the standard's complexity through profiles, which could let users shape specific SMR-oriented applications using only a few selected MPEG-4 features, as is the case for many other parts of MPEG-4. In fact, SMR's added value for multimedia music integration goes beyond the development inside MPEG-

4, because its core part can also act as a standalone tool outside the MPEG framework.

SMR will be integrated into MPEG-4 by

- defining an XML format for text-based symbolic music representation that will allow interoperability with other symbolic music representation and notation formats and act as a source for the production of equivalent binary information stored in files or streamed by a suitable transport layer;
- adding a new object type for delivering a binary stream containing SMR and synchronization information and providing an associated decoder that will let users manage the music notation model and add the necessary musical intelligence to allow human interaction;
- specifying the interface and behavior for the SMR decoder and its relationship with the MPEG-4 synchronization and interaction layer (systems part, binary format for scenes, or BIFS, nodes, and advanced graphics parts).

Figure 4 illustrates SMR's use during the authoring phase, highlighting the MPEG-SMR Extensible Markup Language (XML) format's central role. Users can produce SMR XML content using appropriate converters or a native SMR music editor. An MPEG-4 SMR-enabled encoder can then multiplex the SMR XML file into an MPEG-4 binary file (standard XML binarization is also available in MPEG). A server can store or distribute MPEG-4 content as a binary stream to clients. The SMR binary stream contains information about music symbols and their synchronization in time and space. A decoder in the user's terminal (player) converts this stream into a visual representation, which can be rendered, for example, in the BIFS scene.

Figure 5 reports a simple example of an MPEG-4 player supporting MPEG-4 SMR. The player uses the SMR node in the BIFS scene to render the symbolic music information in the scene (for example, by exploiting functionality of other BIFS nodes) as the SMR decoder decodes it. The user can interact with the SMR (to change a page, view, transpose, and so on) through the SMR interface node, using sensors in association with other nodes defining the audiovisual, interactive content. The user sends commands from the SMR node fields to the SMR decoder (dashed lines in Figure 5), which generates a new view for displaying the scene.

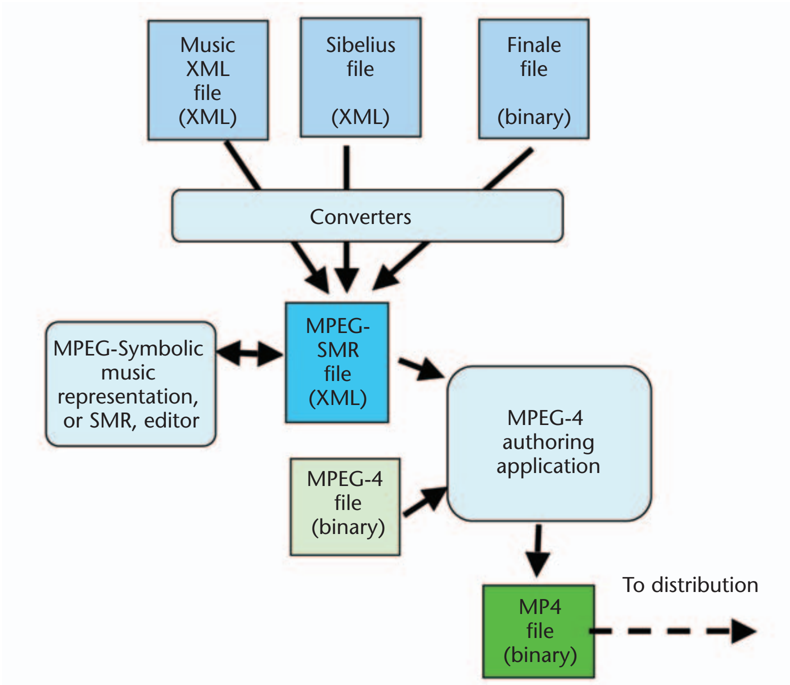
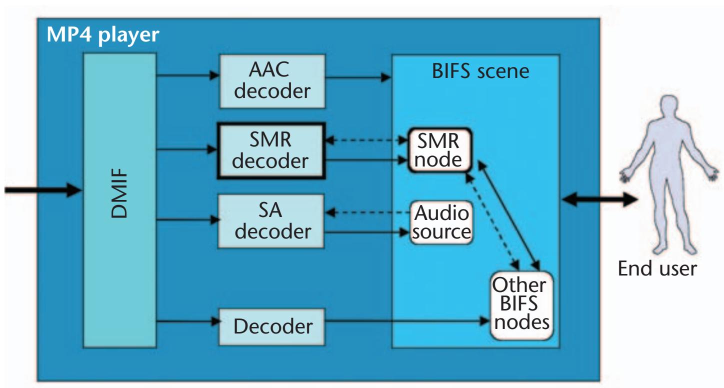


Figure 4. SMR in the authoring phase.

The SMR decoder consists of several components, as Figure 6 (next page) illustrates:

- The binary decoder decodes the binary stream coming from the MPEG-4 delivery multimedia integration framework (DMIF) interface or from an MPEG-4 file. The decoder extracts the optional SMR rendering rules and synchronization information from the SMR access units (AUs), loading the SMR rendering rules data structure to any SMR rendering rules engine and sending the synchronization information to the SMR manager.



AAC	Advanced audio coding
BIFS	Binary format for scenes
DMIF	Delivery multimedia integration framework
SA	Structured audio
SMR	Symbolic music representation

Figure 5. SMR inside an MPEG-4 player.

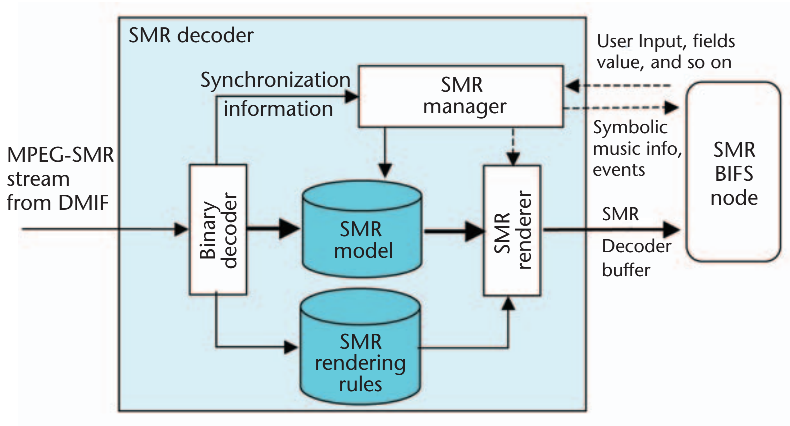


Figure 6. Structure of an MPEG-SMR decoder.

- The SMR model contains SMR elements and information only and refers other object types to other MPEG objects.

- The SMR renderer, controlled by the SMR manager, uses the SMR model with its parameter values and the SMR rendering rules to produce a view of the symbolic music information in the SMR decoder buffer. The rendering part of the SMR in the visual domain is based primarily on the Wedelmusic solution.[11] Wedelmusic uses modules of music justification and symbols positioning working together to arrange the SMR symbol placement on the page or screen according to parameters such as page size, justification values, formatting rules, and style.

- The SMR decoder buffer can contain pixels and vector graphics information, depending on the specific SMR coded content. The graphic functionality of BIFS (or its textual format, Extensible MPEG-4 Textual format [XMT-A]) partially overlaps that of scalable vector graphics (SVG), which is a current tool used for implementing the graphic layer of symbolic music.

- The SMR manager coordinates the SMR decoder's behavior. It receives and interprets events from the SMR node interface. Depending on the command type, it modifies parameters in the SMR model (such as transposition) and controls the SMR renderer. It also controls the synchronized rendering.

- The SMR node and other BIFS nodes produce composition buffers. The SMR node specifies the interface events to the rest of the BIFS scene and the user.

In MPEG-4, integration and synchronization rely on AUs, which are individually accessible portions of data in elementary streams. An AU is the smallest data entity to which the system (or media synchronization) layer can attribute timing information.

We can encode SMR into the bit stream AUs in three ways:

- We can include part (or all) of the SMR synchronization information (that is, time labels associated with events) in the header (ObjectTypeSpecificInfo).

- We can insert SMR synchronization information in AUs with embedded time stamps and thus can deliver AUs with content intended for a later use.

- We can insert SMR synchronization information in AUs without embedded time stamps. In this case, the AU's decoding time stamp is the reference time; events are to be scheduled upon receipt.

We can associate each possibility with different use cases. In some situations, content is more or less available before its playback, and everything is codable. In other situations, we must use AUs (for example, if the performer is playing a MIDI device in real time and transmitting it in a stream).

SMR dedicates particular care to the relationship with MIDI information. If a publisher wants to use only some MIDI files in MPEG-4-compliant devices (using the simplest object type defined in the SA subpart), and if these devices support SMR visualization, the specification will let a user client tool automatically convert MIDI files (through a specific algorithm) into SMR on the client side and render them. Similarly, it might only deliver the SMR. In these cases, the client can generate the MIDI information from SMR for use with MIDI-compliant devices. This is particularly important to guarantee straightforward adaptation of current devices.

Conclusion

Integration of SMR in multimedia frameworks, and particularly MPEG, is opening the way to the implementation of a large set of new applications in education, entertainment, and cultural valorization. The overall architecture and functionality presented in this article are consolidated, and the standardization process has already produced the reference model and software. Some side issues will be solved as the standardization progresses. These issues include the relationship with parts of SA other than MIDI, the exact scope of formatting rules for music symbols, and the precise specification of standard interaction with a compliant decoder.

Looking ahead, MPEG SMR might have value as an independent standard for music representation and as a data type to be integrated into applications based on MPEG-21, which address other fundamental issues for multimedia frameworks, such as digital rights management and media adaptation. **MM**

Acknowledgments

We express our deep thanks to everyone who took part in the MPEG ad hoc group on SMR, all those belonging to the MusicNetwork working group on music notation, and those who attended joint meetings of MPEG and MusicNetwork. In particular, we thank Jerome Barthelemy, David Crombie, Matthew Dovey, Eleanor Selfridge Filed, James Ingram, Neil McKenzie, Steve Newcomb, Kia Ng, Hartmut Ring, Perry Roland, Martin Ross, Jacques Steyn, Tillmann Weyde, and Tom White. We apologize in advance if we forgot to mention other names.

References

1. P. Bellini et al., “Multimedia Music Sharing among Mediateques, Archives, and Distribution to Their Attendees,” *J. Applied Artificial Intelligence*, Taylor and Francis, Sept.- Oct. 2003, pp. 773-795.
2. P. Bellini, F. Fioravanti, and P. Nesi, “Managing Music in Orchestras,” *Computer*, Sept. 1999, pp. 26-34.
3. P. Bellini, P. Nesi, and M.B. Spinu, “Cooperative Visual Manipulation of Music Notation,” *ACM Trans. Computer-Human Interaction*, vol. 9, no. 3, Sept., 2002, pp.194-237.
4. D. Blostein and H.S. Baird, “A Critical Survey of Music Image Analysis,” *Structured Document Image Analysis*, H.S. Baird, H. Bunke, and K. Yamamoto, eds., Springer Verlag, 1992, pp. 405-434.
5. D.A. Byrd, *Music Notation*, PhD dissertation, Dept. of Computer Science, Indiana Univ., 1984.
6. R.B. Dannenberg, “A Structure for Efficient Update, Incremental Redisplay, and Undo in Graphical Editors,” *Software Practice and Experience*, vol. 20, no. 2, Feb. 1990, pp. 109-132.
7. R.B. Dannenberg, “A Brief Survey of Music Representation Issues, Techniques, and Systems,” *Computer Music J.*, vol. 17, no. 3, 1993, pp. 20-30.
8. G.M. Rader, “Creating Printed Music Automatically,” *Computer*, June 1996, pp.61-68.
9. E. Selfridge-Field, ed., *Beyond MIDI: The Handbook of Musical Codes*, MIT Press, 1997.
10. F. Pereira and T. Ebrahimi, eds., *The MEPG-4 Book*, USC Integrated Media Systems Center Press, 2002.
11. P. Bellini, I. Bruno, and P. Nesi, “Automatic Formatting of Music Sheets through MILLA Rule-Based Language and Engine,” *J. New Music Research*, vol.34, no. 3, 2005, pp. 1-21.

Pierfrancesco Bellini is a contract professor in the Department of Systems and Informatics at the University of Florence. His research interests include object-oriented technology, real-time systems, formal languages, and computer music. Bellini has a PhD in electronic and informatics engineering from the University of Florence, and has worked on Moods, Wedelmusic, Imutus, MusicNetwork, Axmedis, and many other projects funded by the European Commission.

Paolo Nesi is a full professor in the Department of Systems and Informatics at the University of Florence and chief of the Distributed Systems and Internet Technology research group. His research interests include object-oriented technology, real-time systems, computer music, and parallel and distributed architectures. Nesi has a PhD in electronic and informatics engineering from the University of Padoa. He has coordinated several multipartner international research and design projects of the European Commission such as Moods, Wedelmusic, Axmedis, and the Interactive Music Network.

Giorgio Zoia is a scientific advisor at the Signal Processing Institute of the Ecole Polytechnique Fédérale de Lausanne (EPFL). His research interests include audio include 3D spatialization, audio synthesis and coding, representations and description of sound, interaction and intelligent user interfaces for media control. Other research interests include compilers, virtual architectures, and fast execution engines for digital audio. He has a PhD in technical sciences from EPFL.

Readers may contact Paolo Nesi at nesi@dsi.unifi.it.