IEEE 1599: Music Encoding and Interaction

Denis Baggi, *University for Applied Science of Southern Switzerland*

Goffredo Haus, *University of Milan, Italy*

IEEE Std 1599 allows interaction with music content such as notes and sounds in video applications and in any interactive device.

A new standard, *IEEE 1599, Definition of a Commonly Acceptable Musical Application Using the XML Language*, was proposed to the IEEE Standards Association in 2001. The proposal was supported by the IEEE Computer Society's Standards Activities Board and the Intelligent Manufacturing Systems international global research fund, and financed by the Commission for Technological Innovation of the Swiss Federal Government. Submitted to balloting in 2007 and 2008, IEEE 1599 received a 96 percent approval rating, with the final draft approved unanimously by NESCOM, the NEw Standards COMmittee, in 2008. It is featured in Standards Wire at http://standards.ieee.org/standardswire/archives/sw_nov08_email.html, and its DTD code is available at http://standards.ieee.org/downloads/1599/1599-2008/. An initial, though complete, description appeared in *Computer* in 2005 (D. Baggi, "An IEEE Standard for Symbolic Music," Nov. 2005, pp. 100-102).

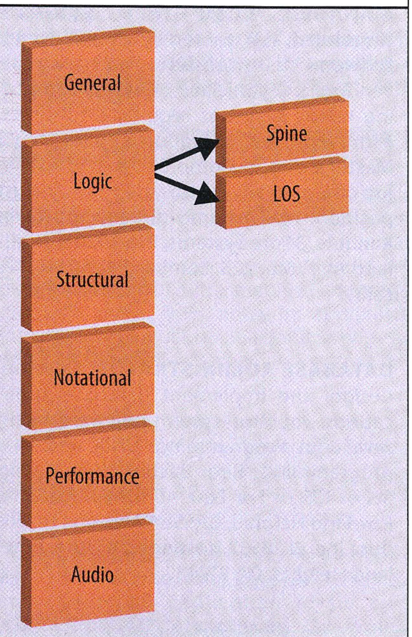

Figure 1. A musical piece can have several layers.

The main ideas behind the standard are that music can and should be encoded with symbols independently of the format in which, for example, its audio is delivered, and that, to make it interactive and navigable, this representation must take into account all aspects of music. Simply stated, music is not *just* sounds such as those generated by an MP3 file or some other binary and impenetrable format. Rather, it has several aspects such as sound, images, score, texts, videos, reference data, alternative versions, random access, and possibly separate audio and video tracks for every instrument and voice within the piece.

Several demo applications have been built that demonstrate these features, transforming the music experience from passive listening to navigating through a virtual world, similar to what happens when surfing the Web or reading a novel.

IEEE 1599'S BASIC FEATURES

Essentially, two features distinguish IEEE 1599 from any other standard or encoding: the definition of *layers* (G. Haus and M. Longari, "A Multi-Layered Time-Based Music Description Approach Based on XML," *Computer Music J.*, Spring 2005, pp. 70-85) and the use of *symbols*.

There can be several layers in a musical piece, as Figure 1 shows.

The General Layer provides a general description of the piece and groups information for all related instances, including titles, author, type, number, date, genre, and related items. The Logic Layer, which represents the format's core, describes music from a symbolic viewpoint. This layer contains the main time-space construct for localization and synchronization of music events, the description of the score with sym-

Published by the IEEE Computer Society

0018-9162/09/$25.00 © 2009 IEEE

bols, and information about graphical implementations of the symbolic contents, as well as the Spine, with logically organized symbols (LOS), a sorted list of music events. This list makes it possible to interact by, for example, using a layer's information to find the corresponding information in other layers. All layers are related to each other, as Figure 2 shows.

This demonstrates that the IEEE 1599 standard allows representation, performance, and audible and visual fruition of a musical piece independently of the original standard or format with which it was previously encoded. Thus, it supports existing formats and—thanks to links to the Notational, Performance, and Audio layers—handles media files and contents in their original encoding. Not all layers might be present in a piece of music, and others could be added that do not appear in Figure 1.

All these layers are synchronized through the encoding of *events* expressed as XML symbols, as Figure 3 shows. A musical event, such as a note, has a unique tag that is found again in other layers, so that interacting with it—on a music score, for example—automatically resets the performance to that instant and to the corresponding sound, an image points to the appropriate graphics, a text pointer to the corresponding text, and so on, depending on the layer. Thus, interacting in one layer links all other layers throughout the performance, restarting it at the proper point.

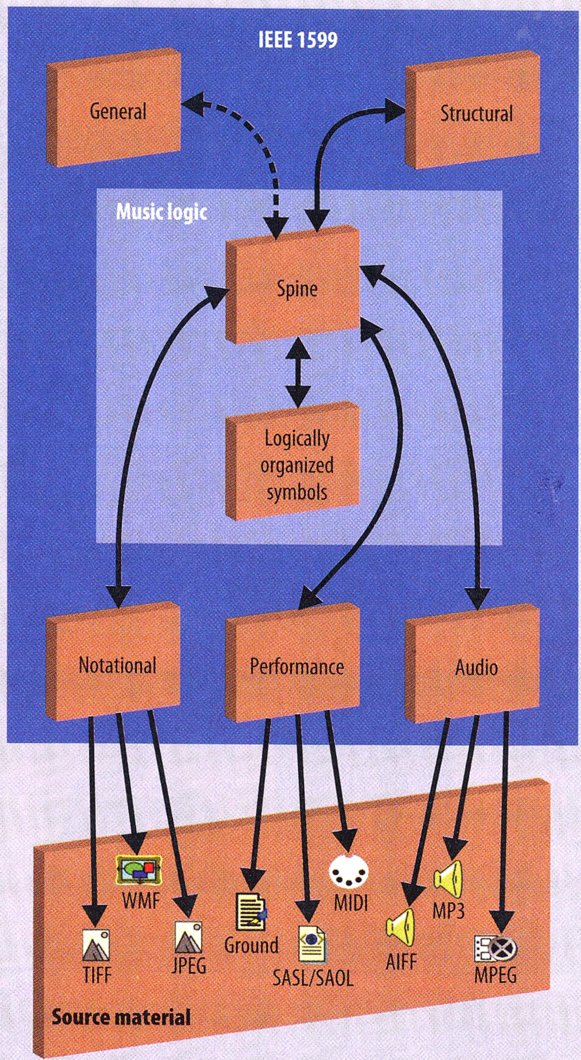

Figure 2. Synchronization among the layers of Figure 1.

Treble	<clef type="G" staff_step="2" event_ref="c1"/>

Figure 3. Example of an XML symbol to represent events for synchronization.

DEMO EXAMPLE

The demonstration in Figure 4 is built from an improvised piece of jazz, on the structure of the tune *Crazy Rhythm*. This piece was recorded in Paris in 1937 by American saxophonists Benny Carter on alto and Coleman Hawkins on tenor, and French specialists André Ekyan on alto and Alix Combelle on tenor, with the support of Django Reinhardt on guitar.

Figure 4 shows a screenshot of this performance. Since there is no score, the harmonic grid is used to control the piece interactively, which consists of 32 squares that represent a measure or bar, with symbols representing the underlying harmony, a music syntax well known to jazz musicians.

Normally, the user starts the piece and lets it run to its end, while the display changes, showing the full band for the tune exposure, then each soloist in turn for 32 bars, and a pop-up at bar 31 for Hawkins, with Reinhardt shouting "go on, go on" to continue the improvisation beyond the length planned initially. Hence, a user learns how to identify soloists

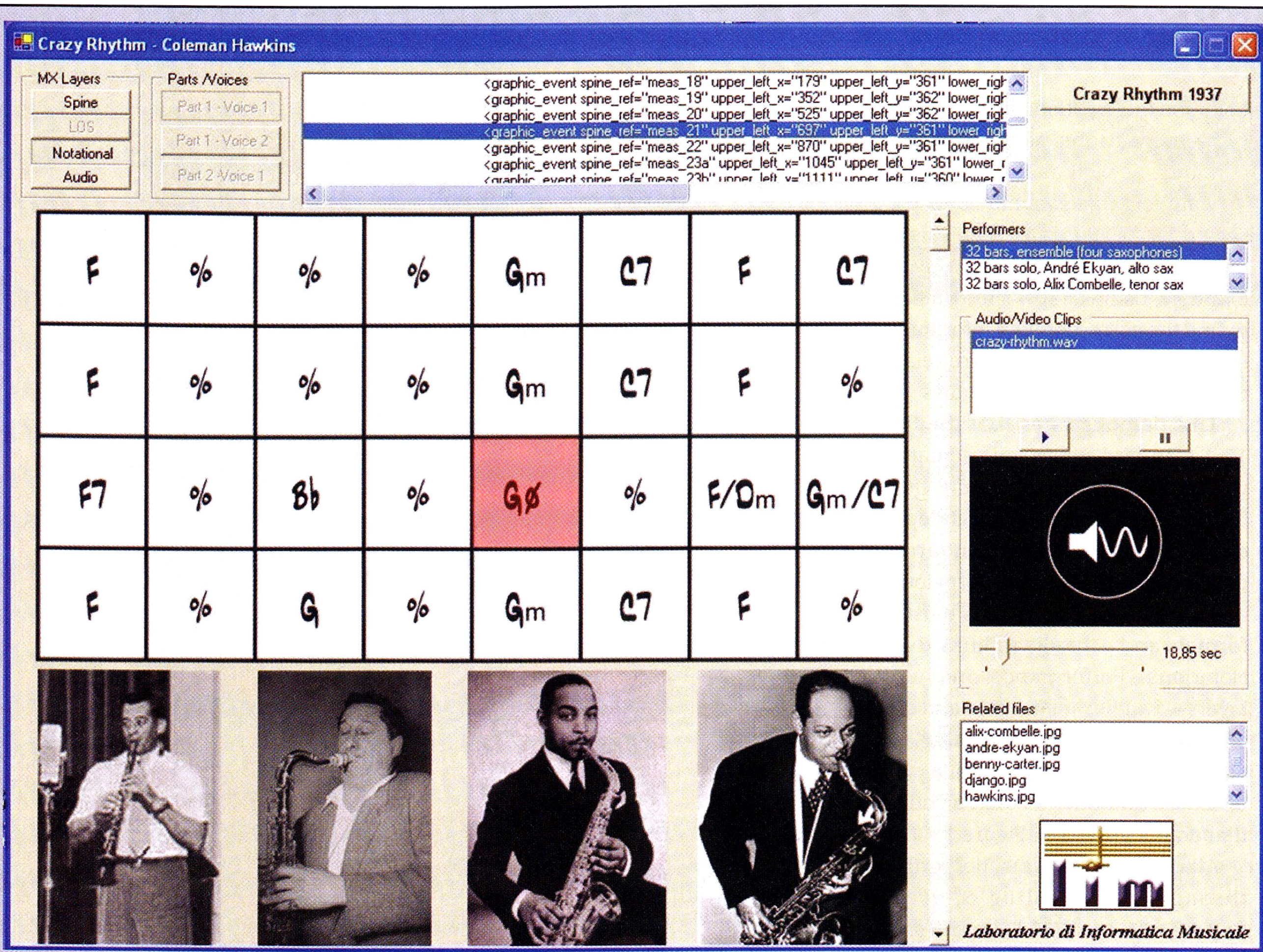

Figure 4. Screenshot for the jazz piece *Crazy Rhythm*.

by image and name in the Performers window, see the XML code in the MX Layers window, and restart at any moment by clicking either on the soloist image or on the grid, or by moving the slide on the media player. The user is also taught to catch details—such as Reinhardt's shout—typical of an improvised and partly unplanned performance.

We have planned a game that, when complete, in addition to allow clicking on the grid during a solo will let the user interact and select features of the music, such as enhancing only certain notes, replacing some notes with others, transforming the sound into another—for example, from sax to trumpet or xylophone—hashing notes and moving them in the time domain to affect the swing, and so on. In this way, alternate pieces of music derived from the original could be realized and compared with what the musicians improvised more than 60 years ago.

A MUSIC GAME: TO PLAY THE BLUES

The 12-bar blues is one of the simplest structures on which to improvise in jazz and underlies the vast majority of jazz tunes. The grid in Figure 5 shows its basic structure, which—for readability—has been reduced to its simplest expression, and does not correspond to what advanced musicians use by substituting chords of the grid.

The game, still under development, currently contains three versions of the blues evolved from the basic grid, corresponding to different stylistic epochs that can be selected: 1920s through 1930s, 1940s through 1950s, and 1960s through 1970s. The user further interacts either with a MIDI device connected to the computer, such as a keyboard, wind instrument, guitar, or a piano keyboard simulated on the computer keyboard. As the rhythm section (piano or guitar, bass, and drums) plays, the user enters his or her own solo. Instructions for beginners suggest using notes included in the grid chord, and little by little the user learns how to experiment with other notes.

At the session's end, the system grades and assesses the performance, possibly according to criteria that can be set up, preconfigured, and dimensioned by the user, such as basic correctness, stylistic consistency, originality, and similar metrics that have previously been defined according to jazz musicology (D. Baggi, *Readings in Computer Generated Music*, IEEE CS Press, 1992).

BEYOND MUSIC ENCODING

Interaction is a major component of a successful videogame. However, with few exceptions, music is relegated to the role of background noise, as in *Tetris*, flippers, and car chases. The few cases for which this does not hold, such as *Guitar Hero* (www.cnn.com/2008/TECH/ptech/10/21/guitar.hero/index.html), are enjoying such success that pop artists have begun to create music products for that media rather than for CDs (www.ew.com/ew/article/0,,20222085,00.html). The Nintendo Wii has dedicated much effort to interaction for controlling the Wii racket based on Wiimote, which has resulted in the capture of a consistent market. However, even in the cases where music is interactive, much more can be done by accessing the musical features of a sequence of sounds. But doing so requires an appropriate technology, such as that of synchronized layers.

Videogames and consoles such as the Nintendo Wii and Sony Playstation—as well as recent cell phones such as the iPhone—typically dedicate an audio component for playing music, but fail to use it to its fullest. Thus, the new IEEE 1599 standard has been developed to allow interaction with music content, such as notes and sounds in video applications and in any interactive device.

IEEE 1599 supports a technology that makes music navigable and interactive. Its layers represent all the multimedia aspects of music, which can be expressed in existing specialized formats such as WAV, MP3, MPEG, and JPEG. XML files of symbols synchronize the layers, which represent an event referring and pointing to different instances of the same event in the layers. As a result, interaction is built-in and automatically propagates throughout all layers.

The examples we have described are only a first step toward achieving the integration of the IEEE 1599 technology for describing musical information—thanks to symbols, layers and spine—with entertainment platforms for multimodal human-computer interaction, in which musical modes—such as vocal and instrumental audio, music writing and displaying, and rhythmic and melodic sequences—offer the user new and intriguing possibilities to further expand market successes such as *Guitar Hero*.

There is little doubt that future work on IEEE standards for game technology will require expertise in music technology. The IEEE Computer Society, the Technical Committee on Computer Generated Music, and Standard IEEE 1599 (sponsored by the IEEE Computer Society Standards Activities Board) can assist in the realization of interactive and navigable musical applications for videogames. ■

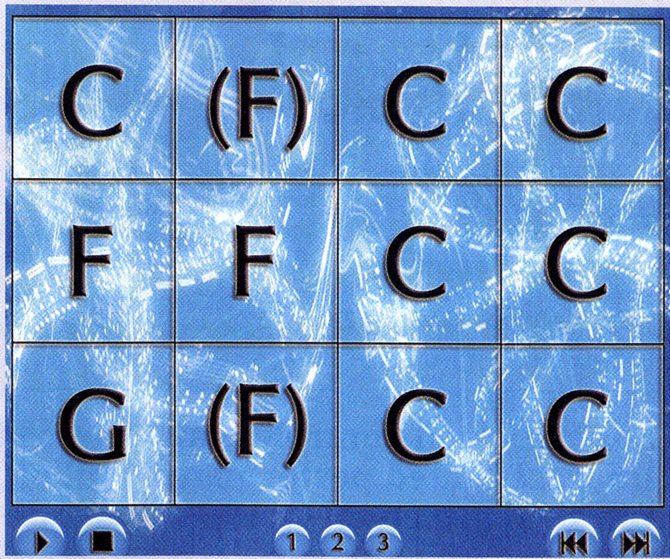

Figure 5. A music game: Learn how to play the blues.

Denis Baggi *is Working Group Chair of IEEE 1599 and Founder of the IEEE CS Technical Committee on Computer Generated Music. He is a faculty member in the Department of Technological Innovation at the University for Applied Science of Southern Switzerland. Contact him at denis.baggi@supsi.ch.*

Goffredo Haus *is the Official Reporter for IEEE 1599, Founder and Director of the Laboratory for Musical Informatics (LIM), and head of the Department of Communication and Information Science, University of Milan, Italy. Contact him at haus@dico.unimi.it.*

Editor: John Harauz, Jonic Systems Engineering, Inc., Willowdale, Ont., Canada; j.harauz@ieee.org