Advanced Interfaces for Music Enjoyment

Adriano Baratè
Laboratorio di Informatica Musicale
Università degli Studi di Milano
Via Comelico, 39 – 20151 Milano
+39 02 50316382
barate@dico.unimi.it

Luca A. Ludovico
Laboratorio di Informatica Musicale
Università degli Studi di Milano
Via Comelico, 39 – 20151 Milano
+39 02 50316382
ludovico@dico.unimi.it

ABSTRACT
Music enjoyment in a digital format is more than listening to a binary file. An overall music description is made of many interdependent aspects, that should be taken into account in an integrated and synchronized way. In this article, a proposal for an advanced interface to enjoy music in all its aspects will be described. The encoding language that allows the design and implementation of such interface is the IEEE P1599 standard, an XML-based format known as MX.

Categories and Subject Descriptors
H.5.1 [**User Interfaces**]: Graphical user interfaces (GUI)
H.5.5 [**Sound and Music Computing**]: Modeling

General Terms
Algorithms, Standardization, Languages.

Keywords
Music, XML, MX, multimedia, synchronization.

1. INTRODUCTION
Describing music in all its aspects can be a challenging matter. For example, pop songs are usually distributed in form of tracks (audio content), nevertheless those music pieces are based on a score (symbolic content), have their own lyrics (text content), can be associated to a video clip (video content), etc.; besides, for each of those multimedia categories, many descriptions are allowed: for instance, the radio, the "unplugged" and the live version of audio tracks, the official video clip and video recordings of a live concert, etc. Of course, even more complex examples could be cited: for an opera, also sketches, fashion plates, on-stage photos, playbills constitute descriptions of music from a particular point of view.

In other words, in order to describe a music piece, many complementary multimedia objects can be used. Heterogeneity is involved from two different standpoints: i) the number of different multimedia descriptions (metadata, music symbols, text, still graphics, audio, and video), and ii) the number of different objects belonging to each category. As a matter of fact, institutions oriented to music performance - such as *Teatro alla Scala* - or to audio archives - such as *Discoteca di Stato* (the Italian national discotheque) - preserve not only the material belonging to their mission (scores and audio supports, respectively), but also a number of related objects, in order to provide the most comprehensive description of music pieces.

If heterogeneous music-related documents are available, the definition of a multimedia and multimodal environment is particularly interesting for a user who wants to investigate a music work from a number of perspectives, ranging from score analysis to performance and interpretation comparisons.

The advanced interface proposed in this work has two goals: first, it is aimed at demonstrating the integrated and synchronized description of music within a single XML-based file; besides, it provides an example of future ways to enjoy music at different degrees of comprehension and abstraction.

Avi'08, May 28-30, 2008, Napoli, Italy

2. MX (IEEE P1599)
The design and the implementation of the interface is based on an XML format named MX, an acronym which stands for *Musical application using XML*. MX international standardization is in progress, and its development follows the guidelines of IEEE P1599, *Recommended Practice Dealing With Applications and Representations of Symbolic Music Information Using the XML Language* [1]. This project proposes to represent music symbolically in a comprehensive way, opening up new ways to make both music and music-related information available to musicologists and performers on one hand, and to non-practitioners on the other. Accordingly, its ultimate goal is to provide a highly integrated representation of music, where score, audio, video, and graphical contents can be appreciated together. For further details please refer to [2] or to the on-line documentation at http://www.mx.dico.unimi.it.

3. REPRESENTATION OF MUSIC IN MX
MX provides a framework for a comprehensive description of music. It is based on two key issues: i) richness in the kinds of description related to the same music piece, ranging from symbolical and logical to analytical and media descriptions, and ii) the possibility to link a number of instances for each media type, i.e. a number of media objects of the same type.

These requirements can be satisfied thanks to the MX's multi-layer structure, made of 6 levels: *General*, *Logic*, *Structural*, *Notational*, *Performance*, and *Audio*. This structure has been proposed and studied in [3]. Each layer aims at describing a music piece from a different and complementary perspective. The

General layer contains catalog metadata about the piece and its XML encoding. The *Logic* layer describes the music piece in terms of symbols, such as notes, rests and other music signs. The *Structural* layer provides an environment to define music objects and to describe their relationships, such as in harmony or formal analysis. The *Notational* layer describes the digital objects containing graphical representations of scores, typically graphic files. The *Performance* layer allows to link and synchronize computer-driven performance formats, such as MIDI or SASL/SAOL. Finally, the *Audio* layer contains audio and video recordings.

It is worth to stress the presence of synchronizable and non-synchronizable objects within a single MX file. Audio, video and still graphics usually belong to the former family, whereas catalog metadata fall into the latter. Where a number of homogeneous or heterogeneous synchronizable objects are available for a given music piece, MX implements mechanisms to provide full synchronization. In this way, it is possible to enjoy music in a highly integrated environment where a cursor highlights the current chord in the score and simultaneously the corresponding point in an audio track is playing. Similarly, it is possible to switch from a score version to another, or from an audio performance to another in real-time, while the music is being played. Of course, this feature is not available for non-synchronizable objects such as metadata (track title, author, …) or music-related material (on-stage photos, sketches, playbills, …).

4. DESIGN OF THE INTERFACE

The design of the interface follows guidelines which strictly depend on the definitions and the key issues provided in the previous sections.

First, heterogeneity in music contents should find a counterpart in the layout of controls and views. Players, panels, floating windows or other devices should be used to present multimedia contents in a unique framework. A simple way to view and navigate complex contents consists in keeping different multimedia types separated by using different controls, and grouping a number of objects of the same type within the same control. As a matter of fact, homogeneous media types require similar controls and imply similar behavior, so this approach proves to be both user-friendly and effective. For instance, the part of the interface dedicated to audio/video contents should contain the playlist of such media objects (dynamically loaded from the MX file) and the usual controls of a media player. On the contrary, the panel dedicated to score images should contain the list of scores and pages of each score (dynamically loaded from the MX file) and image-oriented navigation tools. Please note that the simultaneous presence of all the six layers is not required for a generic music piece: a jazz piece could present no traditional score, as well as a never performed music work could be described from a symbolic point of view only, without any media attached. As a consequence, also the corresponding controls of the interface should be dynamically shown or hidden according to the characteristics of the encoding.

Besides, our comprehensive approach assigns the same dignity to all the forms of music description, thus the interface should present no “privileged” media type. Nevertheless, from a user-oriented standpoint, it is preferable to have a main window where a given media is shown with greater evidence. A solution to this dichotomy could be allowing any media to be played on the main window, as well as to be resized to a secondary view panel.

Full synchronization among synchronizable objects should be provided. As a consequence, the interface should allow the simultaneous enjoyment of all the views involved in the representation of media objects. A problem could occur with objects belonging to the same media category: for instance, following music simultaneously on pages belonging to different score versions could be difficult, but not impossible for a human user; on the contrary, listening to many performances, each with its own absolute temporization of music events, is confusing and difficult to implement. Finally, also non-synchronizable descriptions should be accessible, but in this case layout requirements are less problematic.

As stated before, a comprehensive music description can be performed by using six layers. In the following, we will discuss the design of an interface through a layer-by-layer approach.

4.1 The General layer

In this articulate approach to music description, the *General* layer contains catalogue metadata and additional information about the music work. Music entities are not directly involved here, neither as graphical nor as audio objects; moreover, the *General* layer does not convey symbolical or structural information. This section mainly contains metadata such as the title of the piece, the name of its authors and their role, the catalogue number (if any) and so on. The type of information presented here is typically text-based and non-synchronizable, thus *ad hoc* players are not required. Usually, titles and authors are shown in the title bar of the main window, and – if other information is required – a popup control could provide a more detailed view. In addition to catalogue metadata and other text contents, the *General* layer could host the description of digital objects which do not describe music itself but related aspects: sketches and fashion plates for an opera, on-stage photos and playbills for a rock concert, etc. In this case, an *ad hoc* player is required, above all if related contents should be somehow synchronized to music events. For example, during a duet between opera soloists, the photos of the performers or the pictures of their characters could appear as soon as they start singing and disappear when they stop, providing a graphical representation of the duet. A comprehensive interface for music enjoyment should support even this sort of “music-driven slideshows”.

4.2 The Logic layer

In the *Logic* layer, music contents are represented in symbolic format. As regards traditional pieces, this section contains the description of notes, rests and other music signs. The encoding could be text-based (e.g. DARMS, GUIDO, Plaine and Easie Code), binary (Sibelius 5, MakeMusic Finale 2008, NIFF), or XML-based (e.g. MusicXML, MusiXML, MX). Please note that no absolute information about timing is provided here, as this aspect depends on specific performances and it is treated in the *Audio* layer. The *Logic* layer can be displayed by parsing the symbolic description and providing the corresponding layout. This latter aspect could imply a traditional representation – namely a score – or a “revised” version aimed at stressing inner properties of the piece. Examples of non-traditional representations are the typical view of note pitch and duration

provided by MIDI sequencers, and experimental layouts coming from musicological studies [4].

4.3 The Structural layer

The *Structural* layer investigates the musicological aspects of the work. In this section, music objects can be identified and put in relationship in order to justify the architecture of the piece. The locution *music object* is intentionally vague, so that any aggregation of music entities with distinctive features and a common meaning can fall under this definition. For instance, chords can be considered music objects built as vertical sets of notes, and their structural relationships produce harmonic grids and harmonic paths. Other examples of music objects are represented by melodic sequences, which can constitute themes and music subjects; in this case, discovering their relationships brings to considerations about the form and the architecture of the work, at various degrees of detail. A set of interfaces could be designed to underline structural relationships and to make them evident to the untrained user. An easy-to-implement proposal consists in using colors and geometrical shapes over traditional scores to characterize recurrent music objects or peculiar behaviors of parts and voices.

4.4 The Notational layer

The *Notational* layer contains information about graphical scores in form of digital objects. In a standard environment aimed at music enjoyment, this section is fundamental: in fact, it provides graphical contents to be synchronized. When traditional scores are involved, their scans can be described here, so that the user can follow symbolic contents within the interface. A more interesting representation can take place when no traditional score is available. In this case, a more general approach to music description and representation, based on graphical objects instead of symbolic representations, can solve the score-following problem. The *Notational* layer typically describes synchronizable music objects: thanks to its absolute timing, any audio performance can drive a cursor that points the corresponding music symbols over the score. In notation software, the cursor is usually represented as a vertical line that embraces all the staves of a system and moves along the horizontal axis. However, a more detailed view can be provided by drawing a number of bounding boxes around the current events, so that: i) also vertical movements of melodic lines can be appreciated, and ii) score following can be filtered by parts/voices.

4.5 The Performance layer

The *Performance* layer contains symbolic codes aiming at computer-driven music performances. Once again, the materials that fall under this category belong to the family of synchronizable objects. Examples of in-use digital formats for performance information are MIDI and SASL/SAOL. Their nature intrinsically allows a number of different representations. If we consider the case of a MIDI file, usually it is performed in a media player, so that the user can enjoy its contents as a waveform; as a matter of fact, the original file actually does not contain audio information, but numeric instructions to allow a synthesizer (or a more complex audio chain) to produce audio information. From another standpoint, many MIDI-related software applications try to provide a symbolic view of MIDI contents, by interpreting pitches (namely frequency-based classes) and durations (expressed in MIDI ticks) in order to recreate traditional scores. Finally, software such as sequencers show MIDI information by some kind of graphical representation, like colored rectangles and circles disposed on a grid. All these forms represent effective visualizations for computer-driven performance information, and our comprehensive approach should take them into account. In the interface we propose, the notational approach (even if the score is not expressed, but somehow recreated) or the audio performance (even if the waveform is generated by some MIDI instrument) can rely on the dedicated players provided for the corresponding layers, namely the *Notational* and the *Audio* layers. The innovative contribution of this layer could be providing an alternative representation of music data, such as depicting chords as geometrical shapes, assigning given colors to pitch families.

4.6 The Audio layer

In general terms, the *Audio* layer addresses music tracks encoded in some digital format. First, music events in an audio file are synchronizable objects; moreover, from the implementation point of view, such contents drive the synchronization of the other layers, as they have the most strict temporization requirements. A basic interface to listen to music should provide the standard controls to play, pause and stop the current track. Other features, such as the possibility to adjust the volume, should be provided too. The same interface, with *ad hoc* extensions, can be adopted to include also video contents: e.g. videoclips, live concert recordings, movie scenes with a soundtrack, 3D-animations, etc.

5. A PROPOSAL FOR AN INTERFACE

After discussing the levels of abstraction and the different perspectives in music description, and after introducing a number of features for a comprehensive music-oriented visualization, now a generic implementation of the interface will be proposed and shown in Figure 1. The format for music files is MX, which has been described in Section 3.

In our approach, the layout is made of a number of floating panels which can be either enabled or disabled depending on the features of the piece, and can be either opened or closed depending on the user's needs. Each panel is dedicated to a specific kind of visualization. All the homogeneous objects, namely the material belonging to the same layer, are selectable in each panel designed to reproduce them. In a simple case, we could assume a 1:1 relationship between music layers and the corresponding panels. However, this would limit the representational capabilities of the interface for the following reasons. First, even a single layer can host various kinds of information. For example, the *Logic* layer describes music symbols (such as notes and rests) as well as lyrics; and the *Structural* layer can contain harmony grids as well as musicological analyses of music forms. The coexistence of such heterogeneous aspects within a single viewer would be difficult to design and implement. Besides, the same information can be described and represented in different ways. For instance, the performance data contained in a MIDI file can originate both a traditional score and an audio rendition; and an audio track itself can be played but also visualized through some graphic algorithm. As a consequence, the interface we propose includes more than the six players related to the corresponding layers. The goal is to fulfill all the user's needs and desires about a comprehensive

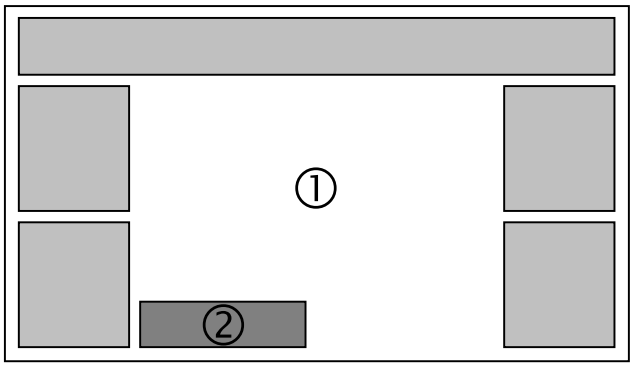

Figure 1. A proposal for the interface.

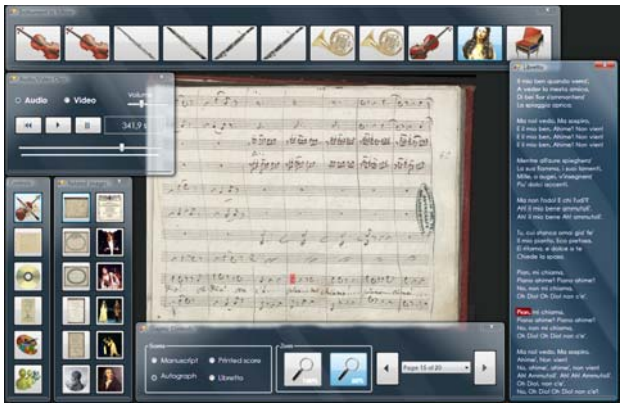

Figure 2. An implementation of the proposed interface.

representation of music and music-related information, both for a professional use or for untrained people.

Two panels are fundamental for the interface: the *main monitor*, denoted by ① in Figure 1, and the *meta-panel*, identified by ②. The purpose of the main monitor is to play graphic or audio/video materials in full-screen mode, thus allowing a better enjoyment of the contents. Among the standard uses of the main monitor we can cite showing the score with a running cursor, playing a related videoclip or providing other graphical representations of music contents, including slideshows. The meta-panel contains the controls to enable and show the other panels, when available. When the application is launched, the default behavior is the following: the MX file is parsed, and all the panels related to available materials are opened. Thus, the presence of each layer, and of the corresponding instruments, depends on the characteristics of the piece. Besides, the user should be allowed to close any panel, but the main monitor and the meta-panel, at any time.

Finally, let us recall the discussion about synchronizable and non-synchronizable objects. On the one side, the latter objects never conflict, which means that opening *n* panels with this kind of information do not generate errors or abnormal behaviors. On the other side, synchronizable objects should be accurately studied, in order to understand which situations bring to implementation or fruition problems. The possibility to view and play simultaneously contents from the same layer or from different layers derive from such considerations. For example, it is possible to follow two or more score versions simultaneously, even if the score scans all belong to the same layer. On the contrary, two simultaneous audio tracks can not be managed together, unless the timing characteristics of one of them is adjusted. Please note that even heterogeneous descriptions, coming from different layers, could conflict: this is the case of playing a MIDI file, described in the *Performance* layer and having its own event timing, together with an MP3 file, coming from the *Audio* layer.

6. CONCLUSIONS

The generic interface defined in this article can provide guidelines towards an integrated enjoyment of music. The proposed application represents the evolution of a number of earlier software demos and working applications developed by the LIM staff, as documented in [5] and [6]. The most recent implementation is N.I.N.A., standing for *Navigating and Interacting with Notation and Audio*. The interface is shown in Figure 2. The music piece chosen for this demonstration is the operatic aria "Il mio ben quando verrà", from Giovanni Paisiello's *Nina, o sia la pazza per amore*. This software was designed and implemented for the exhibition "Napoli, nel nobil core della musica" held in May 2007 at ResidenzGalerie in Salzburg, Austria. One of the purposes of the exhibition was that of making music tangible and visible, bringing together all five senses, beyond hearing.

7. ACKNOWLEDGMENTS

Our thanks to the LIM staff, who has contributed to the development of the MX standard and to the design of a number of XML-based music-oriented applications. The authors want to acknowledge researchers and graduate students at LIM, and the members of the IEEE Standards Association Working Group on Music Application of XML (P1599) for their cooperation and efforts. Special acknowledgments are due to Denis Baggi and Goffredo Haus for their invaluable work as working group chair and co-chair of the IEEE Standard Association WG on MX.

8. REFERENCES

[1] Baggi, D. 1995. Technical Committee on Computer-Generated Music. In: Computer, vol. 28, no. 11, 91-92.

[2] Baratè, A., Haus, G., and Ludovico, L.A. 2007. Music representation of score, sound, MIDI, structure and metadata all integrated in a single multilayer environment based on XML. Intelligent Music Information Systems: Tools and Methodologies, Idea Group Reference.

[3] Haus, G., and Longari, M. 2005. A Multi-Layered, Time-Based Music Description Approach Based on XML. Computer Music Journal, vol. 29, no. 1, 2005, 70-85.

[4] Hsu, K.J., and Hsu, A.J. 1990. Fractal Geometry of Music. Proceedings of the National Academy of Sciences of the United States of America, Vol. 87, No. 3, 938-941.

[5] Baratè, A., Haus, G., and Ludovico, L.A. 2006. An XML-Based Format for Advanced Music Fruition. Proceedings of the Sound and Music Computing Conference 2006 (SMC06), Marseille, France.

[6] Baggi, D., Baratè, A., Haus, G., and Ludovico, L.A. 2007. NINA - Navigating and Interacting with Notation and Audio. Proceedings of the 2nd International Workshop on Semantic Media Adaptation and Personalization (SMAP 2007), London, Great Britain.