MX Navigator: An Application for Advanced Music Fruition

Adriano Baratè, Goffredo Haus, Luca A. Ludovico
LIM – DICo
Università degli Studi di Milano
Via Comelico, 39
20135 Milano, ITALY
+39 02 50316382
{barate, haus, ludovico}@dico.unimi.it

Abstract

This paper describes an application that allows an advanced fruition of music contents. Thanks to our music XML format, namely MX (IEEE PAR1599), and to the interface provided by such application, users can enjoy music from different points of view: it is possible to navigate different scores, video and audio performances of the same piece; besides, each content type can be used as the leading media to keep all the other related contents synchronized. The working application has been installed at the exhibition “Tema con Variazioni” (December 2005 – January 2006), organized by the Italian Ministry of Cultural Heritages and held in Rome.

1. Introduction

Creating a visual interface to enjoy music is a challenging matter. First, it is necessary to identify the means, the purposes, and the receivers of such form of communication. The visual interface we will present in this paper, namely MX Navigator, is a computer-based application for a “comprehensive” representation and enjoyment of music and it addresses to musicologists, performers, students and people simply interested in music.

The key characteristics of MX Navigator can be summarized as follows:

• Richness in the multimedia descriptions related to the same music piece (graphical, audio, and video contents).
• Possibility to link and perform a number of media objects of the same type (for instance, many performances of the same piece or many score scans coming from different editions).
• Complete synchronization among time-based contents, meaning that audio and video contents are kept synchronized with score advancing, even when the user switches from a particular performance to another, or from a particular score edition to another.
• Interaction, that is the possibility for users to click any point of the score and jump there also in the audio content currently performed, as well as the possibility to navigate the audio track moving the related slider control and highlighting the related portion of the score.

MX Navigator was installed at the exhibition “Tema con Variazioni. Musica e innovazione tecnologica” [Theme with variations. Music and technological innovation], a voyage into the Italian musical heritage through the rooms of Rome’s Music Park Auditorium. One of the purposes of the exhibition was making music tangible and visible bringing together the five senses, not just hearing. In this context we have designed a simple user interface, conceived for musically untrained people, in order to listen to and visualize a track alongside variously interpreted scores.

MX Navigator represents the natural evolution of MX Demo [1], an experimental prototype presented in a scientific research context at AXMEDIS 2005 conference. The main differences between MX Demo and MX Navigator are two:

• The latter is a generalized version of the former, as MX Demo was designed only to demonstrate MX format possibilities and worked on the limited number of pieces consequently chosen, whereas MX Navigator is virtually able to open any MX

1 MX is an XML-based format for music, currently under development at LIM and described in detail in Section 3 of the paper.
file. It is able to adapt itself in real time to the different kind and number of media contained.

- MX Navigator is necessarily characterized by an improved usability and by a simpler user interface. In fact, MX Demo had to be presented to a scientific audience by its authors, while MX Navigator is at common users’ disposal in a public exhibition.

2. Phases of the project

The process of designing and implementing a software solution that responds to the aforementioned requirements is made of a number of steps.

First, it is necessary to conceive of music description in a comprehensive way. In order to appreciate the richness of music communication, let us recall that such language is made up of many different and complementary aspects: music – in its most general meaning – can stimulate the sense of hearing, the sense of sight and even the sense of touch. In other words, music can be (and actually is) the idea that the composer translates to symbols as well as their performance, the printed score that musicians read as well as a number of other related contents. A complete analysis of music richness and complexity is provided in [2], where six different levels of music description are identified – namely General, Structural, Logical, Notational, Audio, and Performance layers. This multilayer structure, suitable for our concept of comprehensive description of music, is reflected by the encoding format we are developing at LIM.2 Our goal is providing a highly integrated performance of music, where score, audio, video, and related graphical contents can be enjoyed together.

A second step is acquiring all the contents related to a particular music piece and encoding them in a suitable format. As we will soon explain, the file formats we employ are the commonly accepted ones. The only novelty is represented by the introduction of our XML-based format, described in the next section, in order to link and synchronize the different contents. In order to simplify the assemblage process of heterogeneous contents within a single music description, we had to develop a suite of software applications, and this will be the subject of another section of the paper.

Finally, the third step is the design and implementation of MX Navigator, the application that allows an integrated and evolved navigation of music contents. MX Navigator presents all the aspects we cited before: richness in multimedia description of music, synchronization among heterogeneous contents and easy user interaction.

3. MX format for music representation

The encoding formats commonly accepted and employed for music description are often characterized by an intrinsic limitation: they describe music data or metadata for score, audio tracks, computer performances of music pieces, but they seldom encode all these aspects together. Many encoding formats, aimed at a precise characterization of different specific music aspects, have been proposed. For example, AAC, MP3 and PCM formats encode audio recordings; MIDI represents a well known standard for computer-driven performance; JPEG and TIFF files can contain the results of a scanning process of scores; Finale, NIFF and Sibelius formats are aimed at score typing and publishing, and so on.

In order to integrate all the aspects of music within a single description, we are developing a new XML-

![Figure 1](image_url)

Fig. 1. The role of spine in MX.

---

2 Laboratorio di Informatica Musicale (LIM), Dipartimento di Informatica e Comunicazione (DICo), Università degli Studi di Milano.
based format, called MX. Currently, MX is undergoing the IEEE standardization process (IEEE PAR1599), as described in [3]. Our approach is different from other kinds of music encodings, in particular because we represent music information according to a multi-layer structure and to the concept of space-time construct. Now we will explain these key concepts in greater detail.

The key feature of MX format is the multi-layer structure. We can conceive each layer as a different degree of abstraction in music information. For a common and exhaustive description, in MX we distinguish the Structural, Music Logic, Notational, Performance and Audio layers. This multi-layered description allows MX to import a number of different formats aimed at music encoding without modifying such commonly accepted encodings. For example, MusicXML [4] could be integrated in our format to describe score symbolic information (e.g., notes and rests), whereas other common file types such as JPEG and TIFF for notational layer, MP3 and WAV for audio layer can be linked to represent other facets of music.

The second peculiarity of the MX format is the presence of a space-time construct, called spine. Considering music as a multi-layered information, we need a means to link and synchronize the heterogeneous facets that compose such information. To this end, we introduced the concept of spine, namely a structure that relates time and spatial information (see Figure 1). Through such a mapping, it is possible to fix a point in a layer instance (e.g. Notational) and jump to the corresponding point in another one (e.g. Performance or Audio). Each layer containing music data refers to the events listed in the spine, and this provides a global internal synchronization. Of course, some data intrinsically cannot be synchronized; it is the case of general metadata or related graphical files (photos, sketches, pictures) which have no relationships towards score representation and performance.

Now, let us briefly describe the meaning and the contents of each MX layer.

The General layer is mainly aimed at expressing catalogue information about the piece. In addition, data about casting, and digital or analogue media objects not directly related to music content can be described. This layer simply contains some basic alphanumeric information about the coded music work. Finally, there exist sub-elements devoted to annotations and digital rights management.

The Logic layer contains information referenced by all other layers, and it represents what the composer intended to put in the piece. It is composed of two elements: i) the Spine description, used to mark the significant events in order to reference them from the other layers and ii) the LOS (Logically Organized Symbols) element, that describes the score from a symbolic point of view (e.g., chords, rest).

The Structural layer contains explicit descriptions of music objects together with their causal relationships, from both the compositional and the musicological point of view. It represents how music objects can be described as a transformation of previously described music objects.

The Notational layer links all possible visual instances of a music piece. Representations can be grouped in two types: notational and graphical. A notational instance is often in a binary format, such as NIFF or Enigma, whereas a graphical instance contains images representing the score. Usually, the latter is in a binary format too (e.g., a JPEG image or a PDF file), but it can also be a vector image encoded in SVG. The information contained in this layer is tied to the spatial part of the Spine structure, allowing its localization.

The Performance layer lies between notational and audio layers. File formats grouped in this level encode parameters of notes to be played and parameters of sounds to be created by a computer performance. This layer supports symbolic formats such as MIDI, Csound or SASL/SAOL [5] files.

Finally, the Audio layer describes properties of the source material containing music audio information. It is the lowest level of the layered structure. The complete Document Type Definition (DTD) of MX 1.5 format is available at http://www.lim.dico.unimi.it/mx/mx.zip, together with a complete example of music representation by means of MX. As regards the music piece MX Navigator is based on (see Section 5), a short example is shown in Figure 2.

4. MX utilities

The process required to map and synchronize heterogeneous media at the moment cannot be performed automatically. This would require, for instance, a good OMR\(^3\) system in order to recognize musical symbols in scores, even when autographical. Besides, such symbols should be automatically put in relationship with the spine structure. As regards audio information, an effective application to extract automatically music events from a complex audio track

\(^3\) OMR stands for Optical Music Recognition.
<xml version="1.0" encoding="UTF-8"/>
<!DOCTYPE mx SYSTEM "http://www.lim.dico.unimi.it/mx/mx.dtd"[]>
<mx>
  <general>
    <movement_title>Tosca - Act III - E lucevan le stelle</movement_title>
  </general>
  <related_files>
    <related_file file_name="playbill_1.jpg"/>
    <related_file file_name="playbill_2.jpg"/>
  </related_files>
  <logic>
    <spine>
      <event id="timesig_in" timing="0" hpos="0"/>
      <event id="intro_0" timing="0" hpos="0"/>
    </spine>
  </logic>
  <notational>
    <graphic_instance_group description="Autographical Score">
      <graphic_instance file_name="autogr_1.tif" format="image_tiff"/>
    </graphic_instance_group>
  </notational>
  <audio>
    <clip>
      <clip_format file_name="Tosca1984.wav" file_format="audio_wav"/>
    </clip>
  </audio>
</mx>

Fig. 2. An example of MX encoding.
should be employed. Some limits in automated music analysis techniques prevented us from reaching this ambitious goal: at the moment, an effective algorithm to perform $n$ voices demixing from a stereo recording is unknown.

On the other hand, acquiring music events by a completely hand-made process would be a terrible waste of time and energy: it would require, for instance, an accurate listening of the audio tracks in a sound editing environment, or the precise computation of the “bounding box” around each music event in a digital imaging software. Calculating milliseconds and pixels by hand is not effective nor efficient.

Our solution was designing and implementing some aiding applications to speed up the mapping process. In particular, two applications were released and employed to feed MX Navigator with a rich MX file: MX Graphic Mapper and MX Audio Mapper.

MX Graphic Mapper, shown in Figure 3, is the application devoted to link the logic events of an MX file to their corresponding graphic counterparts within the score image(s). An MX file is opened, scanned and a list of all notes/rests events is created. The user loads one or more images that contain the score and begins mapping all the events by drawing rectangles in the central window by the well-known drag and drop technique. The representation of the note/rest event is graphically shown in the “Event Graphic Parser”, and the related XML fragment to be added is visible inside the upper part of the interface. When a rectangle is created, the application generates the corresponding XML line by reading the current event in the “Spine Elements” listbox and by computing the coordinates of the rectangle. When a new line is added, the current element indicator is moved forward and the mapping process can continue.

MX Audio Mapper is the application used to compute the time indexes of events in audio/video clips. This application is similar to the previous one, but instead of mapping the graphic representations of a
piece (scores), it maps its audio/video performances (clips). After opening the MX and the audio/video file, some parameters have to be defined, namely the timing unit per quarter used in the logic layer and the rhythm figure to be processed (crotchet, quaver, semi-quaver, etc.). The mapping process is performed by “tapping” the rhythm on a button: when this process is completed, the MX spine is processed and all the timings of the events are computed (in seconds), interpolating figures between two consecutive “taps”. After this preliminary procedure, the computed mappings can be fine-tuned by hearing the selected time positions in the clip and adjusting the wrong timings.

As a consequence, our way to compile an MX file is not completely automated nor completely hand-made: it is a compromise that could be defined as a “semi-automated solution”, as human intervention is still fundamental, but the computer plays an important role too.

5. MX Navigator

As stated before, MX Navigator allows an integrated enjoyment of different media contents related to the same music piece, namely “E lucevan le stelle…” from Puccini’s Tosca – Act III. The choice of that aria was imposed by the leitmotiv of the exhibition: in fact, “Tema con Variazioni” was based on different representations of the same score.

Apart from a number of graphical objects such as greeting cards, playbills, historical photos and sketches, the software is based on two versions of the score (an autographical and a printed one), on a video and two audio performances of the aforementioned Puccini’s aria.

The interface is illustrated in Figure 4. Its central part contains the score, as this is the main media in terms of visual extent and interaction. The upper left part of the window contains the controls related to audio/video interaction. This application shows three different clips: an audio clip of a 1953 version performed by the Italian tenor Di Stefano and both an audio and video clip of the 1984 version performed by Aragall at Verona’s Arena. When one of the aforementioned versions is chosen to be played, the music and/or video is executed. In the score window, a red rectangle indicates the event currently played in the clip; the data used to achieve synchronization are the results coming from MX Audio Mapper and MX Graphic Mapper.

Since there are many parts and voices in the score and they are all coded into the MX file, it is virtually
possible to select and follow any instrumental part. However, this application is intended to be used even by non-musicians, so the choice among all the possible instruments was intentionally simplified, and only two parts can be selected. In fact, in the upper part of the interface there are two buttons that control which part has to be followed by the score indicator: Clarinetto or Tenor (Cavaradossi).

In the upper section of the interface, the controls are devoted to managing score visualization. For this MX file there are only two types of mapped scores: the autographical version by Puccini and the version published by Ricordi editions. The upper left buttons switch (even when an audio/video clip is running) between these two graphical instances. The upper right buttons control the zoom value (50% or 100%) and the current page displayed. This happens only in the Pause state, when the graphical representation of the score is not synchronized with music; otherwise, the current page is automatically selected to follow the audio/video clip execution.

The left bottom section of the interface is used to open other graphical elements related to the piece, such as sketches or photos. Of course, those objects are linked from the current MX file, but they are not synchronized to the other multimedia contents.

This visual interface allows a number of different ways to enjoy music. First, it is possible to select a score version, an audio track, a leading instrument and simply follow the evolution of the instrumental part. This is a first way to explore a music piece, as music can be listened and watched in a synchronized fashion. But a second way to enjoy music through MX Navigator is even more interesting: it consists in switching from an aural/visual representation to another. In other words, it is possible to compare in real time different versions of the score (the hand-made and the printed one) or different performances. When the user decides to switch from a representation to another, MX Navigator continues from the point previously reached. Finally, the application suggests a third way to enjoy music, that consists in altering the original sequence of music events. It is possible to jump – forward or back – from a point to another point of the score, both in its visual and aural representations; of course, the effect will be the same as the former and the latter are synchronized.

6. Concluding remarks

The MX Navigator version we presented at the exhibition “Tema con Variazioni” is an interface conceived for the spread of art and music among people not necessarily cultured. The application is characterized by a comprehensive multimedia description of music, synchronization among heterogeneous contents and easy user interaction.

We think that MX Navigator represents an important experiment in the field of cultural heritages, as users can enjoy music also in non-traditional ways and they can intuitively interact with musical contents.

7. Acknowledgments

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 (PAR1599) for their cooperation and efforts. Special acknowledgments are due to: Denis Baggi for his invaluable work as working group chair of the IEEE Standard Association WG on MX (PAR1599); Maria Pia Ferraris and Cristiano Ostinelli (Ricordi) for their fundamental contributions as regards the original material (autographic and printed scores, sketches, pictures, photos) used in MX Navigator.

8. References


