A Computer Tool to Visualize Score Analysis

Adriano Baratè, Luca A. Ludovico, and Alberto Pinto
Laboratorio di Informatica Musicale (LIM)
Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39/41 – I-20135 Milano, Italy
{barate, ludovico, pinto}@dico.unimi.it

Abstract. This paper deals with a software application to visualize score analysis. Such tool allows both the enjoyment of music in general, thanks to the synchronized execution of heterogeneous multimedia objects, and the highlighting of the results of score analyses. As a consequence, the user can investigate the compositional process and the relationships among the music objects of a given piece while listening to an audio track and viewing the corresponding score. This is made possible by the new IEEE XML-based standard for music representation, which will be shortly described.

1 Score Analysis and its Visual Representation

Musical analysis embraces all those methods aiming at the comprehension of music processes and relationships within a composition. Of course, many different approaches have been employed and are in use to analyze a score, ranging from the traditional methods used in the past centuries to more recent techniques such as the functional analysis by Hans Keller and the Schenkerian analysis by Heinrich Schenker.

A number of ongoing researches are aimed at the analysis of music information through algorithmic processes, in order to obtain automatic segmentation, recognition of recurrent patterns, statistical data, etc. However, the problem of how such results can be obtained, either in a hand-made or computer-based way, is not pertinent to this work.

Rather, we concentrate on the computer-based techniques to produce and enjoy the visual representation of musical analysis. The recipients of this kind of applications include musicologists, music students and untrained people interested in understanding music mechanisms.

Thanks to the application we will introduce in the next sections, we propose a new way to investigate music processes. The results coming from an analysis are presented to the user synchronized with a number of multimedia objects which describe the piece from other points of view. Thus, analysis is not a static description, but it is explained while the score is advancing and the corresponding audio track is playing. As a consequence, an overall comprehension of music can be provided, as regards its symbolic, structural, audio and graphical aspects.
The problem we address has two faces: i) a phase of preparation which includes the encoding of the piece, the recognition of music structures to highlight, the assembly of multimedia objects referable to the given piece, and finally the choice of a format to describe all this information within a unique framework; ii) the design and implementation of interfaces to enjoy music according to the key concepts we have introduced.

Even if the choice of a format to represent music and all the related materials is not the main subject of the present work, it should be evident that only a standard with some distinctive characteristics can provide an integrated representation of heterogeneous music contents. Consequently, such *ad hoc* standard is the base for an evolved interface to enjoy music and the corresponding structural information. Needless to say, popular standards for symbolic scores, graphics, audio, video already exist and are commonly used. Nevertheless, their use for a “comprehensive” description of music presents some disadvantages, as follows:

1. Data and metadata cannot be synchronised across different formats, as they were not conceived for this purpose. For instance, if TIFF is selected to encode the pages of a score and AIFF is used for the corresponding performances, there is no way to keep information synchronised, unless a third format is used as a wrapper. This problem is even more evident when the user also wants to synchronise structural and musicological analyses, computer-driven performances and textual information such as lyrics.
2. Often music contents are encoded in binary form, unreadable for humans, proprietary and strongly dependent on the current technology.

In our opinion, the new IEEE standard known as MX can solve the above-mentioned problems. The general approach and the characteristics of this format are described in the next section.

2 MX, a New Standard for Music

MX is a new IEEE standard to describe comprehensively heterogeneous music contents. Its development follows the guidelines of IEEE P1599, *Recommended Practice Dealing with Applications and Representations of Symbolic Music Information Using the XML Language* [1].

MX is the result of research efforts at the *Laboratorio di Informatica Musicale*, or LIM, of the *Università degli Studi di Milano*. The most recent version of MX DTD and documentation can be downloaded from http://www.mx.dico.unimi.it.

In a single MX file, music symbols, printed scores, audio tracks, computer-driven performances, catalogue metadata, text and graphic contents related to a single music piece are linked and mutually synchronised within the same framework. Heterogeneous contents are organised in a multilayered structure that supports different encoding formats and a number of digital objects for each layer [2].

Tools for music visualisation [3], content-based retrieval [4], and automatic segmentation [5] are currently available.
MX is an XML-based format: its name is an acronym which stands for Musical application using XML. There are many advantages in choosing XML to describe information in general, and music information in particular. For instance, an XML-based language allows inherent readability, extensibility and durability. It is open, free, easy to read by humans and computers, and can be edited by common software applications. Moreover, it is strongly structured, it can be extended to support new notations and new music symbols, and it can thus become a means of interchange for music with software applications and over the Net. Most of these topics have been treated in [6] and [7].

A comprehensive description of music must support heterogeneous materials. MX employs six different layers to represent information, as explained in [8]:

- **General** – music-related metadata, i.e. catalogue information about the piece;
- **Logic** – the logical description of score symbols;
- **Structural** – identification of music objects and their mutual relationships;
- **Notational** – graphical representations of the score;
- **Performance** – computer-based descriptions and executions of music according to performance languages;
- **Audio** – digital or digitised recordings of the piece.

Not all layers must, or can, be present for a given music piece. For instance, the encoding of a piece does not require to describe all its structural, notational, performance or audio aspects. In some cases, even the symbolic score could not be defined, as it does not exist (such as in jazz improvisation on a harmonic grid) or it is not relevant for the user purposes (as in score following, where only the relationships between the Notational and the Audio layers must be encoded).

Not only a number of heterogeneous media descriptions is supported, but each layer can also contain many digital instances. For example, the Notational layer could link to several notated versions of the score, and the Structural layer could provide many different analyses for the same piece. The concept of multi-layered description – as many different types of descriptions as possible, all correlated and synchronised – together with the concept of multi-instance support – as many different media objects as possible for each layer – provide rich and flexible means for encoding music in all its aspects.

It should be clear that the description provided by an MX file is flexible and rich, both in regard to the number and to the type of media involved. In fact, thanks to this approach, a single file can contain one or more descriptions of the same music piece in each layer. For example, in the case of an operatic aria, the MX file could house: the catalogue metadata about the piece, its author(s) and genre; the corresponding portion of the libretto; scans of the original manuscript and of a number of printed scores; several audio files containing different performances; related iconographic contents, such as sketches, stage photographs, and playbills. Thanks to the comprehensive information provided by MX, software applications based on such a format allow an integrated enjoyment of music in all its aspects.
The spine, the second key concept of the MX format, consists of a sorted list of events. This structure provides both an abstraction level and the glue among layers, thus representing an abstraction level, as the events identified in it do not have to correspond to score symbols, or audio samples, or anything else. It is the author who can decide, from time to time, what goes under the definition of music event, according to the needs: the definition and granularity of events can be chosen by the author of the encoding.

Since the spine simply lists events to provide a unique label for them, the mere presence of an event in the spine has no semantic meaning. As a consequence, what is listed in the spine structure must have a counterpart in some layer, otherwise the event would not be defined and its presence in the list (and in the MX file) would be absolutely useless.

For example, in a piece made of \( n \) music events, the spine would list \( n \) entries without defining them from any point of view. If each event has a symbolic meaning (e.g. it is a note or a rest), is graphically rendered in many scores and is relevant to a number of analyses, these aspects are treated in the Logic, Notational, and Structural layers respectively.

Music events are not only listed in the spine, but also marked by unique identifiers. These identifiers are referred to by all instances of the corresponding event representations in other layers. Thus, each spine event can be described:

- in 1 to \( n \) layers; e.g., in the Logic, Notational, and Audio layers;
- in 1 to \( n \) instances within the same layer; e.g., in three different score versions mapped in the Notational layer;
- in 1 to \( n \) occurrences within the same instance; e.g., the notes in a song refrain that is performed 4 times (thus the same spine events are mapped 4 times in the Audio layer, at different timings).

Thanks to the spine, MX is not a simple container for heterogeneous media descriptions related to a unique music piece. It shows instead that those descriptions can also present a number of references to a common structure. This aspect creates synchronisation among instances within a layer (intra-layer synchronisation), and – when applied to a complex file – also synchronisation among contents disposed in many layers (inter-layer synchronisation).
3 Application Interface and Features

After introducing the concept of visual representations of score analysis (see Section 1) and discussing the layer approach of MX to describe different levels of abstraction in music description (see Section 2), now we present applications that join together these two aspects.

Since the main concept of this work is defining a simple method to represent graphical annotation and to share them, the best choice is using fewer textual annotation and more graphical figures. With this approach, the visual analysis is not dependent by the language and presents also a “visual intuitiveness” even for musicology untrained people.

Before presenting our proposed methodology, a brief digression regarding the MX structural elements involved in our approach is necessary. As presented in Section 2, the layer that houses description of structural considerations is the Structural one. In this layer various methods can be used to identify a music object, that is selecting a relevant portion of the music piece. For our purposes we have chosen to use the segmentation element, together with its sub-elements, for their simplicity and scalability. In the MX DTD these elements have the following declaration:

```xml
<!ELEMENT segmentation (segment+)>  
<!ATTLIST segmentation
   id ID #IMPLIED
   description CDATA #IMPLIED
   method CDATA #IMPLIED>

<!ELEMENT segment(segment_event+, feature_object*)>
<!ATTLIST segment
   id ID #REQUIRED>

<!ELEMENT segment_event EMPTY>
<!ATTLIST segment_event
   event_ref IDREF #REQUIRED>

Code 1. The fragment of the MX DTD devoted to segmentation element.
```

In the main element, it is possible to define a method for the segmentation process and to add a description in textual form. Within the segmentation element, a number of segments involved in the current analysis must be declared, specifying for each segment all its concerning spine events. The feature_object element is custom implemented for a particular segmentation.

In Code 2 an example is presented to better understand analysis declarations. Only a segmentation is shown, while it is possible to add as many analyses as the like.

```xml
<segmentation id="segm01" description="Analysis n.1">
   <segment id="s01">
      <segment_event event_ref="p1_meas001_ev01" />
      <segment_event event_ref="p1_meas001_ev03" />
   </segment>
</segmentation>
```

Code 2. An example of analysis declarations.
The visual analysis annotation process is made by the following steps:

- definition of one or more score segments to which the annotation is related; this is done by including one or more `segment` elements in the Structural layer;
- definition of the annotation itself, by inserting symbols and/or drawings such as lines, rectangles, and other figures over a score graphical rendition, i.e. a scan or a computer representation. In MX visual annotations are stored in `feature_object`, a customizable element that in this example is declared as an `svg` (Scalable Vector Graphics) object.

The application developed to edit visual annotations is presented in Figure 2. When the application starts and an MX file is opened, graphical representations of the encoded music piece can be selected; then the application shows the score together with a number of boxes that mark the existing notes, pauses and in general every music symbol described in the Logic layer. By a pointing device, the user is allowed to annotate the score iterating the following procedure:

- a visual annotation session is opened;
- the boxes containing the notes involved in the annotation are selected;
- one of the graphic tools is selected;
- the graphical annotations are drawn;
- the current annotation session is closed.

The graphical tools provided for drawing annotations at the moment are: straight lines, curved lines, rectangles, circles and texts, with a personalization of colors, pen widths, and pen types. Clearly this set of tools can be expanded in future releases of the software. After inserting the visual annotations, they are added to the MX file using the XML elements previously introduced.
Fig. 2. Visual annotations on an MX score. In the main part of the figure, graphical examples are provided to illustrate the capabilities of the software application.

After discussing the editing procedures to create graphical annotations, we now present an example of application that shows analysis in real time, together with the execution of the annotated music piece.

Figure 3 illustrates the interface of the fruition software. The center of the window hosts the graphical score of the music piece, while in the bottom area there is a representation of the corresponding waveform. In the left side of the interface, three examples of visual annotations can be chosen.

The dark rectangle area over the score is the graphical representation of one of the available analyses. During the encoding phase, a macro-segmentation of the piece has been performed, and a number of music objects (in this case, similar phrases) has been described within the Structural layer. When the user enables the visualization of this kind of analysis, in the interface dark rectangles remark the occurrence of a music object.

Thanks to the MX format, this application integrates the possibility to view and play various contents from the same layer or from different layers, all in a synchronized way. For example, it is possible to follow two or more score versions, even simultaneously, while a media file is playing, and the current playing note is highlighted in real-time. Besides, musicological analyses can be enjoyed by taking into account their visual and audio counterparts.
4 BWV 636: a Case Study from J.S. Bach’s Orgelbüchlein

An analysis is composed of many score segments connected by transition maps, which encode their mutual relationships. For instance, a generic segment B might be the transposition of another segment A by one semitone up.

This kind of analysis is extremely useful in order to understand the compositional process of music pieces and its encoding within a language for music description represents a key feature of every software environment for music analysis.

At a first stage, by exploiting the annotating features of our software, musicologists can encode their own analysis, which is stored into MX Structural layer. Then all the analysis can be made available to other scholars or students for musicological investigation or just music fruition.

As an example of analysis, we considered the choral-prelude “Vater unser im Himmelreich” (Our Father who art in Heaven) BWV 636 for organ by Johann Sebastian Bach. This piece has been chosen because of its shortness (just 12 bars long) and because it is characterized by an extremely dense symbolism.

Baroque music in general, and sacred music in particular, is often characterized by the presence of many “rhetorical figures” which relates for example to the text a piece
of music refers to or, more often, to particular moods or “affects” the music wants to induce in the listener’s mind and sensibility.

The Theory of Emotions, as it is sometimes referred to, wants to provide a link between music and the world of affects: for example the feeling of joy is often expressed in music by the so called “circulatio”, namely a sequence of notes rapidly going up and down by close intervals and drawing in the space of pitches a sort of circles.

Rhetorical references can be even more specific, like in this example, and can involve graphical, and thus musical, representation of concrete objects.

![Fig. 4. The scheme of the cross pattern of BWV 636.](image)

Figure 4 reports the abstract structure of a cross-shaped recurring pattern whose concrete instances are spread across alto, tenor and bass parts of BWV 636; the first is in the tenor part of the first measure (Figure 5). This is an evident reference to Christ’s cross [9] and the understanding of this basic musicological fact is of paramount importance in the comprehension of the whole structure of the piece.

![Fig. 5. First cross pattern in the tenor part of BWV 636.](image)

Of course, also the main theme of choral “Vater unser im Himmelreich”, which is present in the soprano part of BWV 636, is the other fundamental component of the piece.

In our implementation all patterns are stored as feature objects together with other feature vectors related to music information retrieval [10] and linked to each music segment that can be displayed during the playback; the user can mark each specific instance of any kind of musicologically relevant pattern (main themes, peculiar patterns, etc.) within the piece and automatically store it in the Structural layer of MX.
The advantage is that those instances can be highlighted during the playback of the piece for didactical and study purposes.

Figure 6 shows a representation of BWV 636 analysis with all cross-patterns highlighted. This kind of representation allows for a comprehensive understanding of the overall structure at a glance.

Fig. 6. Representation of the cross pattern on the manuscript of BWV 636.
5 Conclusions

In this paper, an advanced application to understand music structures and enjoy music at different degrees of abstraction has been illustrated. This tool allows to follow a music piece in its many complementary aspects: symbolic score, graphical notations, audio, and structure.

The key feature is the ease of showing given characteristics of a music piece, highlighting music symbols or more complex aggregations over one of the possible graphical and audio renderings of a given music sheet. Thanks to our proposal, the results of musical analysis are no more written words that must be referred to a symbolic score to be understood, but they can be enjoyed within an integrated framework where the original score, its revised versions, its many audio descriptions, and much more are easily available. Moreover, the deep integration among multimedia objects strengthens the comprehension of music structures.

The MX standard and its applications present no constraint about music genres, cultural areas, historical periods, or different approaches to composition and musical analysis. As a consequence, the methods and the interfaces described so far can be applied to baroque fugue as well as Indian raga, to operatic arias as well as jazz improvisation. Of course, the kind of analysis and the features to be investigated can change noticeably, but the overall approach – consisting in a unique framework for heterogeneous music descriptions – preserves its capability to convey information in an integrated, intuitive and immediate way.

Interesting applications can be implemented, particularly for cultures and kinds of music far from Common Western Notation, such as those studied by ethnomusicology, where there has been little investigation and methodological research.

References