Outline of the MX Standard

Luca A. Ludovico

LIM – DI*Co* - Università degli Studi di Milano
Via Comelico 39/41 – I-20135 Milano, Italia
luca.ludovico@dico.unimi.it

Abstract. MX is a new XML-based format to describe comprehensively heterogeneous music contents. 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.

1 A New Standard for Music

MX, an acronym which stands for *Musical application using XML*, is the code-name for a new file format whose international standardisation is in progress. Its development follows the guidelines of IEEE P1599, *Recommended Practice Dealing with Applications and Representations of Symbolic Music Information Using the XML Language*.

MX is the result of research efforts at the *Laboratorio di Informatica Musicale*, or LIM, of the *Università degli Studi di Milano*. P1599, the IEEE Standard, is sponsored by the Computer Society Standards Activity Board and was launched by the Technical Committee on Computer Generated Music (IEEE CS TC on CGM) [1]. In 2002, a prototypal version of the format was released, originally known as *Musical Application using XML*, or MAX [2]. This format was discussed at IEEE *MAX 2002* international conference. The IEEE final evaluation process, known as balloting, is currently being performed, with the aim of making MX/P1599 an international standard.

The most recent version is *MX Release Candidate 1*, whose DTD and documentation can be downloaded from http://www.mx.dico.unimi.it. Tools for *music visualisation* [3], *content-based retrieval* [4], and *automatic segmentation* [5] are currently available.

2 Key Features of MX

MX is an XML-based format. There are many advantages in choosing XML to describe information in general, and music information in particular. For instance, an

B. Falcidieno et al. (Eds.): SAMT 2007, LNCS 4816, pp. 196–199, 2007.

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. Thanks to the intrinsic capability of XML to provide structures for information, such representations can be organised in an effective and efficient way. MX employs *six different layers* to represent information, as explained in [8] and shown in Figure 1:

- *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. Of course, the higher their number, the richer the musical description.

Richness has been mentioned in regard to the number of heterogeneous types of media description, namely symbolic, logic, audio, graphic, etc. But the philosophy of MX allows one extra step, namely that each layer can contain many digital instances. For example, the *Audio* layer could link to several audio tracks, 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 is possible to adopt some *ad hoc* encoding in addition to already existing formats to represent information. In fact, while a comprehensive format to represent music is not available, popular existing standards must be taken into account. This is a not a contradiction because of the two-sided approach of MX to music representation, which is: keep intrinsic music descriptions inside of the MX file – in XML format – and media objects outside of the MX file – in their original format. The symbols that belong to the score, such as chords and notes, are described in XML, in the *Logic* layer. On the contrary, MP3 files and other audio descriptions are not translated into XML format, rather they are linked and mapped inside the corresponding MX layer, the *Audio* layer.

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, on-stage photographs, and playbills. Thanks to the

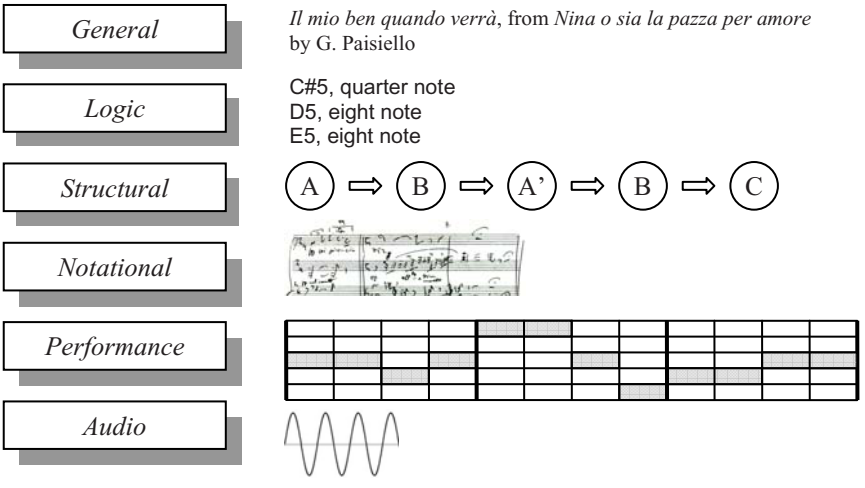

Fig. 1. The characteristic multi-layered structure of MX. In the right part of the figure, intuitive graphical examples are provided to illustrate the purpose of the layers.

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, where the definition and granularity of events can be chosen by the author of the encoding. The spine provides both an abstraction level and the glue among layers, and represents 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. 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 logic definition – e.g. it is a note or a rest – it can be graphically represented in many scores and rendered in a number of audio tracks. These aspects are treated in the *Logic*, *Notational*, and *Audio* 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 audio clips mapped in the *Audio* 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 Conclusions

The format proposed here has been designed to achieve a comprehensive description of music, content interoperability, and deliverability. On one side, this way of encoding music pays special attention to on-line accessibility, digitalisation of analogue material and preservation of artefacts from the past, independently of the cultural origin and language. On the other, the format is meant to provide an integrated and evolved fruition of music, thus representing a new approach to music delivery and enjoyment.

We hope that such an effort will open the way for a large number of new applications with increased power and flexibility, as well as new markets for these kinds of applications. The repercussions may have a wide effect on music education, media entertainment, enjoyment and development of music as a whole.

Specific applications of the MX format to music description and representation will be provided in other articles of this special session.

References

1. Baggi, D.L.: Technical Committee on Computer-Generated Music. Computer 28(11), 91–92 (1995)
2. Haus, G., Longari, M.: MAX 2002. Proceeding of the First International IEEE Conference on Musical Application using XML. IEEE Computer Society, Los Alamitos (2002)
3. Baggi, D.L., Baratè, A., Haus, G., Ludovico, L.A.: A computer tool to enjoy and understand music. In: Proceedings of EWIMT 2005 – Integration of Knowledge, Semantics and Digital Media Technology, pp. 213–217 (2005)
4. Baratè, A., Haus, G., Ludovico, L.A.: An XML-Based Format for Advanced Music Fruition. In: Proceedings of SMC 2006 – Sound and Music Computing 2006 (2006)
5. Haus, G., Ludovico, L.A.: Music Segmentation: An XML-oriented Approach. In: Wiil, U.K. (ed.) CMMR 2004. LNCS, vol. 3310, pp. 330–346. Springer, Heidelberg (2005)
6. Roland, P.: The Music Encoding Initiative (MEI). In: Proceedings of the first IEEE International Conference MAX 2002 – Musical Application using XML, pp. 55–59 (2002)
7. Steyn, J.: Framework for a music markup language. In: MAX 2002. Proceeding of the First International IEEE Conference on Musical Application using XML, pp. 22–29. IEEE Computer Society, Los Alamitos (2002)
8. Haus, G., Longari, M.: A Multi-Layered, Time-Based Music Description Approach Based on XML. Computer Music Journal 29(1), 70–85 (2005)