The MX Formalism for Semantic Web Compatible Representation of Music Metadata*

S. Castano, A. Ferrara, G. Haus, L.A. Ludovico, S. Montanelli, G. Racca, and G. Vercellesi

Università degli Studi di Milano,
DICo - Via Comelico, 39, 20135 Milano - Italy
{castano, ferrara, haus, ludovico, montanelli, racca, vercellesi}@dico.unimi.it

Abstract. Music description is nowadays considered an important matter in Information and Communication Technology. The encoding formats commonly accepted and employed are often characterized by a partial view of the whole problem: they describe music data or metadata for score, audio tracks, computer performances of music pieces, but they seldom encode all these aspects together. In this paper, we present the MX formalism that aims to address this limitation of the existing formats, by providing a Semantic Web compatible representation of music information in terms of structural and semantic features, by means of XML and OWL.

1 Introduction

Music description is nowadays considered an important matter in Information and Communication Technology. The encoding formats commonly accepted and employed are often characterized by a partial view of the whole problem: 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 have been proposed aimed at a precise characterization of different specific music aspects. For example, MP3, AAC and PCM formats encode audio recordings; MIDI represents a well known standard for computer-driven performance; TIFF and JPEG files can contain the results of a scanning process of scores; finally, NIFF and Sibelius formats are aimed at score typing and publishing. The purpose of these file formats is the description of music data and, in a certain measure, also of metadata. Let us consider for example an MP3 file. It describes both music contents (e.g., frequencies, loudness, and other audio characteristics) and the corresponding metadata (e.g., author, song title, album). In general terms, common file formats usually include not only a detailed description of one musical aspect - logical, structural, aural, interpretative, typographic,

* This paper has been partially funded by "Wide-scalE, Broadband, MIddleware for Network Distributed Services (WEB-MINDS)" FIRB Project funded by the Italian Ministry of Education, University, and Research.

K.S. Candan and A. Celentano (Eds.): MIS 2005, LNCS 3665, pp. 78–92, 2005.

or graphical - but also more abstract information about the piece. In order to appreciate the richness of music communication, we can point out that music - in its most general meaning - can stimulate different senses: the sense of hearing, the sense of sight and even the sense of touch. Music language is made up of many different and complementary aspects. A complete analysis of music richness and complexity is provided in [1], where six different levels of description are identified - namely general, structural, logical, notational, audio and performance layers. This multi-layer structure could answer the request of completeness in music description, as the layers we listed can be considered a good coverage of the different domains of music. They take into account the evidence that music is the composition itself as well as the sound a listener hears, and music is the score that a performer reads as well as the execution provided by a computer system. In this paper, we describe the MX Format that provide a XML-based support for the representation of the different layers composing music information. In particular, we focus on the top layer, devoted to the representation of music metadata and their semantics. We propose the use of ontologies in order to address the requirement of a flexible and meaningful description of music metadata, and of genre in particular.

The paper is organized as follows. In Section 2 we provide a short description of the MX format, and we point out the main limitations of the actual metadata description and the requirements for a richer representation of the metadata layer. In Section 3, we describe our approach to the representation of the music metadata by means of an ontology that describes the MX Semantic Layer. In Section 4, we point out the applicability issues related our work, and, in Section 5, we discuss the related work with respect to our approach. Finally, in Section 6, we give conclusions and we focus on the future work.

2 A Comprehensive XML-Based Format for Music

We have developed a new XML-based format, called MX. Currently, MX is undergoing the IEEE standardization process, as described in [2]. Our approach is different from the aforementioned partial perspectives in music description, in particular because we represent music information according to a multi-layer structure and to the concept of space-time construct. In our opinion, music information can be (and should be) structured by using a layer subdivision model, as shown in Figure 1. Each layer is specific to a different degree of abstraction in music information. In our proposal for a common and exhaustive format, we distinguish among General, Structural, Music Logic, Notational, Performance and Audio layers (see Figure 1a). For example, MusicXML could be integrated in the more comprehensive MX format to implement the Logical Organized Symbols layer, that is score symbolic information (e.g., notes, rests), whereas other common file types can be linked to represent other layers: TIFF for the Notational layer, MP3 and WAV for the Audio layer and so on. The issue of the integration between MX and other formats is covered in [3].

The main advantage of MX is the richness of our descriptive format, which is based on other commonly accepted encodings aimed at more specific descriptions.

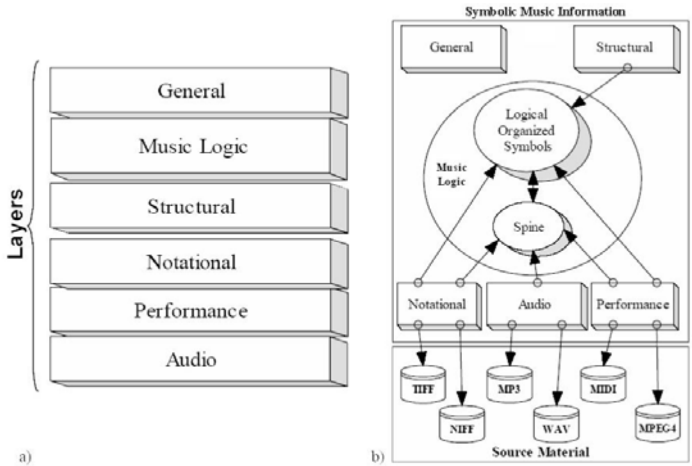

Fig. 1. (a) Music information layers and (b) relations among them

Considering music structure as a multi-layered information, we need a sort of glue to keep together the heterogeneous contributions that compose it. Accordingly, we introduced the concept of *spine*. Spine is a structure that relates time and spatial information (see Figure 2), where measurement units are expressed in relative format. Through such a mapping, it is possible to fix a point in a layer instance (e.g. Notational) and investigate the corresponding point in another one (e.g. Performance or Audio).

2.1 A MX Encoding Example

In the following, we provide a series of significant portions of MX encoding the different layers, together with some comments for demonstration purpose. The complete DTD of MX 1.5 format is available at http://www.computer.org/-standards/1599/par.htm

The Logic layer contains information referenced by all other layers, and 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 second example below illustrates how notes, chords and rests can be represented in MX notation.

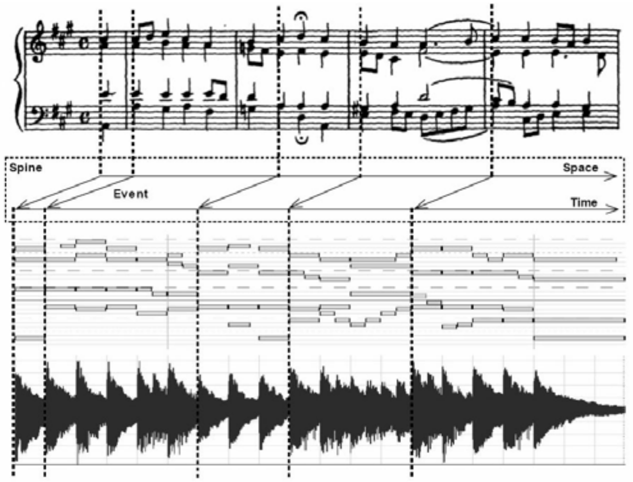

Fig. 2. Spine: relationships between Notational, Performance and Audio layer

```
<spine>
       <event id="e0" timing="0" hpos="NULL"/>
       ...
</spine>

<measure number="1">
      <voice ref="violin_1">
            <rest event_ref="v1_e0" staff_ref="staff_1">
                  <duration den="1" num="2"/>
            </rest>
            <chord event_ref="v1_e1">
                  <notehead staff_ref="staff_1">
                        <pitch step="C" octave="6"/>
                        <duration num="1" den="2"/>
                  </notehead>
            </chord>
       </voice>
</measure>
```

The Structural layer contains explicit descriptions of music objects together with their causal relationships, from both the compositional and musicological point of view, i.e. 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 preformance. This layer supports symbolic formats such as MIDI, Csound or SASL/SAOL files. Finally, the Audio layer describes properties of the source material containing music audio information. It is the lowest level of the layered structure.

```
<audio>
      <clip>
            <clip_format file_name="Beethoven_Symphony#7_4.mp3" file_format="MP3"
            encoding_format="MPEG1">
                  <frequency type="constant" avg_value_Hz="44100"/>
                  <bitrate type="CBR" avg_value_bitsec=""/>
                  <channel channel_number="2" LFE="no"/>
            </clip_format>
            <clip_indexing>
                  <clip_marker timing_type="sec">
                        <clip_marker_event timing="0.00" spine_ref="v1_e0"/>
                        ...
            </clip_indexing>
      </clip>
</audio>
```

2.2 The MX General Layer and Its Limitations

Recalling the previous subsection, MX format is rich in information and description possibilities. One of its layers, the so-called General layer, is not directly related to score and audio contents. This layer simply describes some fundamental alphanumeric information about the coded music work. The situation is shown in Figure 1, where the General block is clearly separated from other levels, even if it belongs to Symbolic Music Information as well. A particular importance is given to sub-element Description, devoted to author, genre and piece information. Such music metadata, even if not related to music symbols or audio performances, are particularly important for music classification and retrieval. In detail, the MX 1.5 DTD description of metadata can be shown by means of the following example:

```
<general>
      <description>
            <work_title>Symphony No.7 in A major</work_title>
            <work_number>7</work_number>
            <movement_title>Allegro con brio</movement_title>
            <movement_number>4</movement_number>
            <genre>
                  <genre_spec name="Classic" weight="40"/>
                 <genre_spec name="Preromantic" weight="60"/>
            </genre>
            <author type="composer">Ludwig van Beethoven</author>
      </description>
</general>
```

In particular the Description element contains 6 sub-elements, corresponding to 6 metadata categories: This structure is aimed at the general description of the original score, as the author(s) conceived it. In the Audio layer, for instance, we will find similar information (and a similar structure) about the audioclips related to the score itself. Some elements would be redundant, so they are not present in the audio instances: typically, the data about the composition itself (e.g., composer, title). Other elements are naturally related only to performances-it is the case of the performers list. Finally, some elements are repeated as their value can differ from the original score: potentially we can conceive a jazz cover version of a classical piece or a rock performance of a baroque music work. Apparently, this schema seems to be complete and powerful. The characteristic information about a single score and its performances can be represented, and all these data are organized in a multi-layered structure. However, even in the General layer, a number of problematic issues arise and pose a number of requirements to be addressed. First, we have general metadata to describe the score and partially different metadata to describe related score performances. Second, the schema and the names we gave to elements are meaningful only for a certain kind of music: if the concepts of movement and work can be useful for a traditional composer's production, it would be difficult to classify a pop song according to this terminology. Besides, the concept of genre is vague. Genre could represent something related to a historical period, to a geographical area, to a musical style, to a musicological classification, and, more generally, different people give usually different meanings to the concept of genre and refer to different genre descriptions. As a consequence, the final key problem is represented by the wide possibilities of classification left to the user who encodes a music piece by MX. For example, a description of the genre in natural language could provide informative richness, but on the other hand it could have dramatic repercussions on music information retrieval. Imagine a query to extract all the baroque pieces from a huge database where each piece was described in natural language and using no conventions about terminology and classification features.

2.3 Ontologies for Classification and Management of Music Metadata

There is no consensus about *what* belongs to *which* genre, and about the taxonomy itself. For example, two users may agree that certain values associated to certain characteristics define a single genre, but they could give two different names to that genre (e.g. Easy Listening music is also called Variety). It is possible to define a piece of music through several style levels. For example, II sang Dixie by Dwight Yoakam is defined as Popular music, or Country music, or Hard Honky Tonk as well, where each genre is a subset of the previous one. Moreover, a piece of music could change the associated genre, and a definition of a genre could change in time (e.g. Underground was a kind of independent music, now the same term defines a kind of disco music). Unfortunately a stable and commonly accepted definition for every genre does not exist. In other terms, the concept of genre is non-well-defined, and in many cases ambiguous. The problems related to the classification and the management of music metadata in a MIR context are well covered in [4]. This is also the result of a study (see [5]) that compared three Internet genre taxonomies - allmusic.com, amazon.com and mp3.com - coming to an important result: there is no consensus in the name used in these classifications, and no shared structure. However, from our point of view, the genre is particularly important, as this information could help music consumers and MIR systems to identify, classify and retrieve music pieces, using genre as a criterion of query or as a base for similarity measures. Our goal is to provide a tool for classifying and describe genre, and music metadata in general, that: i) does not impose a shared and common definition; ii) provides a flexible mechanism that, at the same time, supports different users to give their own description of music; iii) supports a discovery service and provides matching functionalities for comparing different descriptions in order to find similarities among them. These goals can be achieved by using an ontology that describes music metadata.

3 Ontology-Based Representation of the MX Semantic Layer

Ontology is generally defined as a explicit representation of a conceptualization [6]. In our approach, a *Music Ontology* is used for enriching the MX Format by providing a semantic description, called Semantic Layer, of the layer General of MX.

3.1 Conceptual Specification of the MX Semantic Layer

The music ontology that we create to express the semantic information within the MX general layer is specified in OWL [7]. In this paper, in order to give a more comfortable representation and a clearer view of objects involved as well as their relations, we decided to use ER formalism for describing the ontology conceptualization of the semantic layer, and then to describe the OWL implementation of this conceptualization. The ER diagram of our ontology is shown in Figure 3. The ontology has to satisfy three main requirements: i) to separate informa-

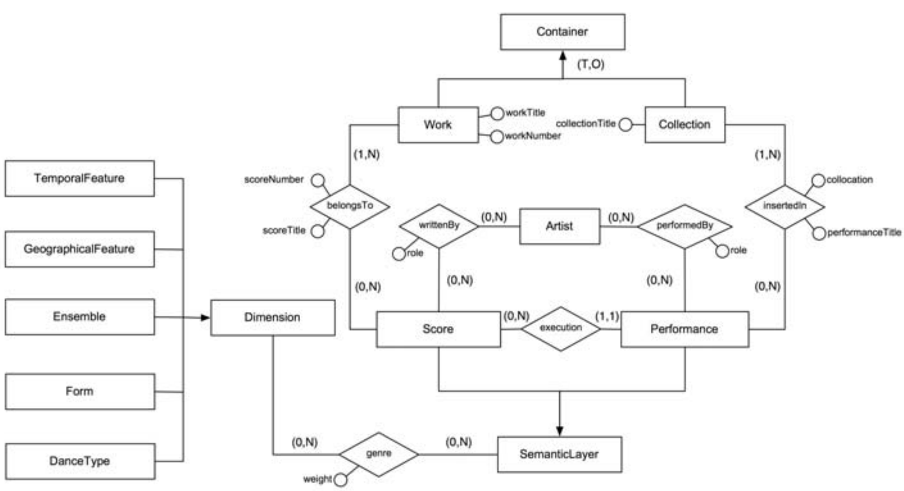

Fig. 3. ER diagram for the music ontology

tion regarding scores and performances; ii) to adequately express the complex relationships among artists and scores/performances; iii) to adequately model the genre classification. The first requirement refers to the need to express the conceptual separation between a score and its (possibly) multiple performances, as well as the logical relationship between a score and all its performances. This distinction is one of the main features of the MX Format. The second requirement refers to the complex relationships between a score and its authors as well as between a performance and the artists involved. A score may have many authors with different roles (e.g. music author, lyrics author), an artist may be the author of a particular score, but we may not have any performance of that score involving its author. Moreover the same artist may be involved in a particular performance with different roles (e.g. vocals and guitar). The third requirement refers to the classification of a performance or score with respect to its genre. In order to deal with this complexity we can think to the genre as a classification along different dimensions. Each dimension refers to a particular set of features and each score or performance can be classified along one or more of these dimensions. Moreover, when we associate a particular score or performance with a particular dimension, we want to specify the strength of the relationship. In the ER diagram shown in Figure 3 we see the entities of our representation as well as their main attributes. Some attributes are not shown to mantain the figure cleaner (e.g. first and last name of artists, creation date of a score). Semantic information contained into the MX general layer is enriched by associating it to an instance of the SemanticLayer entity. This entity is a total and exclusive generalization of semantic information related to scores (the Score entity) and performances (the Performance entity). Each score may be related to its performances. Each score may be related to its authors that are instances of the Artist entity. While associating an artist with a score we must specify a role, that represents the particular kind of authorship (e.g. Composer, Librettist). In the same way, we associate an artist to a particular performance (e.g. Conductor, Orchestra). We also decided to introduce a generic container that would allow to group together scores or performances. This container will be useful for retrieval or classification purposes. We can group scores within works and performances within collections. Since the hierarchy is total but overlapped, works and collections are not disjoint. A Work refers to way the author of the score originally released it, while a Collection refers to all the containers in which a particular performance can be found. A Bach fugue will be contained in an instance of the Work entity, typically referring to its paper-publication, while "Helter Skelter" will be contained in an instance of the Work entity representing "The White Album". Beside, we can have a particular recording of "Helter Skelter" song that can be contained in the instance of the Collection entity referring to "The White Album" as well as in the instance referring to the "Beatles' complete work". In order to deal with the complexity of genre classification, we decided to model the genre as an association between a score or performance and a dimension. This association must be enriched by a value (weight) that specify the strenght of the relation. The Dimension entity is a generalization, since we have specific entities expressing the five dimensions that we decided to include in this first implementation of our ontology. Each dimension refers to a particular feature of the score or performance. The five entites representing genre dimensions are: i) TemporalFeature: this entity describes historical periods (e.g. Pre-Romantic, 80's); ii) GeographicalFeatures: this entity describes countries and regions; iii) Ensamble: this entity elements involved in a particular performance (e.g. String Quartet); iv) Form: this entity describes relations among elements (e.g. melodic themes) of a score. A form instance can be a Fugue or a Sonata; v) DanceTypeFeature this entity describes rhythmic reatures about accents disposition (e.g. Waltzer, Polka). A particular instance of the Score or Performance entity will be associated to one or more instances of the Dimension hierarchy. For each genre association we will have also to express a weight. We can classify our scores and performances focusing our attention to one or more dimension. Moreover, this kind of classification allows us to associate the same score instance to multiple instances of the same dimension, overcoming the limitations of tipical classification methods based on predefined genre attribution.

3.2 OWL Representation of the MX Semantic Layer

To ensure Semantic Web compatibility, the music ontology is represented by means of OWL. The idea is to define an OWL ontology describing the conceptualization of the semantic layer in terms of concepts, properties and semantic relations. In terms of OWL, the music ontology is associated with the MX semantic layer by means of a XML Namespace, denoted as mxsl and available at http://islab.dico.unimi.it/ontologies/mx-semantic-layer.owl. A graphical representation of this approach is shown in Figure 4.The music ontology includes the OWL representation of the concepts described in Figure 3. In the ontology we have defined an OWL class for each entity, and appropriate OWL properties for representing the attributes and the relations reported in the ER model. The hierarchies have been represented by means of the OWL subClassOf construct, specifying an explicit disjunction between the Score and the Performance elements. The problem of representing attributes of the relations (e.g., the attribute weight of the relation genre) in OWL has been addressed as suggested in [8]. The problem is due to the fact that OWL provides constructs for representing only binary relations, without attributes. In order to address this limitation, we have defined a class for each attribute featured relation and a set of properties for representing the relation attribute as well as its second argument. An example of this approach is provided in Figure 5 where we show the OWL description of the class Genre that represents the genre relation of the ER model. The attribute weight and the argument Dimension of the genre relation are represented by means of the restriction mechanism of OWL, which states that the values of the dimension for a genre must be an instance of the class Dimension and that each instance of the class Genre has exactly one dimension associated with one weight. The result is that each genre featuring a semantic layer description is seen as a pair of the form $< Dimension, Weight >$. Each genre feature is then associated with a semantic layer description by means of a property genre, as specified in the class

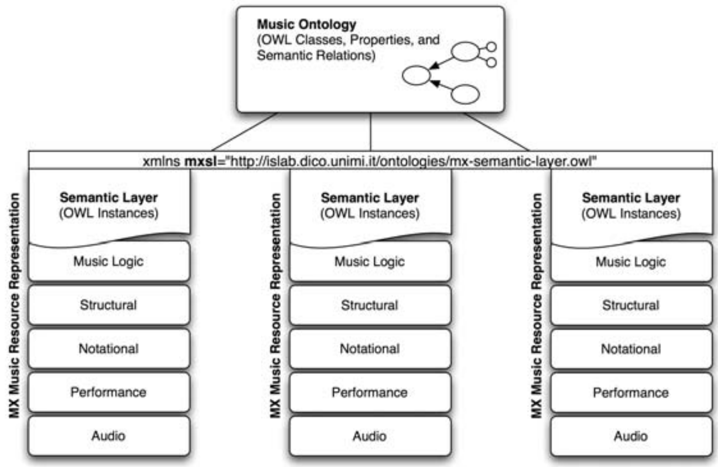

Fig. 4. The music ontology and the MX Semantic Layer

SemanticLayer by an appropriate OWL restriction. A complete description of the music ontology is shown in Figure 6, where the OWL features are represented graphically by means of the HOE (H-Model Ontology Editor) developed for ontology management in the context of the Helios project [9]. In the logical view provided by the HOE tool, the concepts are represented by rectangular labels, while properties are represented by circles. Solid lines and dashed lines represent the domain and range of the properties, respectively, while arrows represent subClassOf relations. By referring to the music ontology, each MX semantic

```
<owl:Class rdf:about="#Genre">
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty rdf:resource="#dimension" />
       <owl:allValuesFrom rdf:resource="#Dimension"/>
       </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty rdf:resource="#dimension" />
       <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
     </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty rdf:resource="#weight" />
       <owl:cardinality rdf:datatype="&xsd;int">1</owl:cardinality>
     </owl:Restriction>
   </rdfs:subClassOf>
</owl:Class>
```

Fig. 5. Example of the OWL description of the genre concept

description is an OWL document that imports the ontology and defines a set of instances for describing a music resource represented with MX. The OWL document starts with a declaration of the mxsl namespace that is used for linking the semantic description to the music ontology, and with the owl:Ontology element that imports the music ontology elements. Each score or performance modeled by MX is described by means of an instance of the class Score or Performance, respectively. The instances describe a MX resource in terms of its properties that are associated with other instances describing information about the artists and the role that they have played in the present MX resource, the containers the resource belongs to, the genre classification, and, in the case of scores, the performances associated with the present resource. As an example, we consider the 4th movement of the Symphony No.7 in A major by Ludwig van Beethoven. The score title depends on the title that this resource have when associated with a work as well as its number in that work. These dependences are represented by creating an instance of the class BelongsTo, that describes the work in which the score is inserted, and the title and number of the score within that context. The main advantage of this approach is that we can associate with a score a multiple set of containers, in the case that the author have inserted the score in more than a work. Moreover the title and number of the score can be different for each work it belongs to. A similar approach has been used for artists. In fact, each score is associated with a number of artists with different roles (e.g., author, composer). A possible solution is to associate with a score a predefined set of roles. The disadvantage is that the score statement is not flexible, in that we cannot add a new role to it. Our solution is based on the idea to define an instance for each pair composed by an artist and its role. In that way, we have two advantages: i) we can create such an information for each kind of role, without limitations, and ii) we create a portion of information that can be reused in further score descriptions. The score genre is represented by instances of the class Genre, each representing the weight set for a dimension. In the example, we state that the score is classic and preromantic. Moreover, we state that the preromantic feature is more relevant than the classic one for this score, by means of a weight associated with the two dimensions. Once these genres are set, we can reuse them for further classifications. An example of genre definition and its use in a score description is shown in Figure 7. Finally, a score is associated with one or more performances that are described by means of the same properties used for the score. The complete description of the example is available at http://islab.dico.unimi.it/ontologies/mxsl-description-example.owl. The approach used for the MX resource description has two main advantages: i) we have a high level of flexibility in describing the features of music information in that there is an extensible mechanism for enriching the set of available instances and the capability of capturing the fact that different people describe music in a high number of different ways; ii) while each description is composed, a new set of reusable instances is defined. In this way, each new description enriches the ontology by adding new features that can be used for further descriptions.

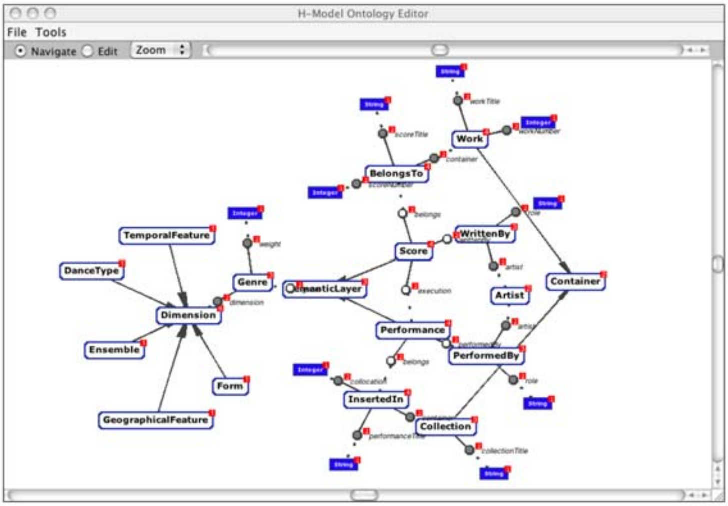

Fig. 6. Graphical representation of the MX semantic layer ontology

```
<mxsl:Genre rdf:ID="Preromanticism_60">
     <mxsl:weight rdf:datatype="&xsd;int">60</mxsl:weight>
    <mxsl:dimension rdf:resource="#Preromanticism"/>
</mxsl:Genre>
<mxsl:Genre rdf:ID="Classicism_40">
    <mxsl:weight rdf:datatype="&xsd;int">40</mxsl:weight>
    <mxsl:dimension rdf:resource="#Classicism"/>
</mxsl:Genre>
<mxsl:Score rdf:ID="Symphony_N_7_A_major_Allegro_con_brio">
     ...
     <mxsl:genre rdf:resource="#Classicism_40"/>
     <mxsl:genre rdf:resource="#Preromanticism_60"/>
     ...
</mxsl:Score>
```

Fig. 7. An example of genre definition

4 Applicability Issues

In this section, we discuss main applicability issues of using the proposed ontology for semi-automated music resources retrieval and classification according to semantic criteria.

4.1 Semantic Retrieval of Music Resources

The MX semantic layer can be used to improve music pieces retrieval from the traditional keyword-based approach to a semantics-based approach. According to the keyword-based approach, a query can contain one or more equality constraints represented by pairs $< metadata\_name = value >$ to be satisfied. Each music piece whose metadata fulfills all the requirements is considered to be relevant for the query. This approach is suitable for searches related to unambiguous metadata, such as Title and Author. Nevertheless, the semantics-based approach allows to take into consideration the different representation of subjective metadata, such as Genre, during the retrieval process. Given a query, a music piece is found to be relevant if the values of its metadata match the query requirements according to some similarity criteria. The meaning of similarity can be user-defined and depends on the context of the search. In general, the semantic retrieval of music can be improved by using ontology-based matching techniques for evaluating the similarity between two different music piece description. An example of this kind of similarity evaluation is provided in [10,11].

4.2 Semi-automated Classification of Music Resources

Some features about genres can be automatically extracted from score features. Melodic profiles, persistent rhythms, voices texture, declared ensemble are only some examples of music characteristics that can be extracted from MX format. Sometimes, this information provides a direct mean of classification: consider a row (or series) identified by a trivial melody analysis, which is an unambiguous proof for serial music. However, in general terms the extractable information is not intended for an automatic classification, rather it is helpful to exclude several possibilities. For instance, a simple harmony analysis could roughly date the composition: some harmonic behaviors are typical of jazz music and can not be applied to baroque pieces. Similarly, it is difficult to classify a piece as a string quartet if the ensemble includes more than four instruments, or those instruments differ from violin, viola, and cello. Our idea is to exploit the rich description provided by MX in describing the structural features of music information, in order to semi-automatically support the classification of music with respect to its semantic description. The application of such a technique is particularly relevant for the publication of catalogues of music that can potentially contain a high number of music items.

5 Related Work

XML is an effective way to describe music information. Nowadays, there is a number of good dialects to encode music by means of XML, such as MusicXML, MusiXML, MusiCat, MEI, MDL (see [3] for a thorough discussion). In particular, we have at least two good reasons to mention MusicXML [12]. MusicXML is a comprehensive way to represent symbolic information. As a consequence, MusicXML was integrated in a number of commercial programs. Among them, its worthwhile to cite one of the leading applications for music notation: Coda Music Finale. One of the key advantages of MusicXML over other XML-based formats is represented by its popularity in the field of music software. However, all the encoding formats we listed before are not interested in semantic descriptions of metadata, and do not contain an ontology approach similar to our. In MPEG-7 context, currently there are initiatives to integrate OWL ontologies in a framework opportunely developed for the support of ontology-based semantic indexing and retrieval of audiovisual content. This initiative follows the Semantic Level of MPEG-7 MDS (*Multimedia Description Schemas*), and TV-Anytime standard specifications for metadata descriptions. Despite of MX Semantic Layer, MPEG-7 Semantic Level describes music information from the real world perspective, giving the emphasis on Events, Objects, Concepts, Places, Time in narrative worlds and Abstraction. Therefore, MPEG-7 ontology is only aimed at the description of music performance and not of score information, as in the case of MX. The methodology of OWL ontology integration and the interoperability methodology and tools have been based on a core OWL ontology, which fully covers the Semantic Part of the MPEG-7 MDS and a methodology for the definition of domain-specific ontologies that extend the core ontology in order to fully describe the concepts of specific application domains, together with two sets of rules, used for the transformation of semantic metadata (formed according to the core ontology and its domain-specific extensions) to MPEG-7 and TV-Anytime compliant XML documents respectively. A complete discussion of the matter is in [13]. Genre, an intrinsic property of music, is probably one of the most important descriptor used to classify music archives. Traditionally, genre classification has been performed manually but many automatical approaches are provided by the state of art. In [14], three different categories of genre classification are proposed: i) manual approch based on human knowledgment and culture; ii) automatic approch based on automatic extraction of audio features; iii) automatic approch based on objective simliraty measures. Taxonomy use is the main difference beetwen the two automatic approaches: in the first a given taxonomy is necessary, in the second is not required. Further methods to classifiy music genre working on audio signals can be found in [15]. In MX, we have tried to classify genres by an OWL ontology, in order to get a taxonomy as flexible as possible and capturing the complexity of real world genre classifications.

6 Concluding Remarks

In this paper, we have presented the MX format for music representation, together with a proposal of enrichment of MX to achieve a flexible and Semantic Web compatible representation of the metadata associated with MX resources. The metadata representation is realized by means of an OWL ontology that describes music information and proposes three main direction for future work. A first activity is the enrichment of the ontology with new classes and properties that capture further features of music information. A particular attention will be devoted to the structure of music and its relations with the problem of genre classification. A second activity will be devoted to the definition of the functionalities of a music retrieval system based on the ontological description of scores. The idea is to adapt ontology matching techniques to capture the similarity among different music resources and to exploit it for retrieval purposes [11]. Finally, a third activity will be devoted to semi-automated, ontology-driven, classification of music resources, with respect to their genre. The idea is to exploit the rich description provided by MX in describing the structural feature of music in order to automatically identify in the music ontology the genre of a given music item and classify it accordingly.

References

1. Haus, G., Longari, M.: Towards a Symbolic/Time-Based Music Language Based on XML. In: Proc. of Musical Application Using XML (MAX 2002), Milan, Italy, IEEE (2002)
2. XML Musical Application Working Group: Recommended Practice for the Definition of a Commonly Acceptable Musical Application Using the XML Language. IEEE SA 1599, PAR approval date 09/27/2001 (2001)
3. Longari, M.: Formal and Software Tools for a Commonly Acceptable Musical Application Using the XML Language. PhD thesis, Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano (2003)
4. Diana, L.: An XML-based Querying Model for MIR Applications Within a Multi-layered Music Information Environment. PhD thesis, Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano (2004)
5. Pachet, F., Cazaly, D.: A taxonomy of musical genres. In: Proc. of Content-Based Multimedia Information Access (RIAO), Paris, France (2000)
6. Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition (2003) 199–220
7. Smith, M.K., Welty, C., McGuinness, D.L.: OWL Web Ontology Language Guide. Technical report, W3C Recommendation (10 February 2004)
8. Noy, N., Rector, A.: Defining n-ary relations on the semantic web: Use with individuals. Technical report, W3C Working Draft (21 July 2004)
9. ISLab Università degli Studi di Milano: The HELIOS Project. http://islab.dico.unimi.it/helios/ (2003)
10. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: Matching Techniques for Resource Discovery in Distributed Systems Using Heterogeneous Ontology Descriptions. In: Proc. of the Int. Conference on Coding and Computing (ITCC 2004), Las Vegas, Nevada, USA, IEEE Computer Society (2004)
11. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: From Surface to Intensive Matching of Semantic Web Ontologies. In: DEXA Workshops 2004, IEEE (2004) 140–144
12. Good, M.: MusicXML for Notation and Analysis. In: The VirtualScore: Representation, Retrieval, Restoration. MIT Press (2001) 113–124
13. Tsinaraki, C., Panagiotis, P., Christodoulakis, S.: Integration of owl ontologies in mpeg-7 and tvanytime compliant semantic indexing. In: Proc. of the 16th International Conference on Advanced Information Systems Engineering (CAiSE), Riga, Latvia (2004)
14. Aucouturier, J.J., Pachet., F.: Respresenting musial genre: A state of the art. Journal of New Music Research **32** (2003) 83–93
15. Tzanetakis, G., Essl, G., Cook, P.: Automatic musical genre classification of audio signals. In: Proc. of 2nd Annual International Symposium on Music Information Retrieval, Indiana University Bloomington, Indiana, USA (2001)