

# A Semantic Web Ontology for Context-based Classification and Retrieval of Music Resources

ALFIO FERRARA, LUCA A. LUDOVICO, STEFANO MONTANELLI,  
SILVANA CASTANO, and GOFFREDO HAUS  
DICO - Università degli Studi di Milano

---

## 1. INTRODUCTION

Music resource representation is nowadays considered an important matter in Information and Communication Technology. In this context, research activity is devoted to the development of advanced formalisms for a comprehensive representation of music resource features, to overcome limitations of current encoding schemes like MP3, JPEG or AAC that are mainly focused on representing singular specific aspects of music resources. A complete solution for music representation is the MX formalism, which provides a structured XML-based representation of music features. A step ahead in this context is to exploit descriptions of music metadata and their semantics to enable automated classification and retrieval of music resources. An open problem regards the classification of music resources with respect to the notion of musical genre. The difficulties arise from the fact that there is no consensus about *what* belongs to *which* genre, and about the genre taxonomy itself. Moreover, a piece of music could change the associated genre, and the definition of a given genre could change in time (e.g., Underground was a kind of independent music, now the same term defines a kind of disco music).

In this paper, starting from the complete MX description of music, we propose a multi-dimensional description of a music resource in a semantic way, on the basis of the notion of *music context* and *musical genre*. This goal are achieved by defining an ontology that describes music metadata. Ontology is generally defined as an “explicit representation of a conceptualization” [Gruber 1993]. In our approach, an ontology is used for enriching the MX formalism by providing a semantic description of the music resource context and genre. The ontology has to satisfy three main requirements: i) to separate information regarding the context and the genre classification; ii) to adequately express the complex relationships among music fea-

---

Author’s address: A. Ferrara, L.A. Ludovico, S. Montanelli, S. Castano, G. Haus DICO, Università degli Studi di Milano, Via Comelico 39, 20135 Milano, Italy. {ferrara, ludovico, montanelli, castano, haus}@dico.unimi.it.

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.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2006 ACM 1529-3785/2006/0700-0001 \$5.00

tures; iii) to adequately model the genre classification. Furthermore, the ontological representation of music is required in order to support a semantic retrieval of music resources. The key idea is to have methods and techniques capable of exploiting the ontology for comparing different classifications of the music resources with their contexts, by evaluating the similarity among them. Main contributions of the work with respect to existing classification approaches regard: i) the availability of a Semantic Web-compatible definition of music genre according to a comprehensive analysis of some key aspects of music resources (i.e., ensemble, rhythm, harmony, and melody); ii) the definition of a flexible mechanism to support music genre classification in order to allow different interpretations of the same music resource by contemporary providing a reference taxonomy of music genres; iii) the use of the classification information for context-driven and proximity-based search of music resources based on similarities among their descriptions.

The paper is organized as follows: in Section 2, we present the MX formalism for music representation and we discuss the process of extracting context information from music score. In Section 3, we describe the MX-Onto ontology, a two layer ontology architecture for context-based representation of music resources, whose population is based on a score analysis process. In Section 4, we describe the use of ontology knowledge for the proximity-driven discovery of music resources. In Section 5, we present the related work together with a discussion of the original contribution of our proposal. Finally, in Section 6, we give our concluding remarks and we outline our future work.

## 2. MUSIC RESOURCES REPRESENTATION AND ANALYSIS

In this section, we discuss the problem of defining the context of a music resource in order to describe its characteristic features. In particular, we present the MX formalism as a XML-based standard for the representation of the music pieces and of their score in particular and we discuss how music context information can be extracted by means of a process of score analysis.

### 2.1 MX formalism for music representation

In order to represent all the aspects of music in a computer system, we propose a XML-based format, called MX, that is currently undergoing the IEEE standardization process, as described in [Haus 2001]. In MX, we represent music information according to a multi-layer structure and to the concept of space-time construct. The first feature of MX format is its multi-layer structure. Each layer is specific to a different degree of abstraction in music information and, in particular, we distinguish the Structural, Music Logic, Notational, Performance and Audio layers. The *Structural* layer contains explicit descriptions of music objects together with their causal relationships, from both the compositional and the musicological point of view, that is, how music objects can be described as a transformation of previously described music objects. The *Logic* layer contains information referenced by all other layers. It represents what the composer intended to put in the piece and describes the score from a symbolic point of view (e.g., chords, rest). 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



Fig. 1. Spine: relationships between Notational, Performance and Audio layer

images representing the score. 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 files. Finally, the *Audio* layer describes properties of the source material containing music audio information. This multi-layered description allows MX to import a number of different formats aimed at music encoding. For example, MusicXML could be integrated in the more comprehensive MX format to describe score symbolic information (e.g., notes and rests), whereas other common file types such as TIFF for notational, MP3 and WAV for audio can be linked to represent other facets. 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 sort of glue to keep together the heterogeneous contributions 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). The complete 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.

## 2.2 Music context and score analysis

Starting from the MX format, a first step for a semantic representation of music information is to capture the relations that hold among the different features of a music resource. The description of these features is captured by means of the notion of *music resource context*:

*Definition 2.1. Music resource context.* Given a music resource  $r$ , the context  $Ctx(r)$  of  $r$  is a 4-tuple of the form  $\langle \mathcal{E}, \mathcal{R}, \mathcal{H}, \mathcal{M} \rangle$ , where  $\mathcal{E}$  denotes the *Ensemble* that is associated with  $r$ ,  $\mathcal{R}$  denotes the *Rhythm*, that is the rhythmic features of  $r$ ,  $\mathcal{H}$  denotes the *Harmony*, that is the harmonic features of  $r$ , and  $\mathcal{M}$  denotes the *Melody*, that is the melodic features of  $r$ , respectively.

The context of a music resource is derived from the analysis of the music resource score. Score analysis can be conducted over a number of different dimensions. For example, melody, rhythm, and harmony are three basic aspects which can be investigated. Interesting results can come also from more complex analytical activities, such as segmentation or orchestration analysis. In our approach, we fixed the analytical context through the four dimensions of the music resource context, namely melody, rhythm, harmony, and ensemble. The main advantages of such choice are simplicity and immediacy in capturing music characteristics. Besides, rhythm, melody, harmony and orchestration are recognized as key aspects not only in western music, but also in other cultural contexts, such as in the case of the peculiar rhythmic patterns in African dances or the micro-tonality typical of Indonesian gamelan music and Indian classical music. In this sense, choosing those music surfaces represents a way to take into account cultural variety and to catch similarities and differences. All the information we need to perform the aforementioned analyses is certainly present in any MX file: melody and rhythm are encoded in a plain way, harmony can be reconstructed by a trivial verticalization process, and also the list of instruments is provided. However, our context definition is not aimed at replicating all the data already present in the original score; rather, the context is constituted by the aforementioned dimensions at a higher degree of abstraction. Thus, in our approach melody, rhythm, harmony and ensemble are defined in a way that is different from the usual meaning of these terms. For our purposes, melody is no more an ordered sequence of pitches, as well as rhythm is no more an ordered sequence of rhythmic figures; and harmonic dimension is still related to contemporaneous sounds, but it is not directly defined by sets of notes, as well as ensemble is derived from the list of parts and voices, but once again it is not expressed as a mere list of instruments. As we will soon explain, all these aspects have been revised in order to obtain a more compact and abstract information. The introduction a this abstraction causes some informative loss, as well as it is evident that the information we are discarding could be useful to obtain more accurate results; but on the other hand this is both necessary and desirable in order to create a more conceptual view of the music features and to support the semi-automated genre classification. In the following, we will describe in detail our approach to score analysis for the different dimensions.

*Ensemble dimension.* A musical ensemble is defined as a group of musicians who gather to perform music. A degenerate case is given by a single performer. In a score, different parts are usually notated on different staves and an instrument name is given to each part. Of course, there are some exceptions: for instance, in Bach's *Art of Fugue* there is no explicit indication about the instrumental ensemble, and the score is playable by two hands on a keyboard but it is often performed by string or wind quartets, and sometimes even by a symphonic orchestra. Usually, a score

indicates if its parts should be performed by single players or by groups of persons; thus, there is also a quantitative aspect to take into account. From the qualitative points of view (i.e. the number and kind of real parts), a composition like Dvořak's *String Quintet in G major Op.77* and a movement for string orchestra such as Barber's *String Adagio* are indistinguishable: they both contain violin I, violin II, viola, cello, and double bass parts, but, of course, the number of performers involved makes the difference. For historical, stylistical, and practical reasons, some instruments or musical ensembles are constantly present in the history of music (e.g., the string quartet), while other are characteristic of a given period (e.g., the instrument known as "tromba marina", typical of the Renaissance and Baroque), and some other are simply incompatible with a number of classifications (e.g., an electric guitar with respect to Romantic music). Thus, the ensemble dimension, in its qualitative and quantitative aspects, is one of the most interesting and promising approaches to music classification, as it provides many indications for a correct arrangement of the piece as information against any incorrect classification. As regards the ensemble dimension, the aforementioned abstraction process consists in the transformation of a mere list of instrumental parts, such as two Violins, one Viola, and one Cello, into a more general and compact information, such as *String Quartet*.

*Rhythmic dimension.* In music theory, rhythm can be defined as the organization of the duration of sounds over time. In a score, rhythm is related at least to time signatures, to accent layout, and to rhythmic figures within a bar. For rhythm, we adopt another kind of aggregation, resulting in a sequence of time signatures. The segmentation we provide splits a score in a number of rhythmical episodes, where each episode is characterized by a different time signature. An aspect we take into account is the length of the single episode, expressed in terms of number of measures. Most pieces have only an initial time signature, however this information is interesting in order to exclude some classification possibilities. For instance, the dance named polonaise is typically in 3/4 time, and this trivial information is sufficient to distinguish polonaise from polka, the 2/4-beat dance of Czech origin. More interesting results can be achieved when the same music piece provides more time signatures. In fact, not only a contrasting time signature can exclude some possibilities (e.g., a standard waltz will never have sections in duple meter), but some rhythmic changes are also typical of characteristic forms, such as Baroque preludes or minuet and trio, which are typical of the Classical period.

*Harmonic dimension.* Harmony is the vertical aspect of music, related to the use and study of pitch simultaneity. Our reinterpretation of harmony consists in collapsing a vertical set of notes on a composite entity whose meaning is similar to the symbols used in continuo figuring and in chord analysis (see Figure 2). We are no more interested in the number of notes the chord is made of, nor in their actual layout. Accordingly, chords are expressed as a list of bichords without octave information, whose roots are the complete chord root. Pitch information is still present, but described in relative terms: for example, we do not define the first chord in Figure 2 as EGB triad, rather as the list (i.e., minor third, perfect fifth) on the first

Fig. 2. Figured Bass in J.S. Bach, *St. Matthew Passion*, *Recitative 57*

Fig. 3. Anton Webern, original series used in *Variations for piano op. 27*

degree of the current key. The order of the events is ignored in the reinterpretation of the harmonic dimension, too. The abstraction process we adopt for harmony simply creates the set of the chord types used in the composition we are analyzing. On the one hand, the detailed information about harmonic patterns gets lost, and this aspect prevents us from basing classification on peculiar harmonic behaviors; but, on the other hand, the mere percentage of some harmonic configurations can be very useful for classification purposes. In Renaissance music, for instance, major and minor triads and their first inversions were predominant, whereas contemporary notated music shuns them and introduces a number of chords that would never be conceived in the 16th century.

*Melodic dimension.* In music theory, melody is defined as a series of linear note events. Accordingly, the melodic aspect of scores is mainly related to notes, and in particular to their name, possible accidentals and related octave information. For our purposes, the abstraction process creates a mapping from detailed melodic patterns to one or more compatible scales. In other words, all the information we consider about a melodic fragment is the scale model(s) it belongs to. First, a segmentation process is required, in order to define a number of melodic fragments. This process can be hand-made or automatic, and in the latter hypothesis the segmentation rules exposed in [Cambouropoulos 1998] can be easily implemented in a computer system. As regards 1:N mappings among a melodic fragment and scale models, we think that it is necessary to allow a number of possibilities, as most melodic lines can fit many models. For instance, the melodic fragment in Figure 3 is proper only to the twelve-tone scale, the key used in dodecaphony. On the contrary,



Fig. 4. Orlando di Lasso, *Cantiones Duarum Vocum*, 3. *Oculus non vidit*

the line in Figure 4 is proper to the natural minor scale, to the melodic minor scale, and to the harmonic minor scale starting on A, and to all the gregorian modes (in particular, the Aeolian mode). In addition to project melodic fragments onto scale models, our approach discards any information about the original sequence order.

### 3. THE *MX-ONTO* ONTOLOGY

The context-based representation of music produced by the process of score analysis is described by means of the *MX-Onto* ontology, whose architecture is illustrated in Figure 5. The ontology is organized in two layers, namely the *Context Layer* and the *Genre Classification Layer*. The *MX-Onto* is implemented by means of the OWL language [Smith et al. 2004], and is organized in three OWL ontologies. The first ontology, called *mxonto-context*<sup>1</sup>, describes the *Context Layer* and contains the classes adopted for representing the music resource context that is derived from the score analysis process. The genre classification layer is implemented by the *mxonto-genre* OWL ontology<sup>2</sup>. The *mxonto-genre* ontology contains a classification of the musical genre dimensions. In fact, in order to deal with the complexity of the notion of genre we propose to think to the genre as a classification along different dimensions. Each dimension refers to a particular set of features and each music resource can be classified along one or more of these dimensions. Moreover, when we classify a particular music resource as an instance of a particular dimension class, we want to specify the strength of the membership relation, expressed by a membership value. The genre classification of music resources is based on the context of each specific resource, that is represented in the context layer. For this reason, the two layers are connected by a number of rule sets, expressed by means of the SWRL language [Horrocks et al. 2004], that specify how to derive a genre classification out of a context feature. The role of the rules is basically to support the human classification of music resources. In fact, the rules cannot determine the music genre in many cases, but they are used in order to ban specific genres from the set of available genres for a specific music resource or to suggest to the user a set of possible genres. The SWRL rules are specified in the comprehensive *mxonto*<sup>3</sup> OWL ontology that refers to the *mxonto-context* and to the *mxonto-genre* ontologies by means of the import functionalities of OWL.

#### 3.1 The Context Layer

The music context is represented in the *mxonto-context* OWL ontology by means of a set of classes and properties. In *mxonto-context*, a music resource is represented

<sup>1</sup><http://islab.dico.unimi.it/ontologies/mxonto-context.owl>

<sup>2</sup><http://islab.dico.unimi.it/ontologies/mxonto-genre.owl>

<sup>3</sup><http://islab.dico.unimi.it/ontologies/mxonto.owl>

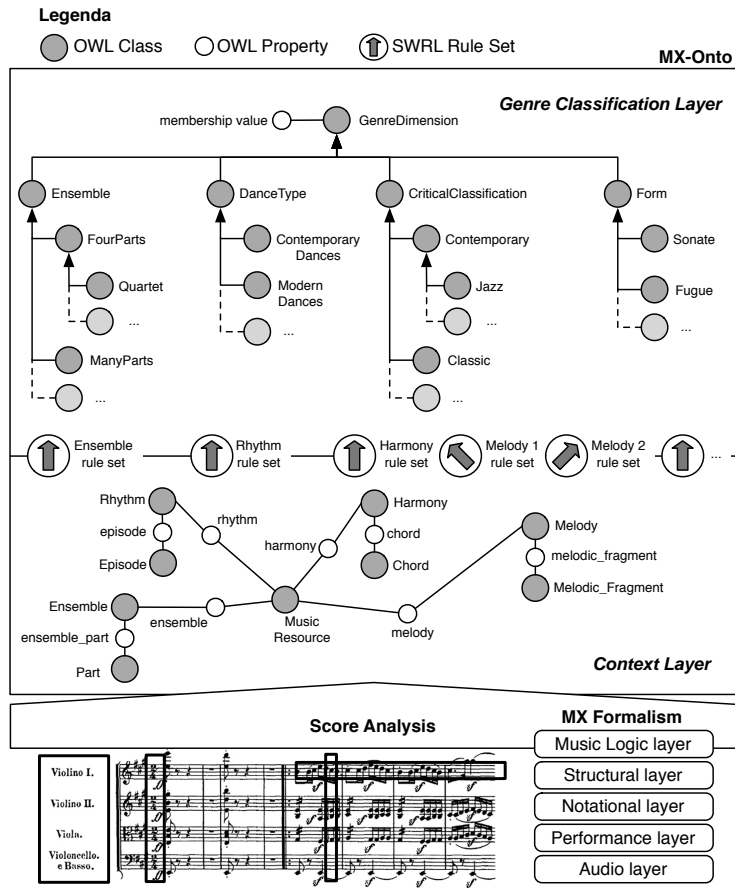


Fig. 5. The MX-Onto ontology for representation and classification of music resources

by an instance of the class `Music_Resource`, that is associated with its context that is represented by the `Ensemble`, `Rhythm`, `Melody`, and `Harmony` classes. Moreover, we have defined a class `Music_Feature` that represents the specific features adopted for characterizing each context dimension. The ensemble is described as a set of `Part` instances, each one characterized by the number of performers and by an instrument. The rhythm is described as a set of `Episode` instances that are characterized by a time signature with a numerator and a denominator (e.g., 3/4, 2/4) and by a number of measures that compose the episode. The melody is described by a set of `Melodic_Fragment` instances, each one characterized by the highest pitch, the lowest pitch and by a scale, that is associated with a first degree, represented as a pitch. Each pitch is represented as a `Note` instance that is characterized by an octave, a pitch, and an accidental (e.g., `Ab` on the third octave, `C` on the first octave). Finally, the harmony is seen as a set of `Chord` instances. Each chord is described by a scale, a fundamental degree, and a set of bichord components. A bichord component is the representation of the distance between two notes of the



**Allegro con brio.  $\text{♩} = 72$ .**

Violino I.  
Violino II.  
Viola.  
Violoncello.  
e Basso.

Fig. 6. A portion of the 4th movement of the *Symphony No.7 in A major* by Ludwig van Beethoven

chord, and it is described by a degree distance and a modifier (e.g., Major, Minor, Perfect). The chords are organized into a taxonomy (e.g., Bichord, Triad).

*Example.* As an example of context definition from a music score analysis, we consider a portion of the 4th movement of the *Symphony No.7 in A major* by Ludwig van Beethoven, that is shown in Figure 6. The melodic features of the score are represented by means of an instance `LVB_7th_4thM_Melody` of the class `Melody`. The melody is associated with a set of melodic fragments. In the example, we consider the fragment `A4_E3_AMj` that represents a melodic fragment associated with the A Major scale and characterized by the upper pitch A4 and the lower pitch E3, that is: <sup>4</sup>

```
Melodic_Fragment(A4_E3_AMj), melody_scale(A4_E3_AMj, A4_Major),
upper_pitch(A4_E3_AMj, A4), lower_pitch(A4_E3_AMj, E3)
```

where, `A4_E3_AMj` is an instance of the `Melodic_Fragment` class, that is associated with the `A4_Major` scale, and with two notes (i.e., A4, E3) which represent the upper pitch and the lower pitch in the fragment, respectively. The complete OWL definition of notes and scales is available at <http://islab.dico.unimi.it/ontologies/mxonto-context.owl>.

The rhythm is represented by an instance `LVB_7th_4th_Rhythm` of the class `Rhythm`, that is characterized by a set of episodes. Each episode is defined by a time signature and a duration expressed in the number of measures involved in the episode. For the example, we consider an episode of 8 measures associated with a time signature of 2/4, that is defined as follows:

```
Duple_Meter ⊆ Time_Signature, Duple_Meter(Duple_Meter_2.4),
Episode(Episode_2.4.8)
time_signature(Episode_2.4.8, Duple_Meter_2.4), number_of_measures(Episode_2.4.8, 8)
```

where `Duple_Meter_2.4` represents the time signature of the episode, while the role

<sup>4</sup>In the examples of the paper, we adopt the following notation in order to represent instances, properties and semantic relations of the OWL representation of the MX-Onto ontology: class instances are represented as a unary relation of the form `C(I)`, where `C` denotes the class name and `I` denotes the instance name. Property instances are represented as binary relations of the form `P(I1, I2)`, where `P` denotes the property name, while `I1` and `I2` denote the names of the instances that are involved in the property, respectively. Finally, the symbol `⊆` denotes subclass relations between classes.

`number_of_measures` denotes the length of the episode that is expressed by the number of measures it is composed by.

The harmony is represented by an instance `LVB.7th.4th.Harmony` of the class `Harmony`, that is associated with chords. Each chord is an instance of the class `Chord` and is characterized by a scale, a fundamental degree, and a bichord, where a bichord is described by an instance of the class `Bichord.Component` and is associated with a modifier and a degree distance. In our example, we have represented three chords, namely a first degree major triad second inversion on A, a fifth degree major triad root position on A, and a fifth degree dominant seventh on A. As an instance, we show in the following the definition of the major triad second inversion. The first step is to define, if not already present in the ontology, the bichords that compose the chord. In our example, we adopt the bichords `Major_Sixth` and `Perfect_Fourth`. Then, the chord is defined as follows:

```
Triad ⊆ Chord, Major_Triad ⊆ Triad, Mj_Triad_Second_Inversion ⊆ Major_Triad,
Mj_Triad_Second_Inversion ⊆ bichord : Major_Sixth,
Mj_Triad_Second_Inversion ⊆ bichord : Perfect_Fourth,
```

where the class `Mj_Triad_Second_Inversion` represents the kind of chord that is associated with the major sixth and perfect fourth bichords. Instances of the class `Mj_Triad_Second_Inversion` are then defined by associating them with a scale and a fundamental degree. The other chords of the example are defined analogously.

The ensemble is represented by an instance `LVB.7th.4thM.Ensemble` of the class `Ensemble`, that is associated with its components parts and by the number of parts. Each part is characterized by a set of instruments, given by the instrument taxonomy, and an optional number of performers. When the number of performers is omitted the part is considered to be associated with multiple performers, such as in the case of symphonic music. In the example, we consider three parts for multiple performers, namely `Violin_I`, `Violin_II`, `Viola`, and `Cello_and_Basso`, that are associated with the instruments that are played in each part, such as violin, viola, cello and basso.

### 3.2 The Genre Classification Layer

The genre classification is represented by the `mxonto-genre` OWL ontology in terms of a taxonomy of music genres and dimensions. The `GenreDimension` class is a generalization, since we have specific classes expressing the four dimensions that we decided to include in our ontology. The four classes representing genre dimensions are: i) `Ensemble`<sup>5</sup>: this class represents the class of music resource genres with respect to the parts involved in a particular performance (e.g., String Quartet); ii) `DanceType`: this class describes the classification of music genres with respect to rhythmic features about metre and accents disposition (e.g., Waltz, Polka); iii) `Critical`: this class describes the music genres with respect to critical and historical evaluations (e.g., Pre-Romantic, Contemporary); iv) `Form`: this class describes

<sup>5</sup>We have chosen to adopt the name `Ensemble` both for a feature of the context and for a genre dimension. In the first case we refer to the parts involved in a music score performance, while in the second case we refer to an ensemble-based genre, such as for instance a quartet. In the ontology, the two terms are distinguished through the namespace mechanism of OWL.

music genres with respect to the structure of a score (e.g., Fugue, Sonate). In the ontology we have defined an OWL class for each one of the genre dimensions described above, and for a number of specific music genres, and appropriate OWL properties for representing the attributes and the relations of each genre. A particular music resource will be associated with one or more instances of the `GenreDimension` hierarchy together with a specific membership value. In such a way, we can associate also the same music resource to multiple instances of the same dimension, overcoming the limitations typical of the classification methods based on predefined genre attribution. When a music resource instance  $r$  is classified within a genre class  $G$ , we want to associate to this membership relation a membership value that expresses the degree of membership of  $r$  in  $G$ . In our approach, the idea behind the fuzzy classification of music resources with respect to the genre taxonomy is that different users can share the same genre taxonomy but, at the same time, they are not required to agree about the classification of a specific music resource in terms of the genres in which a music resource is framed. For example, in the genre taxonomy the Pre-Romantic and the Romantic genres are defined. Although, different users adopt the same taxonomy, they can classify the same music piece into these two genres with a different degree of membership, according to their different understanding of music.

The problem of defining a degree of membership is typical of fuzzy logics and can be addressed in OWL by adopting two different strategies. The first strategy is to adopt a fuzzy extension of OWL that introduces specific constructs and a semantics for expressing fuzziness in OWL ontologies. Specific fuzzy extensions for description logics and OWL in particular have been proposed in [Straccia 2001; Stoilos et al. 2005]. The advantage of this strategy is that we can adopt new specific OWL constructs for supporting fuzzy membership of individuals into classes and that we have a formal semantics for these new constructs. However, the main disadvantage is that we adopt a non-standard OWL version that, furthermore, is not supported by the main tools that are nowadays adopted for working with OWL (e.g., Protégé). The second possible strategy is to provide a mechanism for representing fuzziness in OWL by adopting the standard DL version of OWL. An example of this approach is given in [Ding and Peng 2004] for probabilistic knowledge. In this case, the advantage is that the resulting ontologies are expressed in OWL without specific constructs and are supported by all the OWL-compatible tools for development and management of ontologies. In this paper, we have chosen to adopt this second strategy, in order to produce standard OWL-DL ontologies. However, we note that is simple to convert the `mxonto-genre` ontology into a OWL fuzzy extended ontology. For a discussion about the relations between standard OWL and Fuzzy OWL, the reader can refer to [Stoilos et al. 2005]. Our mechanism for representing fuzziness in standard OWL is based on the idea of defining a `Fuzzy_Membership` class, that is defined as follows:

$$\text{Fuzzy\_Membership} \sqsubseteq = 1 \text{ membership\_value} \sqcap \exists \text{ music\_resource.Music\_Resource} \quad (1)$$

where the properties `membership_value` and `music_resource` are mandatory and they associate each instance of `Fuzzy_Membership` with a membership value and, at least, a music resource, respectively. In other terms, the class `Fuzzy_Membership` represents a reification of the OWL membership relation, due to the fact that we need

to associate the membership value to the standard binary relation of membership. This solution to the problem of representing attributes of the relations has been proposed in [Noy and Rector 2004]. 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. The result is that only the genre classes that are set to be subclasses of the `Fuzzy_Membership` class can be instantiated in a fuzzy way, because they inherit the membership value and the music resource properties from `Fuzzy_Membership`. On the basis of this mechanism, the music resource classification is defined as follows:

*Definition 3.1. Classified music resource.* Given a genre  $G$ , where  $G$  is a subclass `Fuzzy_Membership`, a music resource  $r$  is classified with respect to  $G$  with a membership value  $m$  according to the following procedure: we define an instance  $g \in G$  and we associate it with  $r$  and  $m$ , by means of the properties `music_resource` and `membership_value`, that is:

$$\begin{aligned} G &\sqsubseteq \text{Fuzzy\_Membership}, \\ G(g), &\text{Music\_Resource}(r), \\ &\text{music\_resource}(g, r), \\ &\text{membership\_value}(g, m) \end{aligned}$$

*Example.* As an example of this procedure, let us consider the 4th movement of the *Symphony No.7 in A major* by Ludwig van Beethoven. In the example, we want to state that the score is classic and preromantic. Moreover, we will state that the classic feature is more relevant than the preromantic one for this score, by associating a membership value of 0.8 to the classic genre and a membership value of 0.3 to the preromantic genre. We note that the membership values determined for each genre are independent one from the other. In other words, the sum of the membership values of the different dimensions for a specific music resource can be less than 1.0 or even more than 1.0. This is due to the fact that the membership value denote the degree of membership of a music resource to a genre, and not the probability that a given music resource belongs to a genre. For instance, given to genres like `Opera` and `Operetta`, a user can denote a partial overlapping between the two classes by associating a music resource  $r$  with `Opera` and `Operetta` with membership value 0.8 and 0.5, respectively (e.g., *Les Contes d'Hoffmann* from J. Offenbach). The only constraints that are set on the membership values derive from the fuzzy interpretation of the ontology semantics. For example, the subclass relation is interpreted as follows:  $C_1 \sqsubseteq C_2$  is interpreted as  $C_1^{\mathcal{I}} \leq C_2^{\mathcal{I}}$ , where  $\mathcal{I}$  denotes an interpretation function. In the example, if we have two classes `Music_Drama` and `Opera`, that is `Opera`  $\sqsubseteq$  `Music_Drama`, the membership value associated with an instance of `Music_Drama` has to be higher than or equal to the membership value associated with `Opera`. For a complete interpretation of fuzzy OWL see [Stoilos et al. 2005]. Given the `Music_Resource` instance `LVB_7th_4thM` that represents the music piece to be classified, the first step is to define an instance of the `Preromantic` class and an instance of the `Classic` class, that is: `Preromantic(Preromantic_0.3)`, `Classic(Classic_0.8)` that represent music features that are preromantic with a degree of 0.3 and music resources that are classic with a degree of 0.8. The second step is to associate

```

...
xmlns:genre="http://islab.dico.unimi.it/ontologies/mxonto-genre.owl#"
xmlns:context="http://islab.dico.unimi.it/ontologies/mxonto-context.owl#"
<context:Music_Resource rdf:ID="LVB_7th_4thM"/>
...
<genre:Preromantic rdf:ID="Preromantic_0.3">
  <genre:music_resource rdf:resource="#LVB_7th_4thM"/>
  <genre:membership_value rdf:datatype="xsd:float">0.3</genre:membership_value>
</genre:Preromantic>
<genre:Classic rdf:ID="Classic_0.8">
  <genre:music_resource rdf:resource="#LVB_7th_4thM"/>
  <genre:membership_value rdf:datatype="xsd:float">0.8</genre:membership_value>
</genre:Classic>

```

Fig. 7. Example of the classification of the 4th movement of the Symphony No.7 in A major by Ludwig van Beethoven

the music resource with each of the two instances, together with the corresponding membership value. In the case of the preromantic classification, the association is defined as follows:

$$\begin{aligned} & \text{music\_resource}(\text{Preromantic\_0.3}, \text{LVB\_7th\_4thM}), \\ & \text{membership\_value}(\text{Preromantic\_0.3}, 0.3), \end{aligned}$$

where the first statement associates the genre with the music resource and the second statement associates the genre with the corresponding membership value. The classic classification is defined analogously. We note that, with this mechanism the can reuse the `Preromantic_0.3` instance for the classification of all the music resources that are considered to be preromantic with the same membership value of 0.3. The result of the classification is shown in Figure 7 by means of the OWL XML syntax.

### 3.3 The SWRL ontology rules for context-based classification

In order to support the process of classifying music resources on the basis of their context, we define a set of rules that associate a genre with a set of music resources. These rules have been defined by means of the SWRL rule language. In SWRL a rule is an implication between an antecedent and a consequent, where if the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold [Horrocks et al. 2004]. In our approach, the antecedent is used for capturing a set of conditions over the music resource context, while the consequent specifies the music resource genre. For the crisp classification the structure of a rule is given by a conjunction of conditions in the antecedent and by a class membership declaration in the consequent. For example, let assume to have a trivial rule stating that if the ensemble of a music resource  $r$  is composed by four parts and each part is played by only one performer, then  $r$  is a quartet. In this case, we need an antecedent that captures the music resource instances with an ensemble composed by four parts that are played by a single performer, while the consequent specifies that  $r$  is a quartet. The rule is specified as follows:

$$\begin{aligned} & \text{Music\_Resource}(?r) \wedge \text{ensemble}(?r, ?b) \wedge \\ & \text{number\_of\_parts}(?b, ?c) \wedge \text{swrlb : equal}(4, ?c) \wedge \\ & \text{ensemble\_part}(?b, ?d) \wedge \text{performers}(?d, ?e) \wedge \text{swrlb : equal}(1, ?e) \\ & \rightarrow \text{Quartet}(?r) \end{aligned} \quad (2)$$

where, the first line determines the ensemble  $b$  of the music resource  $r$ , the second line determines the number of parts of the ensemble and checks if they are four, the third line determines the number  $e$  of performers of each part  $d$  and checks that  $e$  is equal to one, and the fourth line classifies  $r$  as a quartet. For the fuzzy classification we adopt in our approach, we need to associate a membership value  $x$  with the result of the classification. This requires to get (or create if not already created) an instance of the class `Quartet` with a membership value equal to  $x$  and to associate the corresponding music resource with it. The rule (2) is reused and extended for the fuzzy classification, by adding a predicate to the antecedent in order to get an instance  $q$  of `quartet` with a membership value of  $x$  and by modifying the consequent in order to associate  $q$  with the music resource  $r$ . In the example, we want to state that  $r$  is a quartet with a degree of 1.0. The resulting rule is defined as follows:

$$\begin{aligned}
& \text{Music\_Resource}(?r) \wedge \text{ensemble}(?r, ?b) \wedge \text{number\_of\_parts}(?b, ?c) \wedge \\
& \text{swrlb : equal}(4, ?c) \wedge \text{ensemble\_part}(?b, ?d) \wedge \text{performers}(?d, ?e) \wedge \\
& \text{swrlb : equal}(1, ?e) \wedge \text{Quartet}(?q) \wedge \text{membership\_value}(?q, 1.0) \\
& \rightarrow \text{music\_resource}(?q, ?r)
\end{aligned} \tag{3}$$

The rules are also adopted for supporting the user in selecting the genres that cannot be used for classifying a music resource  $r$ . In this case, the rule associates  $r$  to these genres with a membership value of 0.0. An example of such a rule is given by considering the music resources with a duple meter time signature that cannot be classified as a waltz. This rule is defined as follows:

$$\begin{aligned}
& \text{Music\_Resource}(?r) \wedge \text{rhythm}(?r, ?b) \wedge \text{episode}(?b, ?c) \wedge \\
& \text{time\_signature}(?c, ?d) \wedge \text{Duple\_Meter}(?d) \wedge \\
& \text{Waltz}(?w) \wedge \text{membership\_value}(?w, 0.0) \\
& \rightarrow \text{music\_resource}(?w, ?r)
\end{aligned} \tag{4}$$

Less trivial rules can be defined by calculating the membership value on the basis of the number of occurrences of a given feature in a context. For example, the presence of a chromatic scale in the music resource  $r$  melody can be used for classifying  $r$  as dodecaphonic music. In this case, we can consider the proportion  $x$  between the number of melodic fragments that are associated with a chromatic scale with respect to the total number of melodic fragments of  $r$ . The resulting  $x$  is then adopted for determining the membership value of  $r$  with respect to the `Dodecaphony` class that represents dodecaphonic music. In many trivial cases, such as in the previous examples, the rules can be adopted in order to obtain a fully automated classification with the advantage of reducing the human activity and of determining the genres that are not compatible with a given music resource. In other cases, however, it is not possible to automatically determine the music genre out of the music resource context. For example, in the case of the 4th movement of the 7th symphony that we have presented in the previous sections, we can adopt a rule based on the harmony. This rule states that, given the proportion  $x$  of chords of type `Mj_Triad`, `Mi_Triad` and `Dominant_Seventh` in the context of a music resource  $r$ ,  $x$  can be considered as the membership value of  $r$  with respect to the classic and romantic genres. However, this statement does not distinguish between classic and romantic and, moreover, does not consider other typical chords of the romantic music. For this reason, such kind of rules are adopted only for suggesting a possible classification to the user. Then, the user can define the actual membership values

by taking into account his knowledge about the music resource, e.g., the author, the title, the historical period, and also the results provided by other rules.

#### 4. CONTEXT-DRIVEN PROXIMITY DISCOVERY OF MUSIC

The context-based representation of music and its classification with respect to the genre are adopted for supporting a semantic rich discovery of music resources. In particular, two main categories of searches are possible: i) *context-driven discovery* and ii) *genre-driven discovery*. The first category is based on the idea to exploit the context for discovering music resources characterized by some specific features of interest from a musicological point of view. The second category is based on the idea of looking for music resources starting from one or more genres of interest.

##### 4.1 Context-driven discovery of music

The context of a music resource provides a semantic description of the features of a music resource in terms of the four dimensions described in Section 3. Each dimension can be investigated alone or in combination with other dimensions in order to extract a number of music pieces characterized by a particular feature. Such kind of queries can be easily expressed by a conjunction of atoms as defined in the SWRL rule language. In general, a context-driven query is seen as a set of conditions that have to be satisfied by the context of the music resources that are retrieved by the query. An example of a query regarding the rhythmic dimension is given by the following:

$$\text{Music\_Resource}(?r) \wedge \text{rhythm}(?r, ?b) \wedge \text{episode}(?b, ?c) \\ \wedge \text{time\_signature}(?c, ?d) \wedge \text{Quintuple\_Meter}(?d)$$

This query retrieves all the music resources  $r$  made of at least an episode in quintuple meter (i.e., 5/4 or 5/8). The resulting list could contain for example: i) Fryderyk Chopin, *Third movement (Larghetto) from Piano Sonata No. 1 in C minor Op. 4*; ii) Pyotr Ilyich Tchaikovsky, *Second movement from Symphony No. 6 "Pathétique", Op.74*; iii) Gustav Holst, *Mars and Neptune* from *The Planets*. In the following, we propose other examples of context-based queries which could interest musicologists, performers, students or people interested in music. Each query takes into account only one of the dimensions involved in the context, but, of course, we can also combine different dimensions for a more specific query. As regards the melodic dimension, a user could invoke a query to determine all the scores containing at least one melodic fragment referable to a whole-tone scale. In this case, the ontology would return a list of compositions such as: i) Béla Bartók, *String Quartet No. 5*; ii) Claude Debussy, *L'isle joyeuse*; iii) Franz Liszt, *Die Trauergondel No. 1*. Also the harmony dimension presents interesting features to be investigated. For example, a context-based query could extract all the scores containing at least a Neapolitan sixth chord. For such a request, the answer could include: i) Franz Schubert, *Andante of Symphony in C major*; ii) Ludwig Van Beethoven, *Adagio of Sonata quasi una Fantasia op. 27 No. 2*; iii) Fryderyk Chopin, *Ballade No. 1 in G minor*. Finally, as far as the ensemble dimension is concerned, a cello player could look for all the compositions for solo cello. In this case, a possible list of results could be: i) Johann Sebastian Bach, *Suites for Solo Cello*; ii) Sergei Prokofiev, *Solo Cello Sonata op. 133*; iii) Anton Webern, *Drei kleine Stücke op. 11*.

## 4.2 Genre-based discovery of music by proximity

The genre classification of music can be exploited as a powerful criterion for music discovery. In many cases, the users searching for music have in mind a genre that they prefer and they are looking for music resources that are classified in the same or similar genres. In many other cases, they have an example of music resource and they are looking for music pieces that are similar to their example. In this section, we show how the genre classification can be exploited for answering these two kinds of queries. The idea is to see the different genres in *mxonto-genre* as different dimensions of a multi-dimensional search space, where the membership value is adopted for localizing a specific music resource in the space. The search space is defined as follows:

*Definition 4.1. Search space.* A search space  $S^n$  is a n-tuple of the form  $\langle G_0, \dots, G_n \rangle$ , where each item  $G_i \in S$  is a dimension of an Euclidean n-space  $\mathbb{R}^n$  that corresponds to a music genre and is represented by a vector of real numbers in the range  $[0,1]$ .

The music resource localization is defined as follows:

*Definition 4.2. Music resource localization.* Given a music resource  $r$  and a search space  $S^n$ ,  $r$  is localized in  $S^n$  by a n-tuple of the form  $\langle g_0, \dots, g_n \rangle$ , where each item  $g_i$  is a real number in the range  $[0,1]$  and represents the membership value associated with the classification of  $r$  in  $G_i$ .

As an example of music resource localization, we consider the example of Figure 7. In this case, the 4th movement of the Symphony No.7 in A major by Ludwig van Beethoven (LVB\_7th\_4thM) is classified to be classic with a degree of 0.8 and preromantic with a degree of 0.3. Along the other dimensions, we have an undetermined degree value, so that, for the sake of clearness, we can assume to work in a 2-dimensional space, where  $S^2 = \langle \text{Classic}, \text{Preromantic} \rangle$ . The localization of LVB\_7th\_4thM is given by the membership values associated with each of the dimensions, that is  $\langle 0.8, 0.3 \rangle$ . In the example, the music resource is represented by a point as shown in Figure 9. The search space and the music resource localization are exploited for defining two different types of query based on genre classification. The first type, called *Query-by-genre*, is based on the idea of selecting a portion of the search space, and return all the music resources that are localized into the selected area. Queries by genre are defined as follows:

*Definition 4.3. Query-by-genre.* A query-by-genre  $Q_G$  is a tuple of the form  $\langle S^n, P \rangle$ , where  $S^n$  denotes a search space with  $n$  dimensions, while  $P$  denotes a set of predicates that are joint by an AND or an OR clause. A predicate  $p \in P$  is an expression of the form  $c(G_i, m)$ , where  $G_i \in S^n$  is a genre dimension,  $c \in \{=|<|>|<|>|<|>|\leq|\geq\}$ , and  $m$  is a value in the range  $[0,1]$ .

Queries by genre are defined by means of a SQL-like template that is shown in Figure 8 together with two query examples. The FROM clause denotes the search space that is given by two genres both in Query 1 and in Query 2. The predicates of the WHERE clause are interpreted as a selection of a portion of the search space, so that all the music resource instances that are localized within the selected space portions are returned. A graphical interpretation of Query 1 and Query 2 is shown in Figure 9. In the case of the example, LVB\_7th\_4thM is returned as an answer to



Template	Query 1	Query 2
SELECT [DISTINCT] *	SELECT *	SELECT *
FROM <i>genre</i> [, ...]	FROM Classic, Preromantic	FROM Classic, Preromantic
WHERE <i>predicate</i> [, ...]	WHERE Classic $\geq$ 0.7 AND Preromantic $\leq$ 0.4	WHERE (Classic $\leq$ 0.7 AND Preromantic $\geq$ 0.4 AND Preromantic $\leq$ 0.6) OR Preromantic $\geq$ 0.8

Fig. 8. Queries by genre template and examples

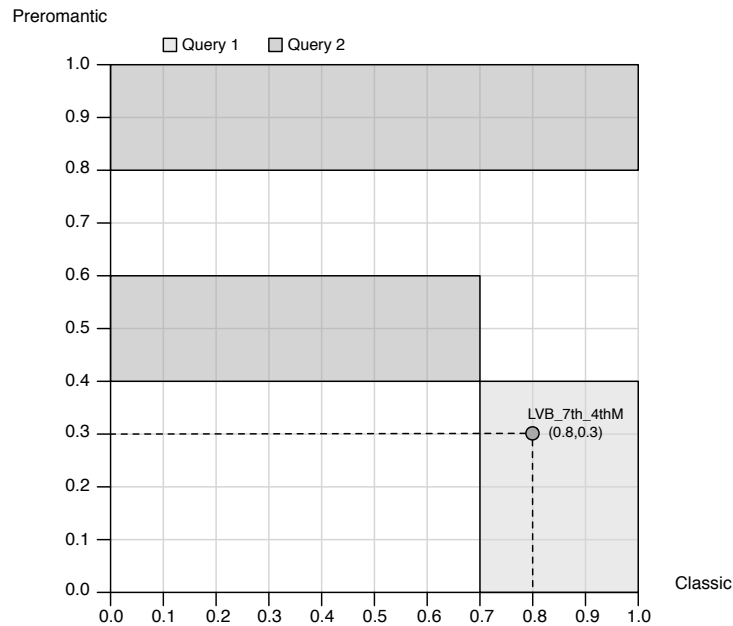


Fig. 9. Graphical interpretation of the two queries by genres of Figure 8

Query 1, while no music resources are retrieved for Query 2.

The second type of queries is called *Query-by-target*. The idea is that in these queries the user specifies a target of the query on a search space, that is, in the example of a 2-dimensional space, a point on the space. Then, the query is resolved by selecting the music resources that are localized in proximity of the target, where the wideness and the shape of the proximity space portion is defined in the query. The idea is that the shared classification provides a common base for genre-driven retrieval of music. On top of it, the queries-by-target have the goal of measuring the distance among different interpretation of a music piece. The result is that a user that searches for a music resource starting by a target (i.e., his personal classification of music resource) will retrieve not only similar music pieces (i.e., music resources that have been classified in a similar way) but also music resources that have been classified by users that have the same or similar understanding of the target music resource. Queries by target are defined as follows:

<b>Template</b>	<b>Query 3</b>	<b>Query 4</b>
SELECT [DISTINCT] <i>target</i>	SELECT (0.8,0.5)	SELECT (0.8,0.5)
FROM <i>genre</i> [, ...]	FROM Classic, Preromantic	FROM Classic, Preromantic
WHERE PROXIMITY = <i>prox-</i>	WHERE PROXIMITY = 0.3	WHERE PROXIMITY = 0.3
<i>imity</i>		WITH MODIFIER (0.5,6.0)
[WITH MODIFIER <i>modifier-</i>		
<i>list</i> ]		

Fig. 10. Queries by target template and examples

*Definition 4.4. Query-by-target.* A query-by-target  $Q_T$  is a 4-tuple of the form  $\langle S^n, T, M, R \rangle$ , where  $S^n$  is a n-dimensional search space,  $T$  is a localization over  $S^n$  and denotes the target of the query,  $M$  is a set of modifiers, where each modifier  $m_i$  is a positive real number associated with a genre  $G_i \in S^n$ , and  $R$  is a positive real number that denotes the degree of proximity for  $Q_T$ . The query  $Q_T$  selects a portion of  $S^n$  that is given by the following formula:

$$m_0 \cdot (g_0 - t_0)^2 + \dots + m_n \cdot (g_n - t_n)^2 = R^2$$

where  $g_i$  denotes a variable on the genre dimension  $G_i$ ,  $m_i \in M$  denotes the modifier associated with  $G_i$ ,  $t_i$  is the localization over  $G_i$ , and  $R$  is the degree of proximity.

Queries by target are defined according to a SQL-like template that is shown in Figure 10, together with two examples of queries. In the queries by target template, the SELECT clause is used for specifying the localization of the target with respect to the genres in the FROM clause. The interpretation of the localization is positionally determined. In the clause WHERE, the proximity degree is specified by the PROXIMITY clause, while the clause WITH MODIFIER specifies the modifiers ordered by position. If the clause WITH MODIFIER is omitted each modifier is set to 1.0. The modifiers are adopted in order to balance the impact of the different genres on the query results. For example in Query 4, we adopt the same target of Query 3 but we require a strict proximity (i.e., 0.12) along the preromantic dimension and a large tolerance (i.e., 0.42) along the classic dimension. A graphical representation of the search space portion selected by Query 3 and Query 4 is shown in Figure 11, where we deal with a 2-dimensional search space. In the case of the example, LVB\_7th\_4thM is returned as an answer to Query 3, while no music resources are retrieved for Query 4.

## 5. RELATED WORK

With respect to the MX-Onto ontology, relevant research work regards i) music metadata representation and ii) music resource classification.

### 5.1 Music metadata representation

A number of different music representation formats have been proposed in the literature with the aim to provide a formal representation of the musical information extracted from a given music resource [Selfridge-Field (Ed.) 1997]. Moreover, appropriate tools have been defined to exploit such representations for providing advanced search functionalities. For instance, Humdrum has been developed to support music researchers in a number of computer-based musical tasks [Huron

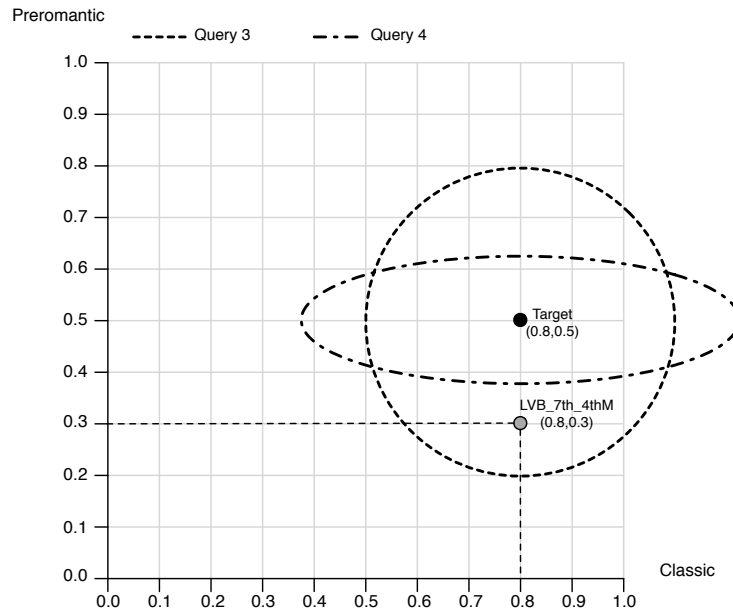


Fig. 11. Graphical interpretation of the two queries by target of Figure 10

1995]. In Humdrum, the *Humdrum Syntax* is defined as a grammar for music information representation, while the *Humdrum Toolkit* is defined to provide a set of utilities for exploiting data expressed in the Humdrum syntax. With respect to the Humdrum syntax, different pre-defined schemes are supported for music representation. In this respect, the *kern* representation is one of the most commonly used pre-defined representations that has been conceived to represent the core musical information of a music piece (e.g., notes, durations, rests, barlines) [Huron 2002]. As an example of the Humdrum Toolkit utilities, the Themefinder Web-based application is defined to perform advanced searches on a set of musical pieces described according to the Humdrum syntax [Huron]. In Themefinder, different techniques (e.g., search-keys, pitch contour, scale degree, date-of-composition) are supported to enable a user to search for musical themes and incipits. In the context of music metadata representation, the role of ontologies is becoming more and more relevant. In [Bohlman 2001], the author emphasizes the need to analyze the music domain from a conceptual point of view with the aim to define the *ontology of music*. To this end, different research projects have been devoted to develop an ontology of music capable of capturing and modeling the most important aspects of the music domain. In this direction, some interesting results have been appeared in the literature [Ranwez 2002; Harris 2005]. In more recent work, ontologies are recognized as a promising technology for music resource representation in a semantic way. In [Celma et al. 2004], the SIMAC (Semantic Interaction with Music Audio Contents) project for semantic description of music contents is presented. In SIMAC, the collective knowledge of a community of people interested in music

ACM Transactions on Multimedia Computing, Communications and Applications, Vol. V, No. N, April 2006.

is handled as metadata and integrated in a OWL music ontology. By exploiting music metadata, the SIMAC project aims to develop a set of prototypes to describe, semi-automatically, music audio content. The authors propose to enhance/extend the FOAF model for describing user musical tastes (i.e., user profile) in order to provide music content discovery based on both user profiling and content-based descriptions. The EV Meta-Model is presented in [Alvaro et al. 2005], as a new system for representing musical knowledge for computer-aided composition. The EV Meta-Model is proposed as a generic tool for multi-level representation of any kind of time-based event. Moreover, the EV Meta-Model is intended to be a dynamic representation system, capable of handling each element as a “living” variable, and transmitting such dynamic character to the music that it represents. The high level of abstraction provided by the EV Meta-Model allows the definition of an ontology for the music event representation and the EVscore Ontology is introduced as an example in this direction.

*Novel contribution of the context-based representation.* With respect to the previous approaches, the context-based representation proposed in the MX-Onto ontology is based on the MX formalism for music metadata extraction through score analysis. We note that the choice of MX is due to the peculiar advantages provided by such a formalization. In particular, the MX multi-layer structure enforces a flexible and extensible representation of the different degrees of abstraction in music information. Furthermore, the XML-based format simplifies the context information extraction by enforcing a Semantic Web compatible representation of a music resource. However, we stress that the context-based MX-Onto ontology is independent from the underlying encoding format used for music information extraction. In this respect, other existing formats (e.g., NIFF, MusicXML, MIDI) could be adopted to directly extract context information from music data, provided that an appropriate wrapper is developed.

## 5.2 Music resource classification

Genre, an intrinsic property of music, is probably one of the most important descriptor used to classify music resources. Traditionally, genre classification has been performed manually but some automatic approaches are being considered in recent literature. According to [Aucouturier and Pachet 2003], existing genre classification approaches can be organized with respect to three main categories: i) manual approach based on human knowledge and culture (*manual classification*); ii) automatic approach based on automatic extraction of audio features (*prescriptive classification*); iii) automatic approach based on objective similarity measures (*emerging classification*).

*Manual classification.* Manual classification of music resources is a time consuming activity that requires music experts involvement. The classification process starts defining an initial genre taxonomy which is gradually enriched when music titles are positioned in the taxonomy and new categories are required to provide a suitable arrangement for a given title. Examples of manual genre classifications are provided by traditional music retailers (e.g., Virgin Megastores, Universal Music) and

Internet music retailers (e.g., Amazon <sup>6</sup>, AllMusicGuide <sup>7</sup>, MP3 Internet site <sup>8</sup>). Traditional music retailers create a genre taxonomy for an internal need of product organization and with the aim at guiding consumers in shops. In this cases, taxonomies are poorly structured and rarely present more than four levels of detail. For what concerns Internet music retailers, taxonomies are created for supporting users while they navigate the music catalogues. In this case, such classifications present a high level of detail in terms of numbers of genre categories and maximum path length. Further examples of manual genre classifications are provided in the Microsoft MSN Music Search Engine project [Dannenberg et al. 2001], and in the Sony Cuidado Project [Pachet and Cazaly 2000], respectively.

The main effort of manual classification approaches is related to the genre taxonomy definition. In this activity, the participation of musicologists and music experts plays a crucial role for obtaining satisfying results in terms of level of detail and for avoiding possible lexical and semantic inconsistencies. In general, manual genre classifications are hard to manage and to maintain. Inserting and updating one or more categories involve the analysis of the entire taxonomy in order to prevent possible inconsistencies. Moreover, manual classifications are based on the subjective interpretation of their authors and, thus, comparisons and integrations among different taxonomies are rarely relevant.

*Prescriptive classification.* Prescriptive approaches attempt to extract automatically genre information from the audio signal. The prescriptive classification process can be distinguished in two main phases: the *feature extraction* phase and the *machine learning/classification* phase. In the feature extraction phase, the music signal of a song is decomposed into frames, and a feature vector of descriptors is computed for each frame. In the machine learning/classification phase, feature vectors are considered by a classification algorithm in order to automatically position the music title in a reference genre taxonomy. The classification phase starts with a supervised learning stage devoted to training the algorithm for the subsequent automatic classification process. The reference genre taxonomy is manually created before the beginning of the classification process. With respect to the feature vectors extraction from audio signal, different approaches can be distinguished. In [Tzanetakis and Cook 2000; Deshpande et al. 2001], feature vectors describe the spectral distribution of the signal for each considered frame, that is, they describe the global timbre that takes into account all the sources and instruments enclosed in the music. In [Lambrou et al. 1998; Tzanetakis et al. 2001; Soltau 1998], feature vectors are extracted by observing time and rhythm structure in the audio signal. For what concerns the machine learning/classification phase, different types of learning algorithms can be adopted during the training period. As described in [Tzanetakis and Cook 2000; Tzanetakis et al. 2001], a gaussian model can be used to estimate the probability density of each genre category over the feature space. Adopting a linear/non linear classifier, a neural network is used to learn the mappings between the dimensional space of the feature vectors and the genre categories in the reference taxonomy [Soltau 1998]. As described in [Deshpande

<sup>6</sup><http://www.amazon.com>

<sup>7</sup><http://allMusic.com>

<sup>8</sup><http://www.mp3.com>

et al. 2001], vector quantization techniques can be defined to identify the set of reference vectors that can quantify the whole feature set with little distortion. Prescriptive approaches are positively affected in terms of efficiency by the adoption of automatic techniques for genre information extraction from audio signals. Usually, the reference genre taxonomies are very simple and manually defined, that is, only broader categories (e.g., Classical, Modern, Jazz) are considered during the classification phase. As a consequence of the use of highly generic reference taxonomies, label ambiguities and inconsistencies can occur in the final classification. Label updates and progressive category insertions on the taxonomy could contribute to increase the flexibility of prescriptive approaches. Unfortunately, this strategy implies that the training stage for the learning algorithm and the music titles classification has to be executed each time a new category is inserted in the taxonomy. Furthermore, the feature set considered during the vectors extraction has an impact on the effectiveness of the final classification. Defining the best feature set is data dependent and such a selection should be dynamically performed according to the songs to be classified.

*Emerging classification.* Emerging classifications attempt to derive automatically the genre taxonomy by clustering songs on the basis of some similarity measure. In the emerging approaches, the signal-based feature extraction techniques adopted in the prescriptive approaches for deriving genre information can be used to evaluate the level of similarity among different music titles. Moreover, signal-based techniques can be combined with pattern-based data mining techniques that can improve the effectiveness of the similarity evaluation. In this respect, two main approaches can be considered: *collaborative filtering* [Shardanand and Maes 1995] and *co-occurrence analysis* [Pachet et al. 2001]. Collaborative filtering approaches are based on the idea that users with similar profile tend to prefer similar music titles. By comparing user profiles, it is possible to recognize recurrent patterns in music preferences and to use such observations to define clusters containing similar music titles [Pestoni et al. 2001; French and Hauver 2001]. Co-occurrence approaches aim at automatically identifying similar music titles by observing their neighborhood (co-occurrence) in different human-defined music sources (e.g., radio programs, CD albums, compilations). Clusters of similar music titles are defined by exploiting the distance matrix which counts the number of times that two titles occurred together in different sources, such as two radio programs with well-known music style addictions. Details regarding similarity measurements based on co-occurrence analysis techniques can be found in [Diday et al. 1981; Schütze 1992].

Experiments show that collaborative filtering and co-occurrence analysis succeed in distinguishing music genres by clustering similar music titles. Moreover, different types of data mining techniques can be combined to further improve the quality of clusters. We note that emerging approaches only work with music titles that appear in more than one source, otherwise pattern recognition is not possible. Furthermore, clusters are not labeled. Appropriate techniques for defining the correspondence between a cluster and a genre label are still required.

*Novel contribution of the context-based classification.* In Table I, we summarize some considerations about the comparison between the context-based classification

Table I. A comparison among different classification approaches

	Manual	Prescriptive	Emerging	Context-based
<b>Music features</b>	Audio signal Metadata	Audio signal	Metadata (e.g., title)	Score metadata (i.e., context)
<b>Feature extraction</b>	Manual (human knowledge)	Automatic (feature vectors)	Automatic (data mining)	Semi-automatic (SWRL rules)
<b>Taxonomy definition</b>	Crisp classification	Crisp classification	Clustering	Fuzzy classification

approach and the traditional classification techniques previously discussed (i.e., manual, prescriptive, and emerging). Regarding the music features considered for genre information extraction, we note that traditional classification techniques can exploit audio signal (e.g., manual, prescriptive) or metadata like the music title (e.g., manual, emerging). In this respect, the context-based classification approach can rely either on score and context metadata (i.e., ensemble, rhythm, harmony, and melody) analysis. The combination of such techniques provides accurate genre information, due to the fact that score and context metadata are a very expressive resource for acquiring structural and style features of a given music piece. Furthermore, the context-based approach joins the efficiency of the automatic techniques (e.g., prescriptive, emerging) with the accuracy of the manual classifications by providing a semi-automatic approach. In particular, SWRL rules enforce an automatic classification process in case that no ambiguities occur, while SWRL rules are exploited to support the user in fuzzy values specification in case that more than one option is offered. We stress that the fuzzy classification supported by the context-based approach fosters ontology management and evolution. With respect to crisp classifications (e.g., manual and prescriptive approaches), user defined fuzzy values allow new category insertions while preserving ontology consistency. Moreover, context-based proximity searches can be exploited to identify labeled clusters of similar music titles, where labels are obtained by combining the genre categories of the considered search space.

## 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 context associated with MX resources. The context representation is realized by means of an OWL ontology that describes music information and proposes rules and classes for music classification. The proposed classification is flexible with respect to the different interpretation of music genres, because it provides the possibility to have multiple relations of membership between a music resource and the music genres, instead of a partition of music genres which is typical of many classification approaches. For the future work, we have three main directions of activity. A first activity is the enrichment of the ontology with new classes and properties that capture further features of music information. These new features are also integrated in the software prototype of the classification and retrieval system that is currently under development. In the context of the software development activity, we are also collecting a complete set of experimentation of the proposed techniques in order to evaluate the advantages of our retrieval

techniques in several different test cases. Moreover, there are interesting feature of music, such as timbre, that can extracted from the audio signal (e.g., MP3) and can be used for enriching the ontology and the classification process. A second activity is to propose a methodology and techniques for comparing not only different classifications of music but also different taxonomies of musical genres by means of ontology matching techniques (see [Castano et al. 2005a; 2005b]). Finally, a third activity will be devoted to extend the experience of ontology-based representation of music to other multimedia resources, such as images and videos.

#### ACKNOWLEDGMENTS

A special acknowledgment is due to Denis Baggi for his invaluable work as working group chair of the IEEE Standard Association Working Group PAR1599 on Music Application of XML.

#### REFERENCES

- ALVARO, J., MIRANDA, E., AND BARROS, B. 2005. EV Ontology: Multilevel Knowledge Representation and Programming. In *Proc. of the 10th Brazilian Symposium of Musical Computation (SBCM)*. Belo Horizonte, Brazil.
- AUCOUTURIER, J. AND PACHET, F. 2003. Representing Musical Genre: A State of the Art. *Journal of New Music Research* 32, 1, 83–93.
- BOHLMAN, P. 2001. *Rethinking Music*. Oxford University Press, Chapter Ontologies of Music.
- CAMBOUROPOULOS, E. 1998. Musical parallelism and melodic segmentation. In *Proceedings of the XII Colloquium of Musical Informatics*. Gorizia, Italy.
- CASTANO, S., FERRARA, A., AND MONTANELLI, S. 2005a. Matching ontologies in open networked systems: Techniques and applications. *Journal on Data Semantics (JoDS) V*. (To Appear).
- CASTANO, S., FERRARA, A., AND MONTANELLI, S. 2005b. *Web Semantics and Ontology*. Idea Group, Chapter Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications. (To Appear).
- CELMA, O., RAMREZ, M., AND HERRERA, P. 2004. Semantic Interaction with Music Content using FOAF. In *Proc. of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*. Galway, Ireland.
- DANNENBERG, R., FOOTE, J., TZANETAKIS, G., AND WEARE, C. 2001. Panel: New Directions in Music Information Retrieval. In *Proc. of the Int. Computer Music Conference*. Habana, Cuba.
- DESHPANDE, H., NAM, U., AND SINGH, R. 2001. Classification of Music Signals in the Visual Domain. In *Proc. of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*. Limerick, Ireland.
- DIDAY, E., GOVAERT, G., LECHEVALLIER, Y., AND SIDI, J. 1981. *Digital Image Processing*. Kluwer edition, Chapter Clustering in Pattern Recognition, 19–58.
- DING, Z. AND PENG, Y. 2004. A probabilistic extension to ontology language owl. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4*. IEEE Computer Society, Washington, DC, USA, 40111.1.
- FRENCH, J. AND HAUVER, D. 2001. Flycasting: Using Collaborative Filtering to Generate a Playlist for Online Radio. In *Proc. of the Int. Conference on Web Delivering of Music (WEDELMUSIC 2001)*. Florence, Italy.
- GRUBER, T. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5, 2, 199–220.
- HARRIS, D. 2005. The KendraBase Web Site. [http://base.kendra.org.uk/music\\_ontology/](http://base.kendra.org.uk/music_ontology/).
- HAUS, G. 2001. Recommended Practice for the Definition of a Commonly Acceptable Musical Application Using the XML Language. IEEE SA 1599, PAR approval date 09/27/2001.
- ACM Transactions on Multimedia Computing, Communications and Applications, Vol. V, No. N, April 2006.



- HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABEL, S., GROSOFF, B., AND DEAN, M. 2004. SWRL: A semantic web rule language combining OWL and RuleML. W3C Member Submission 21 May 2004.
- HURON, D. The Themefinder Web Site. <http://www.themefinder.org/>.
- HURON, D. 1995. The Humdrum Toolkit Reference Manual. Tech. rep., Center for Computer Assisted Research in the Humanities, Menlo Park, CA, USA.
- HURON, D. 2002. Music information processing using the Humdrum Toolkit: Concepts, examples, and lessons. *Computer Music Journal* 26, 2, 15–30.
- LAMBROU, T., KUDUMAKIS, P., SANDLER, M., SPELLER, R., AND LINNEY, A. 1998. Classification of Audio Signals using Statistical Features on Time and Wavelet Transform Domains. In *Proc. of the IEEE Int. Conference on Acoustic Speech and Signal Processing (ICASSP)*. Seattle, Washington, USA.
- NOY, N. AND RECTOR, A. 21 July 2004. Defining n-ary relations on the semantic web: Use with individuals. Tech. rep., W3C Working Draft.
- PACHET, F. AND CAZALY, D. 2000. A Taxonomy of Musical Genres. In *Proc. of Content-Based Multimedia Information Access (RIAO)*. Paris, France.
- PACHET, F., WESTERMANN, G., AND LAIGRE, D. 2001. Musical Data Mining for Electronic Music Distribution. In *Proc. of the Int. Conference on Web Delivering of Music (WEDELMUSIC 2001)*. Florence, Italy.
- PESTONI, F., WOLF, J., HABIB, A., AND MUELLER, A. 2001. KARC: Radio Research. In *Proc. of the Int. Conference on Web Delivering of Music (WEDELMUSIC 2001)*. Florence, Italy.
- RANWEZ, S. 2002. Music Ontology. <http://www.daml.org/ontologies/276/>.
- SCHÜTZE, H. 1992. Dimensions of meaning. In *Proc. of Supercomputing 92*. IEEE Computer Society, Minneapolis, MN, USA, 787–796.
- SELFRIDGE-FIELD (ED.), E. 1997. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press.
- SHARDANAND, U. AND MAES, P. 1995. Social Information Filtering: Algorithms for Automating Word of Mouth. In *Proc. of the ACM Conference on Human Factors in Computing Systems*. Denver, Colorado, USA.
- SMITH, M. K., WELTY, C., AND MCGUINNESS, D. L. 2004. Owl web ontology language guide. W3C Recommendation 10 February 2004.
- SOLTAU, H. 1998. Recognition of Musical Types. In *Proc. of the Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seattle, Washington, USA.
- STOIOS, G., STAMOY, G., TZOUVARAS, V., PAN, J., AND HORROCKS, I. 2005. Fuzzy owl: Uncertainty and the semantic web. In *International Workshop of OWL: Experiences and Directions*. Available online as CEUR Workshop Proceedings, Galway, Ireland.
- STRACCIA, U. 2001. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research* 14, 137–166.
- TZANETAKIS, G. AND COOK, P. 2000. Audio Information Retrieval (AIR) Tools. In *Proc. of the Int. Symposium on Music Information Retrieval*. Bloomington, Indiana, USA.
- TZANETAKIS, G., ESSL, G., AND COOK, P. 2001. Automatic Musical Genre Classification of Audio Signals. In *Proc. of the Int. Symposium on Music Information Retrieval*. Plymouth, Massachusetts, USA.

Received Month Year; revised Month Year; accepted Month Year