An MPEG7 query language and a user preference model that allow semantic retrieval and filtering of multimedia content

ARTICLE *in* MULTIMEDIA SYSTEMS · AUGUST 2007

Impact Factor: 0.62 · DOI: 10.1007/s00530-007-0091-z

Stavros Christodoulakis
Technical University of Crete

Multimedia Systems
DOI 10.1007/s00530-007-0091-z

REGULAR PAPER

An MPEG-7 query language and a user preference model that allow semantic retrieval and filtering of multimedia content

Chrisa Tsinaraki · Stavros Christodoulakis

Abstract We present in this paper the MPEG-7 Query Language (MP7QL), a powerful query language that we have developed for querying MPEG-7 descriptions, as well as its compatible Filtering and Search Preferences (FASP) model. The MP7QL has the MPEG-7 as data model and allows for querying every aspect of an MPEG-7 multimedia content description. It allows the users to express the conditions that should hold for the multimedia content returned to them regarding semantics, low-level visual features and media-related aspects. The MP7QL queries may utilize the users' FASP and Usage History as context, thus allowing for personalized multimedia content retrieval. The FASP model supported is compatible with the MP7QL and has the model of the standard MPEG-7 FASPs as a special case. The proposed FASPs essentially are MP7QL queries. Both the MP7QL and its compatible FASP model allow for the exploitation of domain knowledge encoded using pure MPEG-7 constructs. In addition, they allow the explicit specification of boolean operators and/or preference values in order to allow both the combination of the query conditions according to the user intentions and the expression of the importance of the individual conditions for the users. The MP7QL query results are represented as MPEG-7 documents, guaranteeing the closure of the results within the MPEG-7 space. The MP7QL and the FASP model have been expressed using both XML Schema and OWL syntax. An implementation of the MP7QL, on top of an XML Native Database is currently in progress. A real world-world evaluation study on the expressive power of the MP7QL shows that it covers both general purpose and domain specific requirements in multimedia content retrieval.

Keywords MPEG-7 · MP7QL · Semantic retrieval and filtering · Personalization · Context-based queries

1 Introduction

An open multimedia consumption environment has been recently formed, due to three major reasons: (a) the development of digital multimedia content services that offer high content quality, advanced interaction capabilities, media personalization and adaptation according to the user preferences and access conditions; (b) the emergence of advanced network infrastructures that allow for the fast, efficient and reliable transmission of multimedia content; and (c) the availability and affordability of consumer electronic devices that allow the consumption and management of multimedia content like, for example, MP3 recordable players, digital cameras, DV camcorders and well-integrated smart phones. The users of such an open environment want the services provided by different vendors to interoperate. Such interoperation is achieved, at the syntactic level, through the adoption of standards.

The dominant standard in multimedia content description is the MPEG-7 [24]. MPEG-7 allows the description of (segments of) multimedia objects (i.e. images, audio and video) in terms of *media information* (including media format, quality

The work presented in this paper, originally presented in SMAP 2006, was partially funded in the scope of the DELOS II Network of Excellence in Digital Libraries (IST—Project Record N° 507618).

C. Tsinaraki (✉) · S. Christodoulakis
Technical University of Crete, Laboratory of Distributed Multimedia Information Systems and Applications (TUC/MUSIC), University Campus, 73100 Kounoupidiana, Crete, Greece
e-mail: chrisa@ced.tuc.gr
URL: http://www.music.tuc.gr

S. Christodoulakis
e-mail: stavros@ced.tuc.gr

Journal: 530 MS: 91 CMS: GIVE CMS □ TYPESET □ DISK □ LE □ CP Disp.:2007/6/6 Pages: 23

etc.), *creation information* (including title, creators, subject, related material etc.), *structure*, *usage information* (including rights, availability etc.), *textual annotations*, *media semantics*, *matching hints* for associating the media with audio and visual descriptors, the importance of the multimedia content from specific *points of view*, the *relations* of the multimedia content with other media and metadata items and the low-level *visual* and *audio* features of the multimedia content.

In addition to the syntactic interoperation, which is achieved through the use of well-accepted standards, semantic interoperation is also needed for providing efficient retrieval and filtering services. The semantic interoperation is typically achieved through the integration of domain knowledge, which is usually expressed in the form of domain ontologies. The domain knowledge is subsequently utilized for supporting semantic retrieval and filtering [26,29] and has been shown to enhance the retrieval precision [28]. It is also used for providing semantically personalized services. The semantic personalization is built on top of semantic user preference descriptions, which are used as content-related context during both retrieval (where the user preferences will be used in order to expand and/or disambiguate the user queries) and filtering (where the user preferences will be used as continuous queries that select the content to be returned to the user).

We have shown, in our previous research [27], that domain knowledge, in the form of domain ontologies, can be expressed using MPEG-7 constructs and integrated in the MPEG-7 semantic descriptions. The rich information captured in the MPEG-7 descriptions allows providing powerful retrieval and filtering capabilities on top of them.

Several research groups have been working on MPEG-7 (semantic, content and text) based multimedia content retrieval and filtering. The major limitation of these approaches is that each of them treats some aspects of the MPEG-7 based retrieval and filtering, but they do not provide a uniform and transparent MPEG-7 retrieval and filtering framework.

The proposals for uniform and transparent MPEG-7 retrieval and filtering support were either the use of plain *XQuery* [23] or the use of the existing MPEG-7 *Filtering and Search Preferences* (FASP) for MPEG-7 based multimedia content retrieval and filtering. Both these approaches do not exploit successfully the different MPEG-7 metadata description elements: the XQuery does not take into account the peculiarities of the MPEG-7 description elements and the MPEG-7 FASPs do not cover all the MPEG-7 description elements. This is an important limitation, as among the elements that are not exploited successfully are the semantic description elements, on top of which the semantic retrieval, filtering and personalization can be built. In addition, the MPEG-7 FASPs do not allow the explicit specification of boolean operators (that allow the accurate combination of the query conditions according to the user intentions), while the XQuery does not allow the explicit specification of preference values (that allow expressing the importance of the individual conditions for the users).

In order to overcome the limitations of the existing approaches, a query language for querying MPEG-7 descriptions is needed, with clear, MPEG-7 specific semantics (instead of the generic semantics of the XQuery). These semantics will allow the optimizers to effectively perform consistency checking and first-level optimization. In response to this need, the International Organization for Standardization (ISO) has recently issued the *MPEG-7 Query Format Requirements* [15], in order to guide the MPEG-7 query format standardization.

We present in this paper the MPEG-7 Query Language (MP7QL), a powerful query language that we have developed for querying MPEG-7 descriptions. The MP7QL has the MPEG-7 as data model and satisfies the ISO MPEG-7 Query Format Requirements [15]. It allows for querying every aspect of an MPEG-7 multimedia content description, including semantics, low-level visual features and media-related aspects. It also allows for the exploitation of domain knowledge encoded using pure MPEG-7 constructs. In addition, the MP7QL allows the explicit specification of boolean operators and/or preference values. The MP7QL queries may utilize the user preferences and the usage history as context, thus allowing for personalized multimedia content retrieval and filtering. The MP7QL FASP model allows expressing filtering and search preferences on every aspect of the MPEG-7 multimedia content descriptions. The MP7QL FASP model has the standard MPEG-7 FASPs as a special case, and, at the same time, extends our previous research for supporting semantic user preferences for multimedia content consumption [26]. The MP7QL query output has the form of MPEG-7 documents where the query results are represented as parts of standard MPEG-7 collections, guaranteeing that the MP7QL language has the closure property [5]. This allows the results of the query language expressions to be able to be stored as new MPEG-7 descriptions, and to be reused by the query language in a recursive manner.

The aforementioned features show that the MP7QL and its compatible FASP model overcome the limitations of the existing approaches for uniform and transparent MPEG-7 retrieval and filtering. In particular, they allow the exploitation of all the elements of the MPEG-7 descriptions. In particular, through the exploitation of the semantic elements, they can effectively support semantic retrieval, filtering and personalization. In addition, they allow the accurate expression of the end-user conditions through the explicit specification of both boolean operators and preference values. A real world-world evaluation study on the expressive power of the MP7QL shows that it covers both general purpose and domain specific requirements in multimedia content retrieval.

The MP7QL has been expressed using both XML Schema [7] and OWL [20] syntax. The implementation of the MP7QL is in progress, on top of an XML native database accessed by XQuery.

The rest of this paper is structured as follows: in Sect. 2, an overview of the MPEG-7 standard is provided, focusing on MPEG-7 based multimedia content description and on the MPEG-7 FASP model. Related work is presented in Sect. 3. In Sect. 4 we describe the input format of the MP7QL query language, in Sect. 5 we describe the output format of the MP7QL query language and in Sect. 6 we describe the MP7QL FASP model. We present how the MP7QL query language and its compatible FASP model can be used in several contexts as well as a real world-world evaluation study of their expressive power in Sect. 7 and we conclude in Sect. 8, where our future research directions are also outlined.

2 Overview of the multimedia content description interface (MPEG-7)

We provide in this section an overview of the MPEG-7 standard for multimedia content description. The MPEG-7 constructs are defined using the *MPEG-7 Description Tools*, which are expressed in XML Schema syntax.

The MPEG-7 description tools are the *Descriptors*, the *Description Schemes* and the *Classification Schemes*. A *Descriptor* (*D*) represents a multimedia feature and defines the syntax and the semantics of the feature representation. A *Description Scheme* (*DS*) provides descriptive information and specifies the structure and the semantics of the relationships between its components, which may be both Descriptors and Description Schemes. A *Classification Scheme* (*CS*) essentially is a thesaurus comprised of term hierarchies.

In the rest of the section, we will present the multimedia content description capabilities of the MPEG-7 in Sect. 2.1 and the MPEG-7 FASPs in subsection 2.2.

2.1 MPEG-7 based multimedia content description

In this subsection, we describe the multimedia content description capabilities of the MPEG-7. The MPEG-7 multimedia content descriptions are represented as instances of the subtypes of the abstract type *MultimediaContentType*. The subtypes of *MultimediaContentType* allow the description of all the classes of multimedia objects and are shown in Table 1.

Table 1 The subtypes of *MultimediaContentType*

Type name	Description type
ImageType	Image Content Description
VideoType	Video Content Description
AudioType	Audio Content Description
AudioVisualType	Audiovisual Content Description
MultimediaType	Multimedia Content Description
SignalType	Signal Content Description
AnalyticEditedVideoType	Analytic Content Description
InkContentType	Ink Content Description
MultimediaCollectionType	Multimedia Collection Description

An MPEG-7 multimedia content description consists of the following description units:

- The *Media Information*, which is captured in one of the *MediaInformation*, *MediaInformationRef* and *MediaLocator* elements. The media information includes the unique identification of the media object and its locator, as well as media-related information (including media format, quality etc.).
- The *Creation Information*, which is captured in one of the *CreationInformation* and *CreationInformationRef* elements. The creation information consists of information about the media object *creation* (including title, creators, abstract etc.), *classification* (including genre, subject, language etc.) as well as information about *related material*.
- The *Textual Annotation*, which is captured in the *TextAnnotation* element and consists of the following elements, each of which may occur arbitrary times:
  (a) the *FreeTextAnnotation* element, which represents free text annotations;
  (b) the *StructuredAnnotation* element, which represents structured textual annotations in terms of *who* (people and animals), *what object*, *what action*, *where* (places), *when* (time), *why* (purpose) and *how*;
  (c) the *KeywordAnnotation* element, which represents keyword annotations; and
  (d) the *DependencyStructure* element, which represents textual annotations with a syntactic parse based on dependency structures.
- The *Structural Information*, which is captured in the *StructuralUnit* element and describes the role of the current multimedia object (segment) within the information context. Thus, the *StructuralUnit* may take values like "scene", "shot", "story" etc.
- The *Usage Information*, which is captured in one of the *UsageInformation* and *UsageInformationRef* elements. The usage information consists of information about the *rights* associated with the multimedia object, its *financial results*, its *availability* and its *usage record*.
- Information regarding the importance of the multimedia content from specific *points of view*. This information is captured in the *PointOfView* element.

Table 2 The Relationship Classification Schemes

Classification scheme name	Relationship types
BaseRelation CS	Basic relationship types (equals, inside, covers, overlaps, touches, refines etc.).
GraphRelation CS	Graph node relationship types (identity, equivalent etc.).
SpatialRelation CS	Spatial relationship types (over, below, south, northwest, above etc.).
TemporalRelation CS	Temporal relationship types (precedes, follows, meets, during, contains etc.).
SemanticRelation CS	Semantic relationship types (shows, agent, causer, experiencer etc.).

Table 3 The subtypes of *SemanticBaseType*

Type name	Description
SemanticType	Concrete type used for the description of collections of semantic entities
AgentObjectType	Concrete type used for the description of the actors that appear in a multimedia object. The actors are specified in the *Agent* element of AgentObjectType. Actors in general are represented using the subtypes of the abstract type *AgentType*. *PersonType*, *OrganizationType* and *PersonGroupType* are the subtypes of AgentType and are used for the representation of persons, organisations and groups of persons respectively.
ObjectType	Concrete type used for the description of objects and object abstractions in the material world.
EventType	Concrete type used for the description of events that take place in a semantic world.
ConceptType	Concrete type used for the description of concepts present in an audiovisual segment.
SemanticStateType	Concrete type used for the description of a state of the world described in an audiovisual segment and the parametric description of its features.
SemanticPlaceType	Concrete type used for the description of a place in a semantic world.
SemanticTimeType	Concrete type used for the description of semantic time.

- The *Relationships* of the multimedia content with other media or metadata items as well as the relationships of the semantic entities describing the multimedia content. This information is captured in the *Relation* element, which associates the media object descriptions with instances of the *RelationType* that represent relationships. A relationship may be directed or undirected and features a relationship *type*, the *target* and the *source* of the relationship and the *strength* of the relationship. The standardised MPEG-7 relationship types are more than 100 and are classified in the classification schemes shown in Table 2.
- The *Matching Hints*, captured in the *MatchingHint* element, which allow expressing the criteria for matching the multimedia content with low-level audio and visual descriptors.
- The *Semantic Annotation*, captured in the *Semantic* element, where a set of semantic entities describing the multimedia content are defined or referenced. The semantic entities are instances of the subtypes of the abstract type *SemanticBaseType*, which represent semantic entities of specific types in a narrative world (see Table 3). The *AbstractionLevel* element of the *SemanticBaseType* specifies if a semantic entity is abstract or concrete.
- The low-level *Visual* and *Audio* features of the multimedia objects that have a visual and/or audio component. The visual features are captured in the *VisualDescriptor* and the *VisualDescriptionScheme* elements using, respectively, visual descriptors and visual description schemes defined in [11] and the audio features are captured in the *AudioDescriptor* and the *AudioDescriptionScheme* elements using audio descriptors and audio description schemes defined in [12]. The descriptors that represent the temporal order of the visual features of moving regions structured according to the description schemes defined in [11] are captured in the *VisualTimeSeriesDescriptor* element.

The semantic multimedia content description capabilities of MPEG-7 are general purpose and do not support directly the integration of domain knowledge expressed in the form of domain ontologies. As a consequence, the domain-specific information is captured in the textual parts of the semantic entities (e.g. in labels or textual definitions). For example, a soccer field would be represented as a semantic place with the keyword phrase “soccer field” in its label or its definition. This way, false drops may occur if some semantic entities have, by chance, some keywords in their textual parts (for example, if the phrase “next to the soccer field” exists in the definition of a neighbouring shop).

A methodology for the systematic integration, in MPEG-7 semantic descriptions, of domain knowledge expressed using pure MPEG-7 constructs (i.e. abstract semantic entities and the MPEG-7 relationships *generalizes/specializes*, *exemplifies/exemplifiedBy* and *property/propertyOf*) has been described in [27]. Using this methodology, the soccer fields are related with relationships of type *exemplifies* with an abstract semantic entity that represents the class of all the soccer fields and the abstract semantic entity is related with relationships of type *exemplifiedBy* with every soccer field.

2.2 The MPEG-7 filtering and search preferences (FASP)

The MPEG-7 allows the users to express their preferences regarding multimedia content retrieval and filtering. This is achieved with the *FilteringAndSearchPreferences* (*FASP*) element of the MPEG-7 user preferences, which are defined in the MPEG-7 MDS [13] and are presented here. A FASP element is decomposed in the following sets of sub-elements:

- A set of *CreationPreferences* elements, which describe the user preferences regarding multimedia content creation.
- A set of *ClassificationPreferences* elements, which describe the user preferences regarding the multimedia content classification attributes.
- A set of *SourcePreferences* elements, which describe the user preferences regarding the multimedia content source.
- A set of *PreferenceCondition* elements, which describe, in terms of time and place, the usage conditions for the current FASP description.
- A set of FASP elements, which describe the sub-preferences of the current element, thus allowing the definition of FASP preference hierarchies.

The above elements and their sub-elements have the *preferenceValue* attribute, which allows the users to state which query conditions are more important for them. Notice that the MPEG-7 FASPs do not allow the explicit specification of boolean operators. In addition, they do not include elements that allow the users to express their preferences regarding the multimedia content semantics and the low-level features of the multimedia content. Thus, their expressive power is very limited, as they cannot support semantic retrieval, filtering and personalization.

3 Related work

In this section, we present the research in MPEG-7 based multimedia content retrieval, filtering and multimedia content service personalization that is relevant with the MP7QL.

We have shown, in Sect. 2, that the MPEG-7 allows the creation of rich multimedia content descriptions that provide information on several aspects of the multimedia content. Such descriptions allow the support of powerful retrieval and filtering functionality on top of them. Several research groups have been working on MPEG-7 based multimedia content retrieval and filtering, usually exploiting different elements of the MPEG-7 descriptions. These research efforts can be classified in three categories:

- Systems that exploit the *Visual* [11] and/or *Audio* [12] MPEG-7 Descriptors for content-based (low-level feature based) multimedia content retrieval [3,4,6]. Such systems support similarity queries of the form “give me video segments that contain a region similar to this in one of their frames”.
- Systems that utilize the textual annotations and/or the elements that describe the media related features of the MPEG-7 descriptors [8,22,25,30] for multimedia content retrieval and filtering support. Such systems support queries of the form “give me MPEG-2 video segments created by the Eurosport that contain in any of their elements the keywords ‘Italy’, ‘France’ and ‘goal’”.
- Systems that utilize the semantic metadata descriptions formed according to the *Semantic DS* of the MPEG-7 *Multimedia Description Schemes* (*MDS*) [13] in order to provide semantic-based multimedia content retrieval and filtering support [1,9,26,27]. Such systems support queries of the form “give me video segments that contain goals of France against Italy”.

The major limitation of the above systems is that they treat some aspects of the MPEG-7 based retrieval and filtering, but they do not provide a uniform and transparent MPEG-7 retrieval and filtering framework and cannot support queries that combine conditions on textual, media related, semantic and low-level features. Thus, none of these systems can answer queries like “give me the MPEG-2 video segments created by the Eurosport that contain goals of France against Italy and contain a region similar to this in one of their frames” (the image region given as an example may be the face of a player).

A first research effort for the establishment of a uniform and transparent MPEG-7 retrieval and filtering framework was the use of plain *XQuery* [23] on top of an XML repository for MPEG-7 based multimedia content retrieval [19]. This system does not make use of domain knowledge. The major limitation of this approach is that it does not take into account the following peculiarities of the MPEG-7 description elements: (a) the MPEG-7 semantic model is expressed in an involved way; (b) the domain knowledge integrated in the semantic MPEG-7 descriptions is expressed in the document level; and (c) the low-level visual and audio features should be evaluated using specialized functions. As a consequence, these elements cannot be successfully exploited if they are accessed in the same way with the textual and the media-related elements of the MPEG-7 metadata descriptions. These limitations make it difficult for the average user to express, using plain XQuery, semantic and content-based queries and even more difficult to combine such query conditions with textual and media-related query conditions. In addition, XQuery does not support queries with preference values to allow the users to state which query conditions are more important for them.

Another approach is to use the existing MPEG-7 *filtering and search preferences* (*FASP*) as (instant or continuous) queries that allow multimedia content filtering and retrieval. The MPEG-7 FASPs are very limited in their power, targeting to model preference hierarchies (not complex queries or filtering requests), and in particular preference hierarchies related to user interests in movies. The limitations of this approach are the following: (a) several MPEG-7 description elements are not present in the MPEG-7 FASPs. The most important among these elements are the semantic elements and the low-level visual and audio features; and (b) the boolean operators AND/OR/NOT cannot be explicitly specified in the MPEG-7 FASPs. These limitations do not allow the expression of queries for every aspect of the MPEG-7 descriptions. In addition, due to the lack of domain-specific semantic support, they can support neither semantic multimedia content retrieval and filtering nor semantic multimedia content service personalization (beyond the movie domain).

Proposals for the extension of the MPEG-7 FASPs with semantic elements expressed using the constructs of the MPEG-7 Semantic DS have been presented in [2,26]. The model presented in [26] also allows the explicit specification of boolean operators. These efforts are in the right direction, but they do not cover all the description elements missing from the MPEG-7 FASPs.

In order to overcome the limitations of the existing approaches, a language for querying MPEG-7 descriptions is needed, with clear, MPEG-7 specific semantics. The MP7QL, which is presented in the rest of this paper, satisfies this requirement, as it has the MPEG-7 as data model and allows the explicit specification of both boolean operators and preference values. Having the MPEG-7 as data model allows the MP7QL to express complex queries that combine different types of conditions.

4 The input format of the MP7QL query language

We present in this section the input format of the MP7QL query language that allows querying MPEG-7 descriptions. The input format of the MP7QL allows querying every aspect of an MPEG-7 multimedia object description. The MP7QL queries may utilize the user preferences and the usage history as context, thus allowing for personalized multimedia content retrieval.

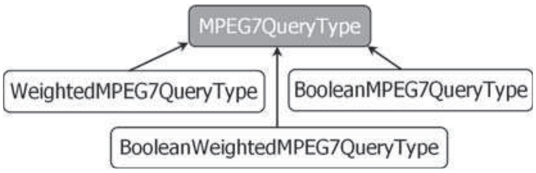


Fig. 1 The MP7QL query type hierarchy

The rest of this section is structured as follows: In Sect. 4.1 we describe the MP7QL query, which is the fundamental MP7QL construct. In Sect. 4.2 we present the MP7QL query conditions, which are defined in the context of the MP7QL query specifications.

4.1 MP7QL queries

In this subsection, we present the structure and the semantics of the MP7QL queries. The MP7QL queries are represented by the *MPEG7QueryType* abstract type. MP7QL allows the explicit specification of boolean operators and preference values for the MP7QL query elements. Three subtypes of MPEG7QueryType have been defined for the representation of all the possible types of queries: (a) the *WeightedMPEG7QueryType*, which represents queries with explicit preference values. The preference values are integers in the range $[-100, 100]$, with default value 10. The type and the range of the preference values are compatible with the preference values used in the MPEG-7 FASPs and the default value is the same with the default value of the preference values of the MPEG-7 FASPs; (b) the *BooleanMPEG7QueryType*, which represents queries with explicit boolean operators (AND/OR/XOR/NOT); and (c) the *BooleanWeightedMPEG7QueryType*, which represents queries with explicit preference values and boolean operators. The MP7QL query type hierarchy is depicted in Fig. 1 (the shading of the root node expresses that the *MPEG7QueryType* type is abstract).

An MP7QL query has a SELECT-FROM-WHERE syntax and is formally described using the regular expression syntax of (1). As shown in (1), all the MP7QL query elements are optional. This way, empty queries that allow browsing the multimedia repository contents are supported.

$$Q = [Select][From][Where][OrderBy][GroupBy] \quad (1)$$

The *Select* element of an MP7QL query allows the specification of the elements and/or attributes of the MPEG-7 descriptions that will be returned in the query results. If the *Select* element is not present in an MP7QL query, the query results will be formed in the default way (see Sect. 5 for details about the query results). The *Select* element of an MP7QL query is formally described using the regular expression syntax of (2).

$$\begin{aligned}Select = {} & Item^{*}\ [format][transformationRules]\\ & [maxItems][num\ Of\ PageItems]\\ & [page][timeLimit] \qquad (2)\end{aligned}$$

The *Item* elements of *Select* represent, in the form of XPath expressions, the elements and/or attributes of the MPEG-7 descriptions that should be returned for each of the query results. The *format* attribute represents the URI of the file, where the structure of the output display format (that is, the format in which the query results will be displayed in the user's terminal device) is specified and has as default value the URI of the default query output format. The *transformationRules* attribute represents the URI of the XSL stylesheet [18] that should be applied in the standard MP7QL output in order to be presented according to a different format. The *maxItems* attribute represents the maximum number of the query results that will be returned to the user and has "unbounded" as default value. The *numOfPageItems* attribute represents the number of the query results that will be displayed in each result page and has 10 as default value. The *timeLimit* attribute represents the time limit in seconds until which the query must be replied and has 300 as default value. The *page* attribute specifies which result page should be returned to the user and has 1 as default value.

The *From* element of an MP7QL query allows the specification of the search domain through the selection of the type(s) of the MPEG-7 descriptions on which the query will be posed and is formally described using the regular expression syntax of (3).

$$From = FromItem^{*} \qquad (3)$$

The *FromItem* elements of *From* may take predefined string values that specify the type(s) of the MPEG-7 descriptions on which the query will be posed. The search domain may be multimedia content descriptions of one or more types (for example, "give me the images where Marques is shown"), the semantic entities that satisfy specific criteria and can be used for the reusable semantic descriptions of multimedia content (for example, "give me the players affiliated to the soccer team Barcelona") or the domain ontology constructs expressed using MPEG-7 syntax (for example, "give me the subclasses of the *SoccerPlayer* class"). The allowed values of the *FromItem* elements of *From* are: (a) all the multimedia content entity names shown in Table 1; (b) "AllMultimediaDescriptions", which is the default value and states that the search domain includes all the multimedia content descriptions, independent of their type; (c) "SemanticEntityDefinition", which states that the search domain includes all the reusable semantic entities; and (d) "Ontology", which states that the search domain includes all the domain ontology constructs expressed using MPEG-7 syntax.

The *OrderBy* element of an MP7QL query allows the specification of the criteria for ordering the result set and is formally described using the regular expression syntax of (4).

$$OrderBy = Criterion^{*} \qquad (4)$$

The *Criterion* elements of *OrderBy* represent ordering criteria and are formally described using the regular expression syntax of (5).

$$Criterion = Item\ [priority][order] \qquad (5)$$

The *Item* element of *Criterion* represents, in the form of an XPath expression, an element or an attribute of the MPEG-7 descriptions, on which the ordering will be based. The *priority* attribute represents the priority of the element/attribute in ordering and has 0 as default value. The *order* attribute represents the type (ascending or descending) of the ordering based on the current element/attribute and has "ascending" as default value.

The *GroupBy* element of an MP7QL query allows the specification of the attribute or element that will be used for grouping the query results. The *GroupBy* element has the form of an XPath expression that describes the attribute or the element of the query results on which the grouping will be based.

The *Where* element of an MP7QL query allows the expression of the query conditions set by the user. The structure of the *Where* element is different for the different types of the MP7QL queries. In particular:

1. The *Where* element of an MP7QL query with explicit preference values (WWhere) is formally described using the regular expression syntax of (6).

   $$WWhere = (WQS\ pv)^{*} \qquad (6)$$

   pv is an explicit preference value and *WQS* is a query specification with explicit preference values (formally described in (10)). The query specification represents the query conditions set by the user.
2. The *Where* element of an MP7QL query with explicit boolean operators (BWhere) is formally described using the regular expression syntax of (7).

   $$\begin{aligned}&BWhere\\ &\quad = BQS[NOT]\ ((AND \mid OR \mid XOR)\ BQS\ [NOT])^{*} \qquad (7)\end{aligned}$$

   BQS is a query specification with explicit boolean operators (formally described in (14)).

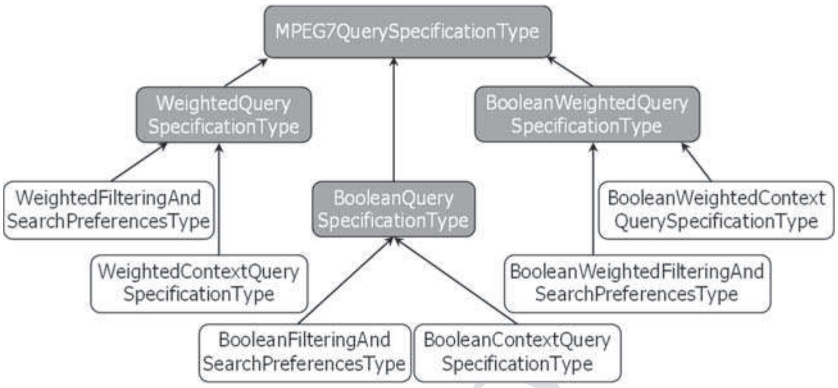

MPEG7QuerySpecificationType
WeightedQuery SpecificationType
BooleanWeightedQuery SpecificationType
WeightedFilteringAnd SearchPreferencesType
BooleanQuery SpecificationType
BooleanWeightedContext QuerySpecificationType
WeightedContextQuery SpecificationType
BooleanWeightedFilteringAnd SearchPreferencesType
BooleanFilteringAnd SearchPreferencesType
BooleanContextQuery SpecificationType

Fig. 2 The type hierarchy of the MP7QL query specifications

3. The *Where* element of an MP7QL query with explicit preference values and boolean operators (BWWhere) is formally described using the regular expression syntax of (8).

$$\begin{aligned} &BWWhere \\ &\quad = BWQS\ pv\ ((AND \mid OR \mid XOR)\ BWQS\ pv)* \qquad (8) \end{aligned}$$

   BWQS is a query specification with explicitly specified preference values and boolean operators (formally described in (19)).

A detailed discussion on the MP7QL query specifications is provided in Sect. 4.2.

4.2 MP7QL query specifications

We present in this subsection the MP7QL query specifications, which contain the query conditions specified by the users.

The MP7QL query specifications are represented by the abstract type *MPEG7QuerySpecificationType*, which is specialized according to the presence/absence of explicit boolean operators and/or preference values as shown in Fig. 2: the *WeightedQuerySpecificationType* represents query specifications with explicit preference values, the *BooleanQuerySpecificationType* represents query specifications with explicit boolean operators and the *BooleanWeightedQuerySpecificationType* represents query specifications with explicit boolean operators and preference values. According to Fig. 2, these types of query specifications are abstract and are further specialized into concrete types that represent filtering and search preferences (*WeightedFilteringAndSearchPreferencesType*, *BooleanFilteringAndSearchPreferencesType* and *BooleanWeightedFilteringAndSearchPreferencesType* respectively) and query specifications that allow the use of a query context during query evaluation (*WeightedContextQuerySpecificationType, BooleanContextQuerySpecificationType* and *BooleanWeightedContextQuerySpecificationType* respectively). The query context is either the *FASP* and the *Usage History* of the user or a set of ad-hoc query specifications that represent the user preferences for this query specification.

The query specifications have been designed to allow expressing conditions on every aspect of a multimedia object that has been described using MPEG-7, so that the MP7QL may be used for querying any MPEG-7 multimedia object description. Thus, every element of an MPEG-7 multimedia object description has a corresponding query specification element in the MP7QL query specifications. The corresponding query specification element is used to impose conditions on the values of the MPEG-7 element, which conditions should hold for the segments retrieved. In order to satisfy the requirement of the MPEG-7 Query Format Requirements [15] stating that the existing MPEG-7 FASPs should be valid query specifications, we decided to use the same naming and typing scheme with the MPEG-7 FASPs for all the elements that exist in the MPEG-7 FASPs. The rest of the MP7QL query specification elements (except from the elements used for "query-by-example" support) follow the naming and typing scheme of the MPEG-7 multimedia object descriptions. The MP7QL query specifications include the (optional) elements shown in Table 4.

The ordering and string comparison operators applied on the elements of the MP7QL query specifications may be explicitly specified and may take the values shown in Tables 5 and Table 6, respectively.

Variables are provided in MP7QL, in order to support joins on the conditions about the features of the MPEG-7 descriptions. From a syntactic point of view, a variable is an identifier that begins with the "$" character.

The MP7QL input query format has been expressed using both XML Schema syntax and OWL syntax.[1] An implementation of the MP7QL using the XML Schema syntax is in progress on top of an XML native database that is accessed by XQuery.

[1] The XML Schema and the OWL syntax of the MP7QL are available at http://www.music.tuc.gr/delos/resources/MP7QL_XS.zip and http://www.music.tuc.gr/delos/resources/MP7QL_OWL.zip, respectively.

Table 4 The MP7QL Query Specification Elements

MP7QL query specification element name and acronym	MP7QL query specification element description	Corresponding MPEG-7 element name
MediaIdentification (MI)	Conditions on the identification of the query results.	MediaIdentification (MPEG-7 Description Element)
MediaProfile (MP)	Conditions on the media features (i.e. media format, quality etc.) of the query results.	MediaProfile (MPEG-7 Description Element)
MediaLocator (ML)	Conditions on the actual media comprising the query results.	MediaLocator (MPEG-7 Description Element)
StructuralUnit (SU)	Conditions on the structure of the requested items.	StructuralUnit (MPEG-7 Description Element)
CreationPreferences (CrP)	Conditions on the creation details of the requested items (i.e. title, creators, related material etc.).	CreationPreferences (MPEG-7 FASP Element)
ClassificationPreferences (ClP)	Conditions on the classification of the requested items (i.e. language, genre, etc.).	ClassificationPreferences (MPEG-7 FASP Element)
SourcePreferences (SoP)	Conditions on the disseminator of the requested items (i.e. dissemination source, format etc.).	SourcePreferences (MPEG-7 FASP Element)
Semantic (SeP)	Conditions on the semantics of the content of the requested items. These conditions are very important in event-based environments (like sports). They are also used in queries about reusable semantic entities and domain ontologies.	Semantic (MPEG-7 Description Element)
PreferenceCondition (PC)	Conditions that should hold, in terms of place and time, for the query specification to be taken into account.	PreferenceCondition (MPEG-7 FASP Element)
UsageInformation (UI)	Conditions on the usage of the requested items (i.e. rights, availability etc.).	UsageInformation (MPEG-7 Description Element)
MatchingHint (MH)	Conditions on the matching of low-level descriptor features with the media element features.	MatchingHint (MPEG-7 Description Element)
PointOfView (PoV)	Conditions on the importance of the multimedia content from specific points of view.	PointOfView (MPEG-7 Description Element)
RelatedMaterial (RM)	Conditions on the related material of the requested items.	RelatedMaterial (MPEG-7 Description Element)
Relation (R)	Conditions on the relationships of the requested items with other media or metadata items.	Relation (MPEG-7 Description Element)
TextAnnotation (TA)	Conditions on the textual annotations of the requested items.	TextAnnotation (MPEG-7 Description Element)
VisualDescriptor (VD), VisualDescriptionScheme (VDS)	Reference to the low-level visual features that should be matched with the corresponding low-level visual features of the requested media items.	VisualDescriptor, VisualDescription-Scheme (MPEG-7 Description Elements)
AudioDescriptor (AD), AudioDescription-Scheme (ADS)	Reference to the low-level audio features that should be matched with the corresponding low-level visual features of the requested media items.	AudioDescriptor, AudioDescription-Scheme (MPEG-7 Description Element)
VisualTimeSeriesDescriptor (VTSD)	Reference to the descriptors that represent the temporal order of the visual features of moving regions that should be matched with the corresponding low-level visual features of the requested media items.	VisualTimeSeriesDescriptor (MPEG-7 Description Element)
DescriptorRef (DR), SemanticEntityRef (SER)	Reference to the existing MPEG-7 descriptions that should guide query-by-example queries about multimedia content descriptions (DR) and reusable semantic entities and domain ontologies (SER).	–

Table 5 Allowed values of the ordering operator

Ordering operator value	Description
equals	Succeeds if the value of the query specification element equals to the value of the corresponding metadata description element (default value).
greaterThan	Succeeds if the value of the query specification element is greater than the value of the corresponding metadata description element.
greaterThanOrEqual	Succeeds if the value of the query specification element is greater than or equal to the value of the corresponding metadata description element.
lessThan	Succeeds if the value of the query specification element is less than the value of the corresponding metadata description element.
lessThanOrEqual	Succeeds if the value of the query specification element is less than or equal to the value of the corresponding metadata description element.
differentFrom	Succeeds if the value of the query specification element is different from the value of the corresponding metadata description element.

Table 6 Allowed values of the String Comparison Operator

String comparison operator value	Description
contains	Succeeds if the value of the query specification element is contained in the value of the corresponding metadata description element (default value).
equals	Succeeds if the value of the query specification element equals to the value of the corresponding metadata description element.
startsWith	Succeeds if the value of the query specification element equals to the start of the value of the corresponding metadata description element.
endsWith	Succeeds if the value of the query specification element equals to the end of the value of the corresponding metadata description element.
keywords	Succeeds if every word contained in the query specification element is contained in the value of the corresponding metadata description element.
notContains	Succeeds if the value of the query specification element is not contained in the value of the corresponding metadata description element.

An MP7QL query example is shown, in formal syntax, in (9) and in XML syntax in Fig. 3.

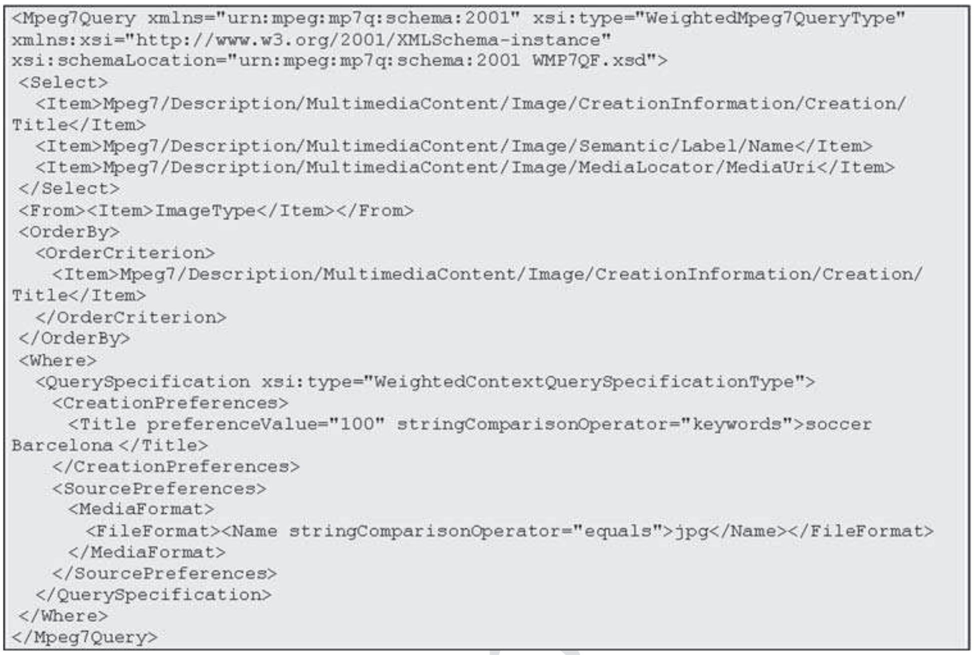
$$
\begin{aligned}
BQS1 = (&Select(Item(Mpeg7/Description/\\
&MultimediaContent/Image/\\
&CreationInformation/Creation/Title)\\
&Item(Mpeg7/Description/MultimediaContent/\\
&Image/Semantic/Label/Name)\\
&Item(Mpeg7/Description/MultimediaContent/\\
&Image/MediaLocator/MediaUri))\\
&From(FromItem(ImageType))\\
&OrderBy(Item(Mpeg7/Description/\\
&MultimediaContent/Image/\\
&CreationInformation/Creation/Title))\\
&From(VideoType)\\
&Where(BQS(CrP\ (Title(\text{'soccer''Barcelona'})\\
&keywords))\\
&SoP(MediaFormat(FileFormat(jpg)))))
\end{aligned}
\tag{9}
$$

This is a query with explicit preference values, which asks for the descriptions of the JPEG images that contain in their title the keywords “soccer” and “Barcelona”. The results will contain the titles of the image descriptions, the labels of the semantic parts of the descriptions and the URIs of the images. The ordering of the results will be ascending, based on the titles of the descriptions.

In the next paragraphs we provide details about the different types of MP7QL query specifications.

Query specifications with explicit preference values. In the query specifications with explicit preference values a preference value may be explicitly specified for every query specification element. The query specifications with explicit preference values are represented by the subtypes of the *WeightedQuerySpecificationType*, as shown in Fig. 2.

These are the *WeightedContextQuerySpecificationType* type, which represents queries for which a query context may be specified and the *WeightedFilteringAndSearchPreferencesType* type represents MP7QL FASPs with explicit preference values.



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001" xsi:type="WeightedMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 WMP7QF.xsd">
 <Select>
   <Item>Mpeg7/Description/MultimediaContent/Image/CreationInformation/Creation/
Title</Item>
   <Item>Mpeg7/Description/MultimediaContent/Image/Semantic/Label/Name</Item>
   <Item>Mpeg7/Description/MultimediaContent/Image/MediaLocator/MediaUri</Item>
 </Select>
 <From><Item>ImageType</Item></From>
 <OrderBy>
   <OrderCriterion>
     <Item>Mpeg7/Description/MultimediaContent/Image/CreationInformation/Creation/
Title</Item>
   </OrderCriterion>
 </OrderBy>
 <Where>
   <QuerySpecification xsi:type="WeightedContextQuerySpecificationType">
     <CreationPreferences>
       <Title preferenceValue="100" stringComparisonOperator="keywords">soccer
Barcelona </Title>
     </CreationPreferences>
     <SourcePreferences>
       <MediaFormat>
         <FileFormat><Name stringComparisonOperator="equals">jpg</Name></FileFormat>
       </MediaFormat>
     </SourcePreferences>
   </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 3 MP7QL query with explicit preference values which asks for the image descriptions that contain in their title the keywords “soccer” and “Barcelona” using XML syntax

The *WeightedContextQuerySpecificationType (WQS)* is formally described in the regular expression syntax of (10).

$$\begin{aligned}\mathit{WQS} = ((&\mathit{MI}|\mathit{MP}|\mathit{ML}|\mathit{SU}|\mathit{CrP}|\mathit{ClP}|\mathit{SoP}|\mathit{SeP}|\mathit{PC}|\mathit{UI}\\ &|\mathit{MH}|\mathit{PoV}|\mathit{RM}|\mathit{R}|\mathit{TA}|\mathit{DR}|\mathit{SER}\\ &|\mathit{VD}|\mathit{VDS}|\mathit{AD}|\mathit{ADS}|\mathit{VTSD}|\mathit{UH}|\mathit{FASP})\ \mathit{pv})^*\end{aligned} \tag{10}$$

MI is a media identification element, *MP* is a media profile element, *ML* is a media locator element, *SU* is a structural unit element, *CrP* is a creation preferences element, *ClP* is a classification preferences element, *SoP* is a source preferences element, *SeP* is a semantic preferences element, *PC* is a preference condition element, *UI* is a usage information element, *MH* is a matching hint element, *PoV* is a point of view element, *RM* is a related material element, *R* is a relation element, *TA* is a textual annotation element, *DR* is a multimedia description reference element, *SER* is a semantic entity reference element, *VD* is a reference to a visual descriptor element, *VDS* is a reference to a visual description scheme element, *AD* is a reference to an audio descriptor element, *ADS* is a reference to an audio description scheme element, *VTSD* is a reference to a visual time series descriptor element, *UH* is a reference to a usage history element and *FASP* is a filtering and search preferences element (or a reference to a filtering and search preferences element). All the WQS elements may also have explicit preference values. This way, the users may specify the search and filtering conditions as well as the importance of each of the conditions.

Semantic multimedia object retrieval, semantic multimedia content filtering and semantic multimedia service personalization are supported through the semantic elements. A semantic element with explicit preference values *(WSeP)* is formally described in (11).

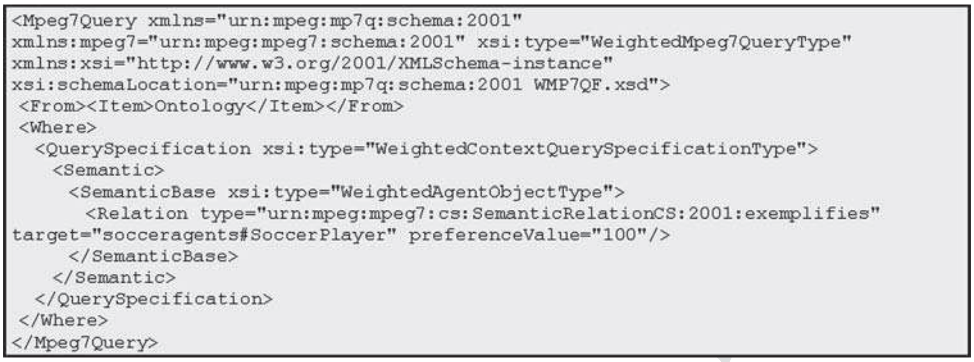
$$\mathit{WSeP} = (\mathit{WSE}\ \mathit{pv})^* \tag{11}$$

WSE is a set of search and filtering conditions on semantic entities with explicit preference values, and is formally described in (12).

$$\begin{aligned}\mathit{WSE} = {}&[\mathit{SID}]\ \mathit{SType}\ \mathit{pv}\ ((\mathit{AName}\ \mathit{AValue}\ \mathit{pv})\ |\\ &(\mathit{EName}\ \mathit{EValue}\ \mathit{pv}\ (\mathit{EAName}\ \mathit{EAValue}\ \mathit{pv})^*\\ &(\mathit{E}\ \mathit{pv})^*)\ |\ (\mathit{RType}\ \mathit{RTarget}\ \mathit{RSource}\\ &\mathit{RStrength}\ \mathit{pv}))^*\ \mathit{maxOccurs}\ \mathit{minOccurs}\end{aligned} \tag{12}$$

A WSE may be identified by *SID*, which plays the role of a variable name. The desired type (*SType*) of the semantic entity may also be specified, as well as: (a) the name (*AName*) and the desired value (*AValue*) respectively of the attributes of the semantic entity; (b) descriptions of the desired values of the semantic entity elements, including the element name (*EName*), the element value (*EValue*), the list of the desired element attribute values represented by attribute name (*EAName*) – desired attribute value (*EAValue*) pairs and the list of the desired values of its sub-elements *(E)*; (c) Relationship description information, consisting of the type (*RType*), the target (*RTarget*), the source (*RSource*) and the strength (*RStrength*) of the relationship. The minimum and maximum number of occurrences of a semantic entity that satisfies a set of conditions is specified in the values of *minOccurs* (with default value 1) and *maxOccurs* (with default value unbounded) respectively.

A query specification with explicit preference values for the retrieval of images has been shown in the query of Fig. 3.



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="WeightedMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 WMP7QF.xsd">
 <From><Item>Ontology</Item></From>
 <Where>
   <QuerySpecification xsi:type="WeightedContextQuerySpecificationType">
     <Semantic>
       <SemanticBase xsi:type="WeightedAgentObjectType">
         <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"
target="socceragents#SoccerPlayer" preferenceValue="100"/>
       </SemanticBase>
     </Semantic>
   </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 4 MP7QL query stating "I want the semantic entities that belong to the SoccerPlayer class", using XML syntax

Another example of query specification with explicit preference values is shown in the query expressed, in formal syntax, in (13) and in XML syntax in Fig. 4.

$$\mathit{BQS1} = \mathit{(From(Ontology)\ Where(BQS(SeP\ (AgentObjectType\ (exemplifies,\ SoccerPlayer)\ 100))))} \tag{13}$$

This is a query on a soccer ontology that asks for the semantic entities that belong to the "SoccerPlayer" class (which is an abstract semantic entity that represents the class of the soccer players).

Query specifications with explicit boolean operators. In the query specifications with explicit boolean operators, the *AND/OR/XOR operators* and the *NOT operator* that should be applied in the element contents may be explicitly specified for each query specification element. The default value of the AND/OR/XOR operator is "OR" and the default value of the NOT operator is "false" (meaning that the users by default would like the media elements that satisfy the specific conditions to be returned to them). The query specifications with explicit boolean operators are represented by the subtypes of the *BooleanQuerySpecificationType*, as shown in Fig. 2. These are the *BooleanContextQuerySpecificationType* type represents queries for which a query context may be specified and the *BooleanFilteringAndSearchPreferencesType* type represents user FASPs with explicit boolean operators.

The *BooleanContextQuerySpecificationType* and the *BooleanFilteringAndSearchPreferencesType* query specifications (*BQS*) are formally described using the regular expression syntax of (14). All the BQS elements may have explicit boolean operators.

$$\begin{aligned}\mathit{BQS} = {} & \mathit{(MI|MP|ML|SU|CrP|ClP|SoP|SeP|PC|UI|MH\ |}\\ & \mathit{PoV|RM|R|TA|DR|SER|VD|VDS|AD|ADS|}\\ & \mathit{VTSD|\ UH|FASP)\ [NOT]\ (AND|OR|XOR)}\\ & \mathit{(MI|MP|ML|SU|CrP|ClP|SoP|SeP|PC|UI|MH\ |}\\ & \mathit{PoV|RM|R|TA|DR|SER|VD\ |}\\ & \mathit{VDS|AD|ADS|VTSD|UH|FASP)[NOT])*}\end{aligned} \tag{14}$$

A semantic preference element with explicit boolean operators (BSeP) is formally described in (15).

$$\mathit{BSeP} = \mathit{[NOT]\ BSE\ ((AND\ |\ OR\ |\ XOR)\ [NOT]\ BSE)*} \tag{15}$$

BSE is a set of search and filtering conditions on semantic entities with explicit boolean operators, and is formally described in (16).

$$\begin{aligned}\mathit{BSE} = {} & \mathit{[SID]\ SType\ ((AND\ |\ OR\ |\ XOR)[NOT]}\\ & \mathit{(AName\ AValue)\ |\ (EName\ EValue}\\ & \mathit{(EAName\ EAValue)*\ (E)*)\ |}\\ & \mathit{(RType\ RTarget\ RSource\ RStrength))*}\\ & \mathit{maxOccurs\ minOccurs}\end{aligned} \tag{16}$$

An MP7QL query specification with explicit boolean operators is provided in the MP7QL query that states "I want the multimedia objects where a goal is scored by *Marques*", which is shown, expressed in formal syntax, in (17).

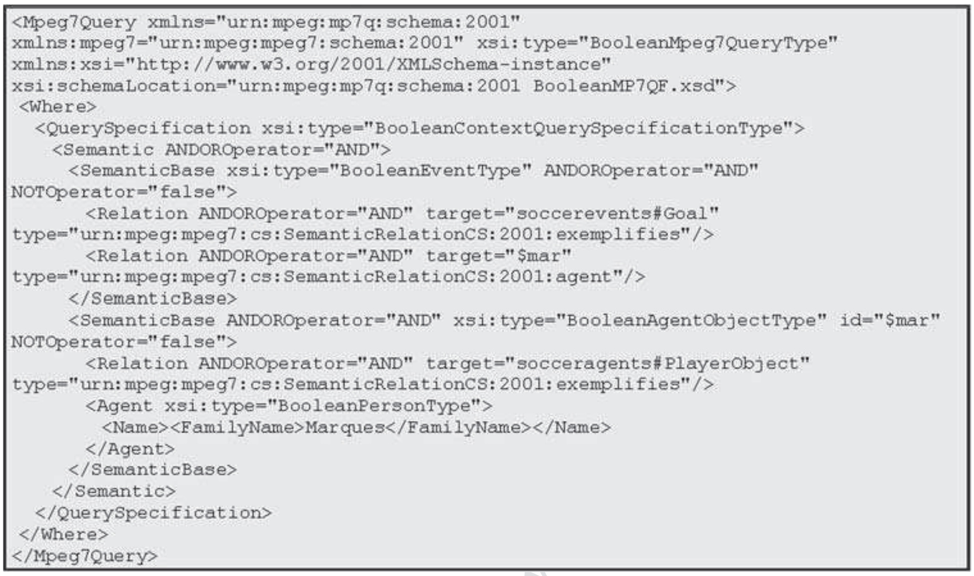
$$\begin{aligned}\mathit{BQS1} = {} & \mathit{(Where(BQS(SeP(EventType\ AND}\\ & \mathit{(exemplifies,\ Goal)\ AND\ (agent,}\\ & \mathit{\$\ mar))\ AND\ ((\$\ mar,\ AgentObjectType)\ AND}\\ & \mathit{(exemplifies,\ PlayerObject,\ \$\ mar)\ AND}\\ & \mathit{(Agent(Name(FamilyName\ 'Marques')))))))}\end{aligned} \tag{17}$$

We assume in this example that the abstract semantic entities "PlayerObject" and "Goal" exist, which represent the classes of all the players and all the goals respectively. We also assume that the soccer player Marques is bound to the "$mar" variable. The same query specification is shown in the query of Fig. 5, expressed using XML syntax.

Another example of query specification with explicit boolean operators is shown in the query expressed, in formal syntax, in (18) and in XML syntax in Fig. 6.

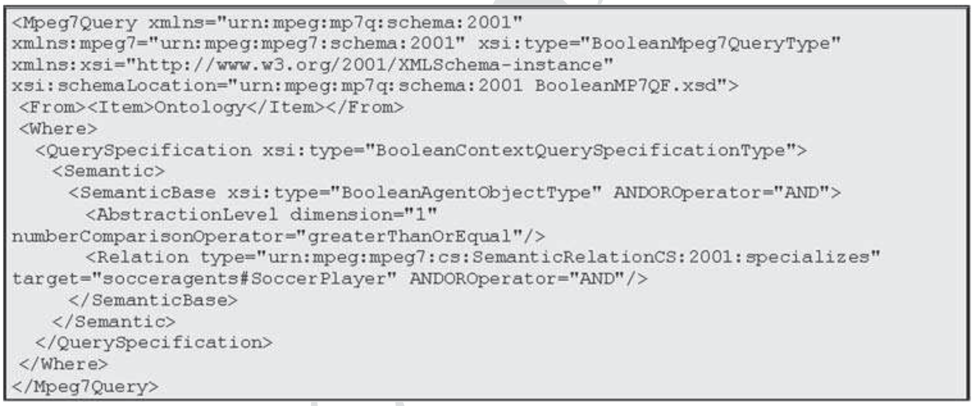
$$\mathit{BQS2} = \mathit{(From(Ontology)\ Where(BQS(SeP\ (AgentObjectType\ AND\ (specializes,\ SoccerPlayer)))))} \tag{18}$$

Fig. 5 MP7QL query stating "I want the multimedia objects where a goal is scored by Marques", using XML syntax



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="BooleanMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 BooleanMP7QF.xsd">
 <Where>
   <QuerySpecification xsi:type="BooleanContextQuerySpecificationType">
     <Semantic ANDOROperator="AND">
       <SemanticBase xsi:type="BooleanEventType" ANDOROperator="AND"
NOTOperator="false">
         <Relation ANDOROperator="AND" target="soccerevents#Goal"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
         <Relation ANDOROperator="AND" target="$mar"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"/>
       </SemanticBase>
       <SemanticBase ANDOROperator="AND" xsi:type="BooleanAgentObjectType" id="$mar"
NOTOperator="false">
         <Relation ANDOROperator="AND" target="socceragents#PlayerObject"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
         <Agent xsi:type="BooleanPersonType">
           <Name><FamilyName>Marques</FamilyName></Name>
         </Agent>
       </SemanticBase>
     </Semantic>
   </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 6 MP7QL query stating "I want the abstract semantic entities that represent the subclasses of the SoccerPlayer class", using XML syntax



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="BooleanMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 BooleanMP7QF.xsd">
 <From><Item>Ontology</Item></From>
 <Where>
   <QuerySpecification xsi:type="BooleanContextQuerySpecificationType">
     <Semantic>
       <SemanticBase xsi:type="BooleanAgentObjectType" ANDOROperator="AND">
         <AbstractionLevel dimension="1"
numberComparisonOperator="greaterThanOrEqual"/>
         <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:specializes"
target="socceragents#SoccerPlayer" ANDOROperator="AND"/>
       </SemanticBase>
     </Semantic>
   </QuerySpecification>
 </Where>
</Mpeg7Query>
```

It is a query on a soccer ontology that asks for the abstract semantic entities that represent the subclasses of the "SoccerPlayer" class.

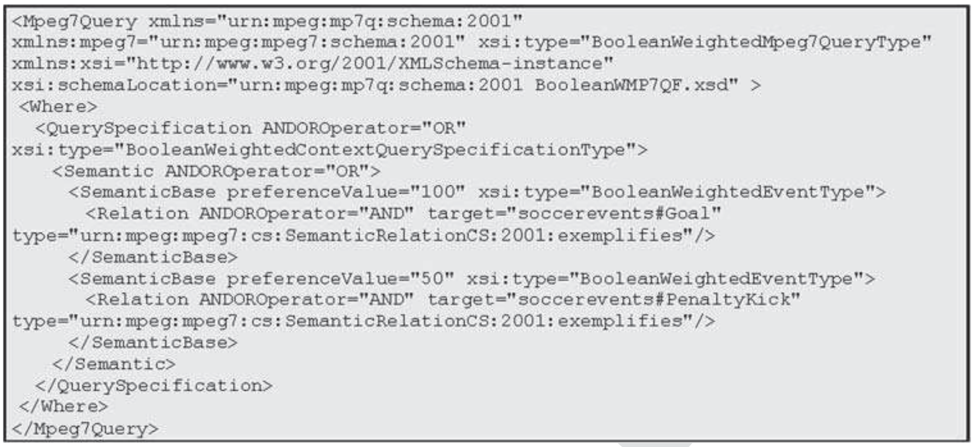
Query specifications with explicit boolean operators and preference values. In the query specifications with explicit boolean operators and preference values the boolean operator and the preference value that should be applied in the element contents may be explicitly specified for each query specification element. The query specifications with explicit boolean operators and preference values are represented by the subtypes of the *BooleanWeightedQuerySpecificationType*, as shown in Fig. 2. These are the *BooleanWeightedContextQuerySpecificationType* type represents queries for which a query context may be specified and the *BooleanWeightedFilteringAndSearchPreferencesType* type represents user FASPs with explicit boolean operators and preference values.

The *BooleanWeightedContextQuerySpecificationType* and the *BooleanWeightedFilteringAndSearchPreferencesType* query specifications (BWQS) are formally described using the regular expression syntax of (19). All the BWQS elements may have explicit boolean operators and preference values.

$$\begin{aligned} \mathit{BWQS} = {} & (\mathit{MI}|\mathit{MP}|\mathit{ML}|\mathit{SU}|\mathit{CrP}|\mathit{ClP}|\mathit{SoP}|\mathit{SeP}\mid \\ & \mathit{PC}|\mathit{UI}|\mathit{MH}|\mathit{PoV}|\mathit{RM}|\mathit{R}|\mathit{TA}|\mathit{DR}|\mathit{SER}|\mathit{VD}\mid \\ & \mathit{VDS}|\mathit{AD}|\mathit{ADS}|\mathit{VTSD}|\mathit{UH}|\mathit{FASP})\ \mathit{pv} \\ & ((\mathit{AND}|\mathit{OR}|\mathit{XOR})\ (\mathit{MI}|\mathit{MP}|\mathit{ML}|\mathit{SU}\mid \\ & \mathit{CrP}|\mathit{ClP}|\mathit{SoP}|\mathit{SeP}|\mathit{PC}|\mathit{UI}|\mathit{MH}|\mathit{PoV}|\mathit{RM}|\mathit{R}\mid \\ & \mathit{TA}|\mathit{DR}|\mathit{SER}|\mathit{VD}|\mathit{VDS}|\mathit{AD}|\mathit{ADS}\mid \\ & \mathit{VTSD}|\mathit{UH}|\mathit{FASP})\ \mathit{pv})^{*} \end{aligned} \tag{19}$$

A semantic preference element with explicit preference values and boolean operators *(BWSeP)* is formally described in (20).

$$\mathit{BWSeP} = \mathit{BWSE}\ \mathit{pv}\ ((\mathit{AND} \mid \mathit{OR} \mid \mathit{XOR})\ \mathit{BWSE}\ \mathit{pv})^{*} \tag{20}$$



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="BooleanWeightedMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 BooleanWMP7QF.xsd" >
 <Where>
  <QuerySpecification ANDOROperator="OR"
xsi:type="BooleanWeightedContextQuerySpecificationType">
    <Semantic ANDOROperator="OR">
      <SemanticBase preferenceValue="100" xsi:type="BooleanWeightedEventType">
        <Relation ANDOROperator="AND" target="soccerevents#Goal"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
      </SemanticBase>
      <SemanticBase preferenceValue="50" xsi:type="BooleanWeightedEventType">
        <Relation ANDOROperator="AND" target="soccerevents#PenaltyKick"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
      </SemanticBase>
    </Semantic>
  </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 7 MP7QL query stating "I want the multimedia objects where a goal is scored (preference 100) or a penalty kick takes place (preference 50)", using XML syntax

BWSE is a set of search and filtering conditions on semantic entities with explicit preference values and boolean operators, formally described in (21).

$$\begin{aligned} \mathit{BWSE} = {} & [\mathit{SID}]\ \mathit{SType}\ \mathit{pv}\ ((\mathit{AND} \mid \mathit{OR} \mid \mathit{XOR}) \\ & (\mathit{AName}\ \mathit{AValue}\ \mathit{pv}) \mid (\mathit{EName}\ \mathit{EValue} \\ & \mathit{pv}\ (\mathit{EAName}\ \mathit{EAValue}\ \mathit{pv})^*\ (E\ \mathit{pv})^*) \mid \\ & (\mathit{RType}\ \mathit{RTarget}\ \mathit{RSource}\ \mathit{RStrength}\ \mathit{pv}))^* \\ & \mathit{maxOccurs}\ \mathit{minOccurs} \end{aligned} \tag{21}$$

An MP7QL query specification with explicit boolean operators and preference values, is provided in the MP7QL query that states "I want the multimedia objects where a goal is scored (preference 100) or a penalty kick takes place (preference 50)", which is shown, expressed in formal syntax, in (22).

$$\begin{aligned} \mathit{BWQS1} = {} & (\mathit{Where}(\mathit{BQS}(\mathit{SeP}(\mathit{EventType}\ \mathit{AND} \\ & (\mathit{exemplifies},\ \mathit{Goal})\ 100)\ \mathit{OR} \\ & (\mathit{EventType}\ \mathit{AND}\ (\mathit{exemplifies}, \\ & \mathit{PenaltyKick})\ 50)))) \end{aligned} \tag{22}$$

We assume in this example that the abstract semantic entity "PenaltyKick" exists, which represents the class of all the penalty kicks. The same query, expressed using XML syntax, is shown in Fig. 7.

5 The output format of the MP7QL query language

In this section, we describe the output format of the MP7QL query language. The MP7QL query output format structures the query results in MPEG-7 descriptions. This allows the results of the MP7QL queries to be stored as new MPEG-7 descriptions, and to be recursively reused by the MP7QL query language. In addition, this feature satisfies an important query language design principle: it guarantees the closure property [5] of the MP7QL language.

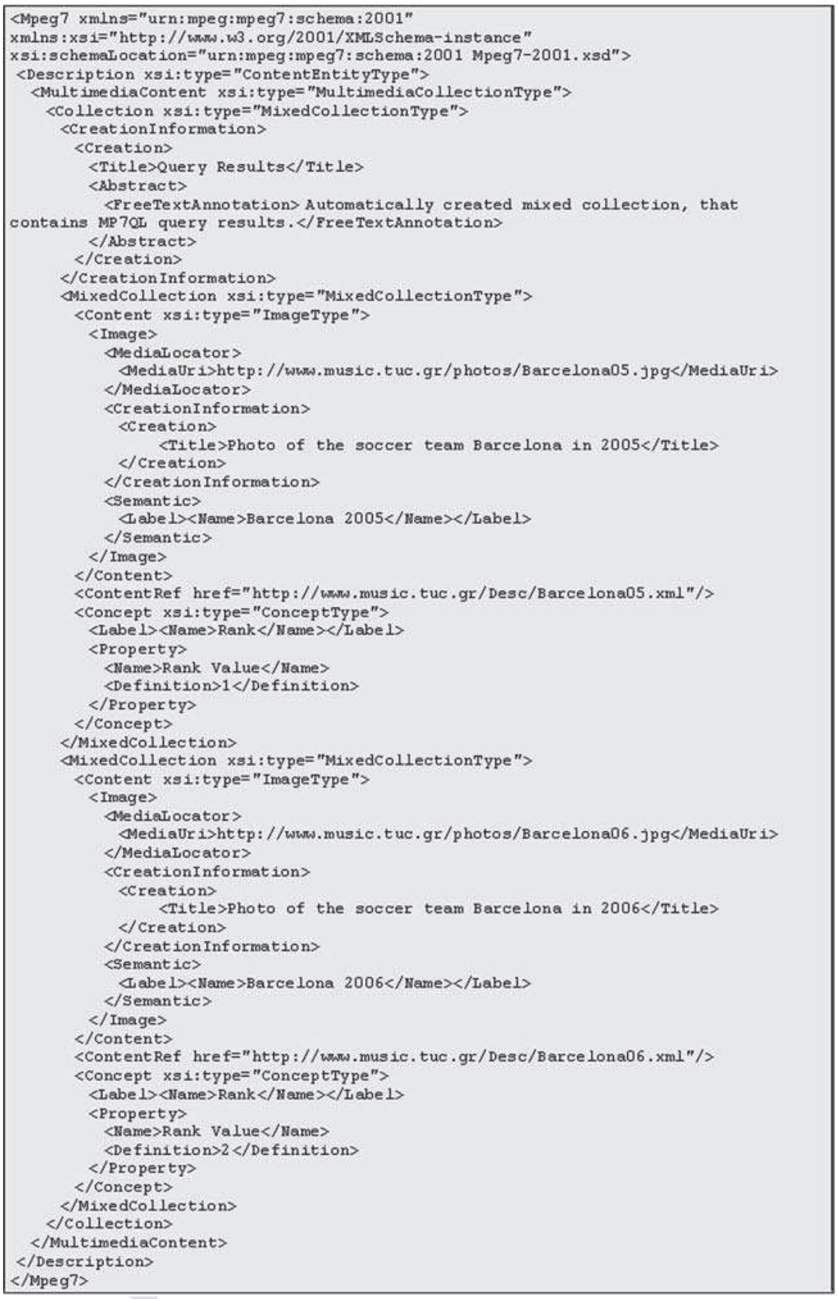
If the user wishes to view the query results structured in another way, he should specify the display format and the XSL stylesheet that should perform the transformation in his display device in the *Select* element of the MP7QL queries.

The MP7QL query output format organizes the query results in MPEG-7 descriptions where the query result sets are represented by MPEG-7 collections.

The collection that represents an MP7QL result set has a *CreationInformation* element, which contains information about the *creation* of the collection. If the query execution terminates normally, the collection title is "Query Results" and the abstract of the collection is "Automatically created mixed collection, that contains MP7QL query results". If an exception occurs during the execution of the query, no result items are returned, the collection title is "Exception" and the abstract of the collection contains the message returned by the query engine and describes the exception.

The result items returned by the MP7QL queries that terminate normally form an MPEG-7 *Mixed Collection*, ordered according to the ordering criteria provided by the user in the *OrderBy* element. If the user has specified a grouping criterion in the *GroupBy* element of the MP7QL query, every group of result items is represented by a mixed collection element that contains the group items ordered according to the ordering criteria. If no results are returned, an empty MPEG-7 collection is returned. Every result item is represented in the collection that represents the query results by a *MixedCollection* element comprised of:

- A set of *Concept* elements that represent the ranking, the relevance and any other information provided by the query engine about the current result item.
- A URI reference to the MPEG-7 description of the current result item, which is represented by a *ContentRef* element if the query is about multimedia content descriptions and by a *ConceptRef* element if the query is about reusable semantic entities or ontologies expressed in MPEG-7 syntax. This element is present for all the query result items, independently of the user's selections.



```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
 <Description xsi:type="ContentEntityType">
   <MultimediaContent xsi:type="MultimediaCollectionType">
     <Collection xsi:type="MixedCollectionType">
       <CreationInformation>
         <Creation>
           <Title>Query Results</Title>
           <Abstract>
             <FreeTextAnnotation> Automatically created mixed collection, that
contains MP7QL query results.</FreeTextAnnotation>
           </Abstract>
         </Creation>
       </CreationInformation>
       <MixedCollection xsi:type="MixedCollectionType">
         <Content xsi:type="ImageType">
           <Image>
             <MediaLocator>
               <MediaUri>http://www.music.tuc.gr/photos/Barcelona05.jpg</MediaUri>
             </MediaLocator>
             <CreationInformation>
               <Creation>
                    <Title>Photo of the soccer team Barcelona in 2005</Title>
               </Creation>
             </CreationInformation>
             <Semantic>
               <Label><Name>Barcelona 2005</Name></Label>
             </Semantic>
           </Image>
         </Content>
         <ContentRef href="http://www.music.tuc.gr/Desc/Barcelona05.xml"/>
         <Concept xsi:type="ConceptType">
           <Label><Name>Rank</Name></Label>
           <Property>
             <Name>Rank Value</Name>
             <Definition>1</Definition>
           </Property>
         </Concept>
       </MixedCollection>
       <MixedCollection xsi:type="MixedCollectionType">
         <Content xsi:type="ImageType">
           <Image>
             <MediaLocator>
               <MediaUri>http://www.music.tuc.gr/photos/Barcelona06.jpg</MediaUri>
             </MediaLocator>
             <CreationInformation>
               <Creation>
                    <Title>Photo of the soccer team Barcelona in 2006</Title>
               </Creation>
             </CreationInformation>
             <Semantic>
               <Label><Name>Barcelona 2006</Name></Label>
             </Semantic>
           </Image>
         </Content>
         <ContentRef href="http://www.music.tuc.gr/Desc/Barcelona06.xml"/>
         <Concept xsi:type="ConceptType">
           <Label><Name>Rank</Name></Label>
           <Property>
             <Name>Rank Value</Name>
             <Definition>2</Definition>
           </Property>
         </Concept>
       </MixedCollection>
     </Collection>
   </MultimediaContent>
 </Description>
</Mpeg7>
```

Fig. 8 The results of the query of Fig. 3

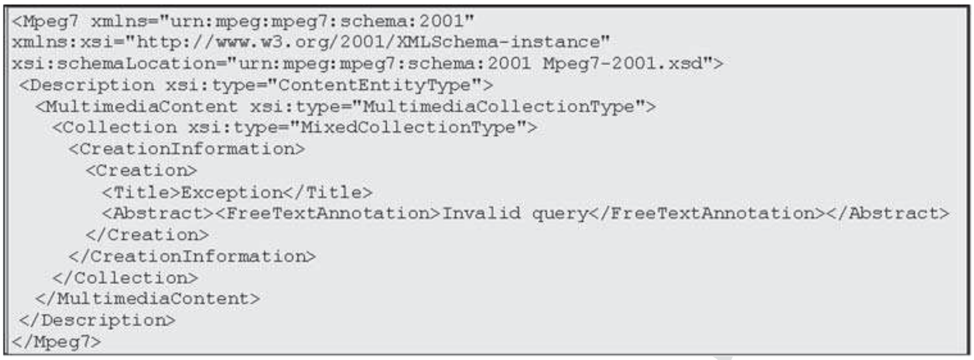
- The element that represents the elements of the MPEG-7 description of the current result item that were returned according to the user selections in the *Select* part of the query. This element is a *Content* element if the query is about multimedia content descriptions and a *Concept* element if the query is about reusable semantic entities or ontologies expressed in MPEG-7 syntax. This element is present only if the user has specified, in the *Select* part of the query, some elements and/or attributes of the results to be returned to him.

As an example, let the result set of the query of Fig. 3 to be a set of two images, showing the soccer team Barcelona in 2005 and in 2006. The query engine returns the rank of the result items. This result set should be structured, according to the MP7QL query output format, as shown in Fig. 8.

Consider now that an "Invalid query" exception occurs. In this case, the query output will be the MPEG-7 collection shown in Fig. 9.

6 The MP7QL Filtering and Search Preference Model

We present in this section the MP7QL *Filtering and Search Preference* (*FASP*) model. As already mentioned, the MP7QL FASPs essentially are MP7QL query specifications. Thus, a



```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
 <Description xsi:type="ContentEntityType">
   <MultimediaContent xsi:type="MultimediaCollectionType">
     <Collection xsi:type="MixedCollectionType">
       <CreationInformation>
         <Creation>
           <Title>Exception</Title>
           <Abstract><FreeTextAnnotation>Invalid query</FreeTextAnnotation></Abstract>
         </Creation>
       </CreationInformation>
     </Collection>
   </MultimediaContent>
 </Description>
</Mpeg7>
```

Fig. 9 MP7QL Output when the “Invalid query” Exception occurs

FASP may have all the elements of a query specification as well as *FilteringAndSearchPreferences* elements that allow forming FASP hierarchies.

According to the presentation of Sect. 4, the MP7QL FASPs are distinguished into FASPs with explicit preference values, which are represented by the *WeightedFilteringAndSearchPreferencesType (WFASP)* (expressed according to the formal syntax of (23)), FASPs with explicit boolean operators, which are represented by the *BooleanFilteringAndSearchPreferencesType* (expressed according to the formal syntax of (24)) and FASPs with explicit boolean operators and explicit preference values, which are represented by the *BooleanWeightedFilteringAndSearchPreferencesType* (expressed according to the formal syntax of (25)). These types extend the abstract query specification types (*WeightedQuerySpecificationType*, *BooleanQuerySpecificationType* and *BooleanWeightedQuerySpecificationType* respectively) as shown in Fig. 2. Notice that the NOT operator and the negative preference values allow the users to express their negative preferences (dislikes) in the FASPs.

$$\begin{aligned} WFASP = {} & ((MI|MP|ML|SU|CrP|ClP|SoP|SeP|PC\,| \\ & UI|MH|PoV|RM|R|TA|DR|SER|VD| \\ & VDS|AD|ADS|VTSD|FASP)\ pv)^{*} \end{aligned} \tag{23}$$

$$\begin{aligned} BQS = {} & (MI|MP|ML|SU|CrP|ClP|SoP|SeP|PC|UI\,| \\ & MH|PoV|RM|R|TA|DR|SER|VD|VDS|AD| \\ & ADS|VTSD|FASP)\ [NOT]\ (AND|OR|XOR) \\ & (MI|MP|ML|SU|CrP|ClP|SoP|SeP|PC|UI| \\ & MH|PoV|RM|R|TA|DR|SER|VD| \\ & VDS|AD|ADS|VTSD|FASP)[NOT])^{*} \end{aligned} \tag{24}$$

$$\begin{aligned} BWQS = {} & (MI|MP|ML|SU|CrP|ClP|SoP| \\ & SeP|PC|UI|MH|PoV|RM|R|TA| \\ & DR|SER|VD|VDS|AD|ADS|VTSD \\ & |FASP)\ pv\ ((AND|OR|XOR)\ (MI| \\ & MP|ML|SU|CrP|ClP|R|SoP|SeP| \\ & PC|UI|MH|PoV|\ RM|TA|DR|SER| \\ & VD|VDS|AD|ADS|VTSD|FASP)\ pv)^{*} \end{aligned} \tag{25}$$

The MP7QL FASP model enhances the FASP model we proposed in [26] for the extension of the MPEG-7 FASPs with semantic user preferences. The model of [26] has extended the MPEG-7 FASPs with semantic user preferences only, while the MP7QL FASP model allows one to express preferences for every aspect of the MPEG-7 multimedia object descriptions. In addition, the usage of boolean operators is more flexible in the MP7QL FASP model. The discussion shows that the FASP model we proposed in [26] is a special case of the MP7QL FASP model.

The MPEG-7 FASPs are also a special case of the MP7QL FASP model. In particular, an MPEG-7 FASP is also an MP7QL FASP of type WeightedFilteringAndSearchPreferencesType, which has only the *CreationPreferences*, *ClassificationPreferences*, *SourcePreferences*, *PreferenceCondition* and *FilteringAndSearchPreferences* elements.

Figure 10 illustrates graphically that all the elements of the MPEG-7 FASP Model and the FASP model proposed in [26] also exist in the MP7QL FASP model.

An MP7QL FASP with explicit preference values, which states “I want the multimedia objects where a goal is scored (preference 100) and their title contains the keyword ‘soccer’ (preference 90)”, is shown, expressed in formal syntax, in (26).

$$\begin{aligned} FASP1 = {} & (Where((EventType\ (exemplifies,\ Goal)) \\ & 100(Title\ \text{‘soccer’})\ 90)) \end{aligned} \tag{26}$$

The same FASP, expressed using XML syntax, is shown in Fig. 11.

The user filtering and search preferences can be either included in the user queries or referenced in them. MP7QL allows including references to the user filtering and search preferences and the usage history, which can be used as query context in the MP7QL queries. As an example, in the query of Fig. 12 the user’s filtering and search preferences (with preference value 100) and the usage history (with preference value 50) are used in an MP7QL query, which will return to the user the recommended material.

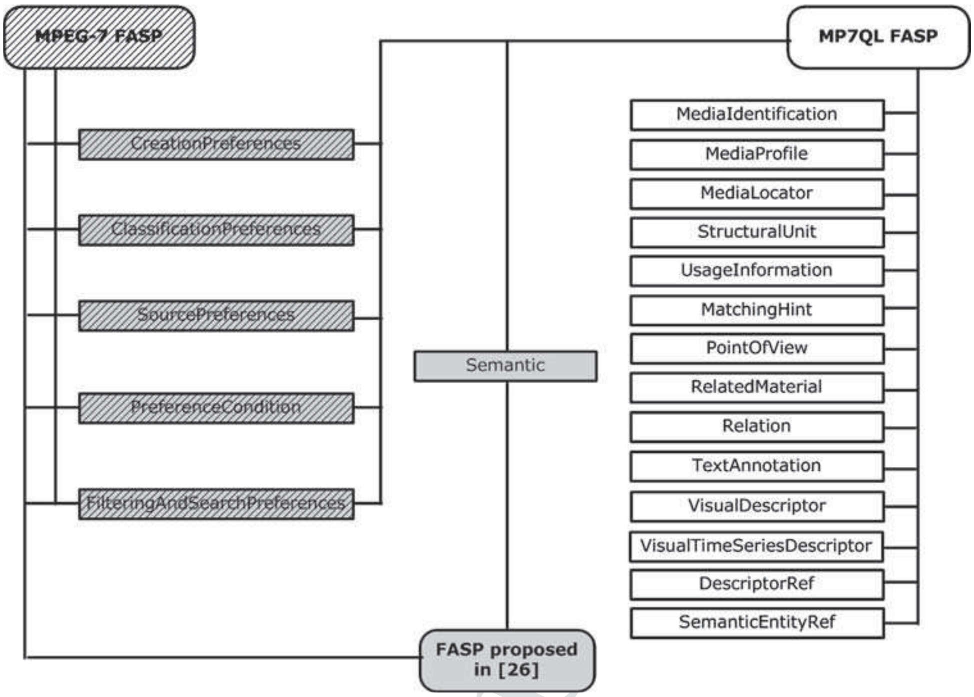
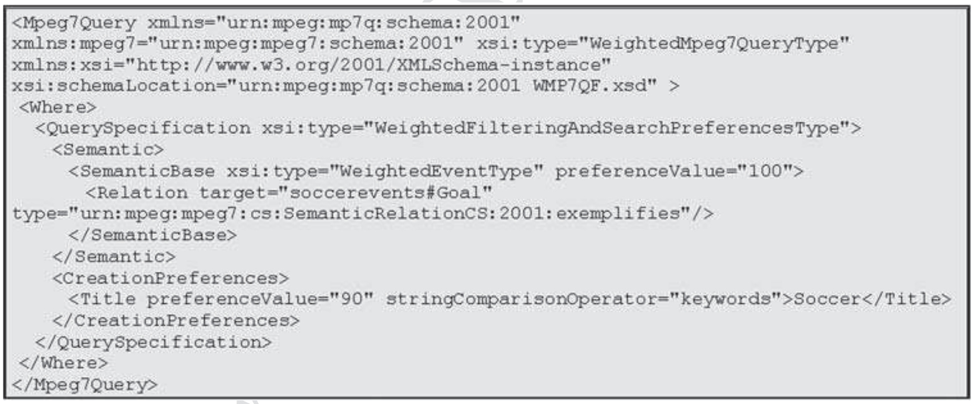
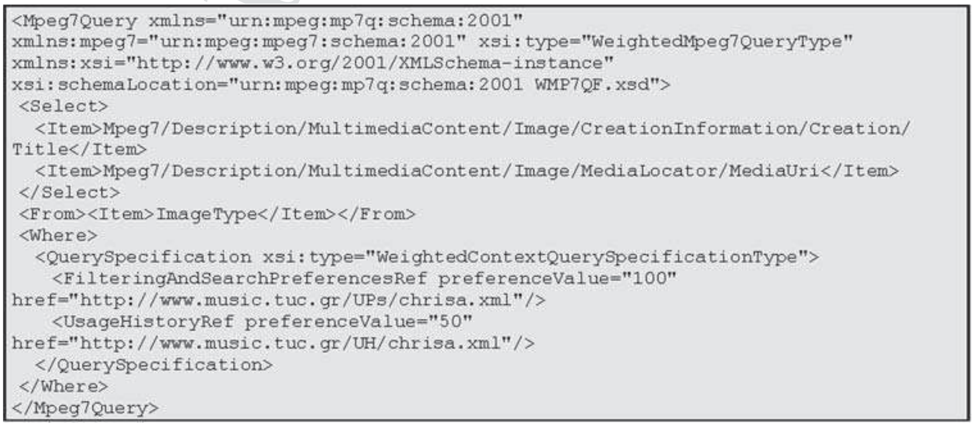


Fig. 10 The elements of the MPEG-7 FASP Model, the FASP model proposed in [26] and the MP7QL FASP model



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="WeightedMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 WMP7QF.xsd" >
 <Where>
   <QuerySpecification xsi:type="WeightedFilteringAndSearchPreferencesType">
     <Semantic>
       <SemanticBase xsi:type="WeightedEventType" preferenceValue="100">
         <Relation target="soccerevents#Goal"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
       </SemanticBase>
     </Semantic>
     <CreationPreferences>
       <Title preferenceValue="90" stringComparisonOperator="keywords">Soccer</Title>
     </CreationPreferences>
   </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 11 MP7QL FASP stating "I want the multimedia objects where a goal is scored (preference 100) and their title contains the keyword 'soccer' (preference 90)", using XML syntax



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="WeightedMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 WMP7QF.xsd">
 <Select>
   <Item>Mpeg7/Description/MultimediaContent/Image/CreationInformation/Creation/
Title</Item>
   <Item>Mpeg7/Description/MultimediaContent/Image/MediaLocator/MediaUri</Item>
 </Select>
 <From><Item>ImageType</Item></From>
 <Where>
   <QuerySpecification xsi:type="WeightedContextQuerySpecificationType">
     <FilteringAndSearchPreferencesRef preferenceValue="100"
href="http://www.music.tuc.gr/UPs/chrisa.xml"/>
     <UsageHistoryRef preferenceValue="50"
href="http://www.music.tuc.gr/UH/chrisa.xml"/>
   </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 12 MP7QL query that uses the User's FASP and Usage History

Fig. 13 The pure MPEG-7 working environment

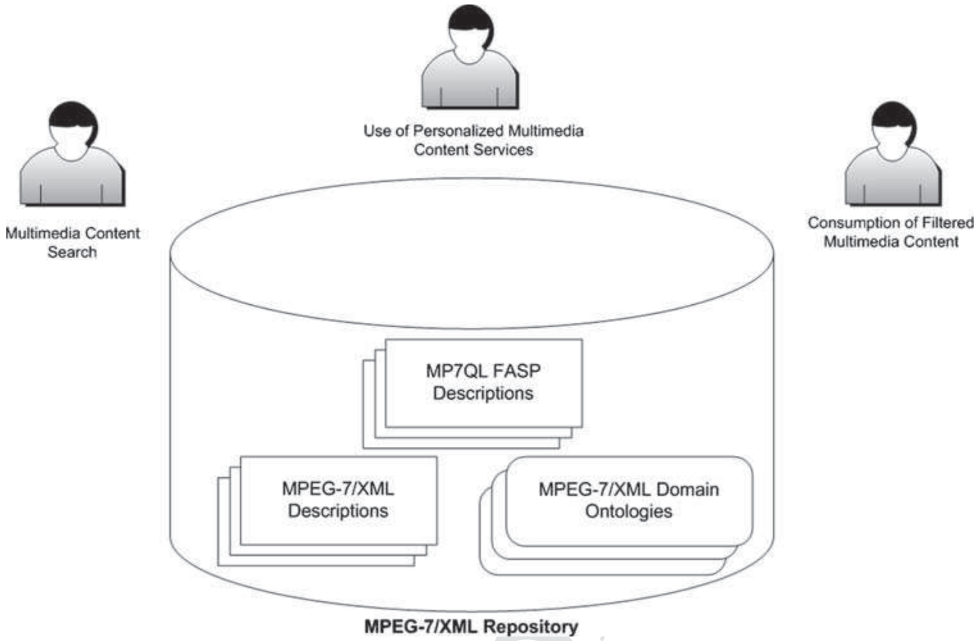
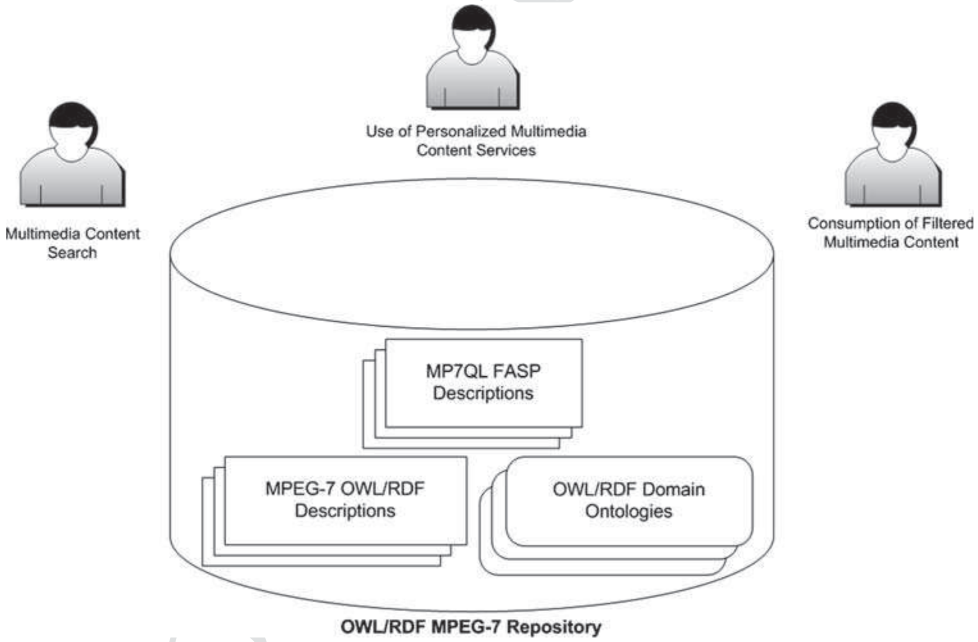

Use of Personalized Multimedia Content Services
Multimedia Content Search
Consumption of Filtered Multimedia Content
MP7QL FASP Descriptions
MPEG-7/XML Descriptions
MPEG-7/XML Domain Ontologies
MPEG-7/XML Repository

Fig. 14 The semantic MPEG-7 working environment


Use of Personalized Multimedia Content Services
Multimedia Content Search
Consumption of Filtered Multimedia Content
MP7QL FASP Descriptions
MPEG-7 OWL/RDF Descriptions
OWL/RDF Domain Ontologies
OWL/RDF MPEG-7 Repository

7 Usage scenarios and effectiveness evaluation

We present in this section how the MP7QL query language and its compliant FASP model may provide semantic retrieval, filtering and personalization in different working environments. We also present the results a real world evaluation study of the effectiveness of the MP7QL, in terms of expressive power.

In particular, we present the MP7QL usage scenarios in Sect. 7.1 and the MP7QL evaluation in Sect. 7.2.

7.1 MP7QL usage scenarios

In this subsection, we describe how the MP7QL and its compliant FASP model can be used in different working environments in order to provide uniform and transparent multimedia content retrieval and filtering as well as multimedia content service personalization.

The working environments we will study are the *Pure MPEG-7 Working Environment* shown in Fig. 13 and the *Semantic MPEG-7 Working Environment* shown in Fig. 14.

In the pure MPEG-7 working environment, an MPEG-7 metadata repository developed on top of a native XML database stores MPEG-7 descriptions expressed in XML syntax. These descriptions include MPEG-7 multimedia content descriptions, user preference descriptions and domain ontologies expressed using MPEG-7 constructs. The domain ontologies and the reusable semantic entities contained in the multimedia content descriptions may be utilized during the query definition and the user preference specification. The end-users access the repository in order to use personalized multimedia content services, to search for multimedia content that interests them and in order to filter the multimedia content they are going to consume. In such an environment, the XML Schema based syntax of the MP7QL can be used in order to express the user queries and the MP7QL FASP model (expressed in XML Schema) can be used for the expression of the users' filtering and search preferences for multimedia content filtering and service personalization. The query engine performing the MP7QL query evaluation and the MP7QL user preference matching can be built on top of the XQuery language that allows accessing the repository. This way, the users will be offered a uniform and transparent system for accessing the MPEG-7 descriptions. The implementation of a query engine for the XML Schema based syntax of the MP7QL on top of a native XML database is in progress in the context of the DS-MIRF framework and is being used by an ontology-based natural language interface generator [17] that allows locating multimedia information of interest using natural language queries.

In the semantic MPEG-7 working environment, an MPEG-7 metadata repository developed on top of an OWL/RDF repository stores MPEG-7 descriptions expressed in OWL/RDF syntax, essentially being individuals of the classes of an OWL ontology that captures the semantics of the MPEG-7 [28]. These descriptions include MPEG-7 multimedia content descriptions, user preference descriptions and domain ontologies. The end-users are provided with the same functionality offered by the pure MPEG-7 working environment. In such an environment, the OWL/RDF syntax of the MP7QL can be used in order to express the user queries and the MP7QL FASP model (also expressed in OWL/RDF syntax) can be used for the expression of the users' filtering and search preferences for multimedia content filtering and service personalization. The query engine that will perform the MP7QL query evaluation and the MP7QL user preference matching can be built on top of the SPARQL language [21] that allows accessing the repository.

7.2 Application effectiveness evaluation

We present in this subsection the evaluation of the MP7QL in terms of expressive power. During the evaluation, we first checked the MP7QL against the general-purpose ISO MPEG-7 Query Format Requirements [15], which range from the support of "Query by Example" to the support of spatio-temporal queries. We found that the MP7QL covers these requirements. The ISO MPEG-7 Query Format Requirements are listed in Table 7, which also describes in brief how they are met by the MP7QL and indicates if they are met by the MPEG-7 FASPs. In addition to the functionality that the ISO MPEG-7 Query Format requires, the MP7QL provides functionality for supporting semantic queries using domain ontologies, and it has also the important in our opinion characteristic that the output format of MP7QL is MPEG-7, thus providing closure in the language [5].

We then evaluated the MP7QL against a real world application environment. We describe next the evaluation study that we performed, in order to allow the reader to understand more clearly the motivation and the applications of both the MP7QL query language and its compliant FASP model.

We have chosen to look in depth and to test extensively the benefits and the expressive power of the language in a domain-specific complete application. We selected the soccer domain, for which we have developed in the past, in the context of the DS-MIRF framework [28], a very detailed soccer ontology which has also been expressed using the MPEG-7 syntax. Next we specified a set of representative query types (not concrete queries). In order to do that, we used the following procedure: first, we visited the website of a popular betting company[2] and studied the complete set of bets available for the soccer games. Presumably the bets in such a site express also the summary interests of a very wide class of users which are soccer funs. Then, we transformed the statistics-oriented bet expressions used in the website of the betting company to content-oriented queries. For example, the bet expression *How many goals have been scored by Team X?* has been transformed to the query *Give me the multimedia objects showing the goals scored by Team X*, the bet expression *How many red cards will be given in Game Y?* has been transformed to the query *Give me the multimedia objects showing the red cards given in Game Y*, the bet expression *How many yellow cards will Player Z receive in Tournament W?* has been transformed to the query *Give me the multimedia objects showing the yellow cards of Player Z in Tournament W* etc. This way, we collected more than 100 query types for soccer games.[3]

All these query types were found to be capable to be expressed using the MP7QL and the conditions stated in them can also be incorporated in MP7QL FASPs without any changes or additions in the soccer ontology that we had described in [28] or changes in the MP7QL language. That means that the queries in MP7QL were capable to express

[2] The betting company is *BetandWin*, http://www.betandwin.com.

[3] The query types are available at: http://www.music.tuc.gr/delos/resources/SoccerQueryTypes.zip.

Table 7 The ISO MPEG-7 Query Format Requirements and their coverage by the MP7QL

ISO MPEG-7 query format requirement	MP7QL support	MPEG-7 FASP support
Integration with MPEG-7	The MP7QL input query format is well integrated with the MPEG-7 MDS, Visual and Audio parts, as it allows querying every aspect of an MPEG-7 description that is based on these parts. The MP7QL query output format is well integrated with the MPEG-7 MDS, Visual and Audio parts, as it is structured as a standard MPEG-7 mixed collection.	Yes
XML Technology	The MP7QL can be expressed both in XML Schema and OWL syntax, which are both XML-based. This way, the MP7QL queries are XML documents.	Yes
Language and Character Set Independence	The MP7QL query results are structured as standard MPEG-7 mixed collections. This way, the mechanisms provided by the MPEG-7 for language and character set independence are also used in the MP7QL query results. The same mechanisms have also been adopted in the MP7QL input query format, so that the textual parts of the queries can be expressed using any language and character set.	Yes
"Query-by-Textual Description"	The MP7QL allows queries that contain conditions on all the textual elements and attributes of the MPEG-7 descriptions. The string comparison operator provided by the MP7QL allows the users to specify how the textual values of the queries should be compared with the corresponding values of the MPEG-7 descriptions.	No
Free Text Queries	The *Keyword* element of the *CreationPreferences* allows specifying the keywords and/or keyword phrases they want to search for.	Yes
"Query By Example"	The *MediaURI* element of the media locator of the *Where* element of the MP7QL queries allows specifying the URI of the multimedia object that will be used as an example.	No
"Query By Segment Example"	The *DescriptionRef* element of the *Where* element of the MP7QL queries allows giving MPEG-7 segment descriptions as query examples.	No
"Query By Mixed Example"	Capability of defining multiple instances of the *MediaURI* element of the media locator in the *Where* element of the MP7QL queries.	No
"Query By ID"	The *EntityIdentifier* element of the media identification of the *Where* element of the MP7QL queries allows specifying the unique identifier of the multimedia object.	No
"Query by MPEG-7 Description"	The *DescriptionRef* element of the *Where* element of the MP7QL queries allows referencing the MPEG-7 descriptions that should be used as query examples.	No
Queries based on User Preferences and/or Usage History	The MP7QL allows including the user FASPs in the MP7QL queries. In addition, it allows including references to the user filtering and search preferences and the usage history, which can be used as query context.	Partial (Queries based on FASP only)
Combination of Query Conditions	The MP7QL provides several query elements that can be used in the same query and allow the specification of conditions on different aspects of the multimedia descriptions. In addition, the availability of boolean operators and preference values allows the combination of the different conditions according to the user's intentions. The string and number comparison operators further enhance the accuracy of the conditions specified in the MP7QL queries.	Partial (only on the MPEG-7 elements included in the MPEG-7 FASPs)
Empty Queries	The MP7QL supports empty queries, as all the MP7QL query elements are optional.	Yes
Use of Personal Information during Query Execution	The user information is made available in the MP7QL queries only if the user includes or references it in the queries. This way, the user explicitly states if he wishes his personal information to be used during query execution. In addition, the users may state in the *PreferenceConditions* element of their FASPs under which condition the FASP should be taken into account. This way, the conditions (in terms of location and time) are always checked before the filtering and search preferences of the users are utilized during the query execution.	Yes
Spatiotemporal Queries	Support of querying on the spatial and the temporal relationships provided by the MPEG-7.	No
Specification of the Information contained in the Result Set	The *Select* element of the MP7QL queries allows specifying which part(s) of the MPEG-7 descriptions should be present in the query results.	No
Specification of the Media Formats/Types of the Result Set	The *From* element of the MP7QL queries allows specifying the media type of the multimedia objects they are looking for. The *MediaFormat* element of the source preferences of the *Where* element of the MP7QL queries allows specifying the media format of the multimedia objects they are looking for.	No

Table 7 Contined

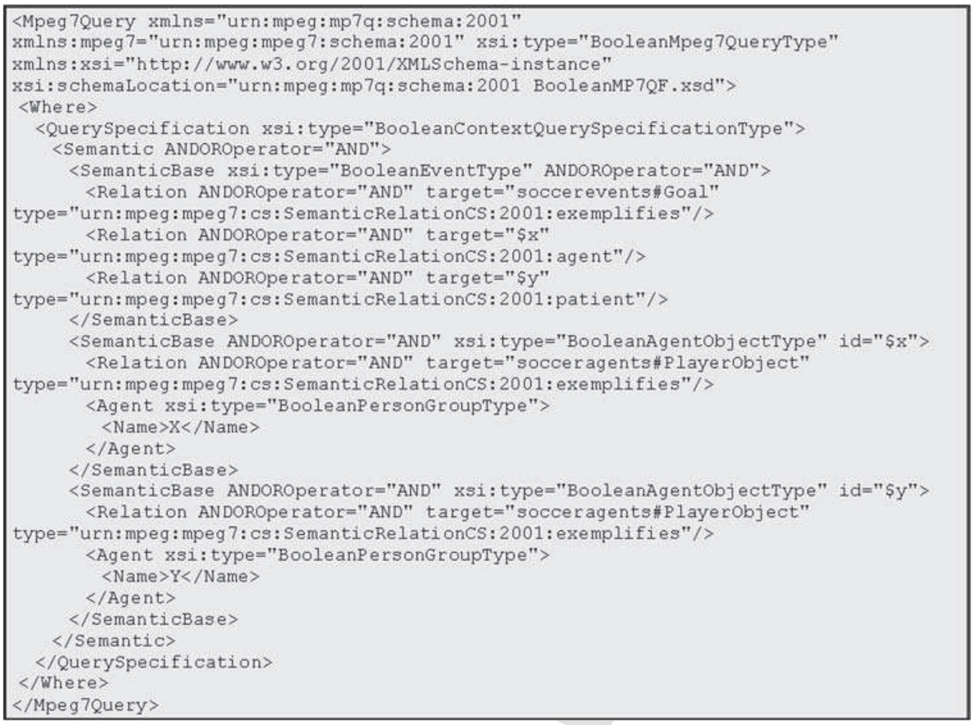
ISO MPEG-7 query format requirement	MP7QL support	MPEG-7 FASP support
Sorting and Grouping Parameters for the Result Set	The *OrderBy* element of the MP7QL queries allows specifying the criteria for ordering the query results. The *GroupBy* element of the MP7QL queries allows specifying the criterion for grouping the query results.	No
Specification of the Display Format of the Result Set	The MP7QL allows the users to specify the result set structure by providing: (a) A *display format* structure in the *format* attribute of the *Select* element of the MP7QL queries; and (b) The *rules* for transforming the MP7QL output format in the display format, in the *transformationRules* attribute of the *Select* element of the MP7QL queries.	No
Limiting the Size of the Result Set	The *maxItems* attribute of the *Select* element of the MP7QL queries allows specifying the maximum number of the query result items they wish to receive.	No
Paging the Result Set	The *Select* element of the MP7QL queries allows specifying which of the result pages should be returned first to them.	No
Default Format of the Result Set	The default format has the form of a standard MPEG-7 mixed collection that contains references to the MPEG-7 descriptions of the query results.	No
Specification of the Exceptions	If an exception occurs, the result returned is an empty MPEG-7 collection, which has "Exception" as title and the description of the exception as abstract.	No
Server/Service Provider Selection	The *Disseminator* element of the source preferences of the Where element of the MP7QL queries allows specifying which should be the server/service provider to which the query should be sent.	Yes
Relevance Feedback Support	The capability to define MP7QL queries that use existing MPEG-7 descriptions as examples allows providing relevance feedback support. Thus, the users may select the query results they prefer and form a new query with these descriptions. They may also assign preference values to the descriptions, in order to distinguish the result items that satisfy them more.	No
Support for Searching within a Result Set	A query result set formed according to the MP7QL query output format is an MPEG-7 description and can be stored and recursively queried.	No
Provision of Time Limit to the Query Response	The *timeLimit* attribute of the *Select* element of the MP7QL queries allows specifying the time limit for receiving the query results.	No

exactly the semantic meaning of the natural language queries of the bets, and as result there were no false drops in any of the queries. There were no queries in the set of the 100 query types (bet expressions) that gave us a particular difficulty in expressing them. The amount of difficulty was not drastically different among the query types. In particular, these query types can be expressed as query templates that use semantic entities to describe the desired content. In event-based environments like the soccer (and the sports in general) the query conditions expressed by users are about the events, including the agents participating in them, the time they occur, etc. For example, the query type *Give me the multimedia objects showing the goals scored by Team X against Team Y* is expressed in formal MP7QL syntax as shown in (27) and in XML syntax as shown in Fig. 15.

$$\begin{aligned}
\mathit{BQS3} = {} & (\textit{Where(BQS(SeP(EventType AND}\\
& \textit{(exemplifies, Goal) AND (agent,}\\
& \textit{\$ x) AND (patient, \$ y))}\\
& \textit{AND ((\$ x, AgentObjectType) AND}\\
& \textit{(exemplifies, SoccerTeam, \$ x) AND}\\
& \textit{(Agent(Name 'X')) AND}\\
& \textit{((\$ y, AgentObjectType)}\\
& \textit{AND (exemplifies, SoccerTeam, \$ y)}\\
& \textit{AND (Agent(Name 'Y')))))}
\end{aligned} \tag{27}$$

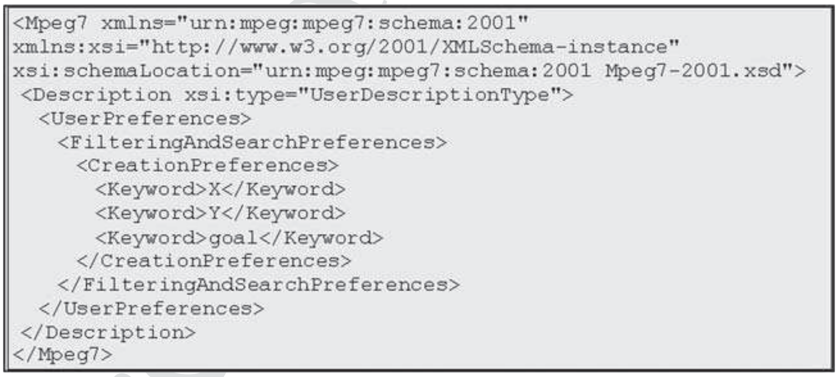
We assume in this example that the abstract semantic entities "SoccerTeam" and "Goal" exist, which represent the classes of all the soccer teams and all the goals, respectively. We also assume that the soccer teams X and Y are bound to the "$x" and "$y" variables, respectively.

Some of the collected query types may be well expressed using the keyword-based approach of the MPEG-7 FASPs. For example, the query type *Give me the multimedia objects showing the goals scored between Team X and Team Y* should be expressed in an MPEG-7 FASP using the keywords "X", "Y" and "goal" as shown in Fig. 16 and would return all the goals between the two teams. The MPEG-7 FASPs are limiting when a directed relationship exists between the participants of the event. As an example, consider the query type *Give me the multimedia objects showing the goals scored by Team X against Team Y*. This would be expressed using again the keywords "X", "Y" and "goal" as shown in Fig. 16 and would return both the goals scored by team X against team Y and the goals scored by team Y against team X, thus



```
<Mpeg7Query xmlns="urn:mpeg:mp7q:schema:2001"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:type="BooleanMpeg7QueryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mp7q:schema:2001 BooleanMP7QF.xsd">
 <Where>
  <QuerySpecification xsi:type="BooleanContextQuerySpecificationType">
   <Semantic ANDOROperator="AND">
    <SemanticBase xsi:type="BooleanEventType" ANDOROperator="AND">
     <Relation ANDOROperator="AND" target="soccerevents#Goal"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
     <Relation ANDOROperator="AND" target="$x"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"/>
     <Relation ANDOROperator="AND" target="$y"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patient"/>
    </SemanticBase>
    <SemanticBase ANDOROperator="AND" xsi:type="BooleanAgentObjectType" id="$x">
     <Relation ANDOROperator="AND" target="socceragents#PlayerObject"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
     <Agent xsi:type="BooleanPersonGroupType">
      <Name>X</Name>
     </Agent>
    </SemanticBase>
    <SemanticBase ANDOROperator="AND" xsi:type="BooleanAgentObjectType" id="$y">
     <Relation ANDOROperator="AND" target="socceragents#PlayerObject"
type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:exemplifies"/>
     <Agent xsi:type="BooleanPersonGroupType">
      <Name>Y</Name>
     </Agent>
    </SemanticBase>
   </Semantic>
  </QuerySpecification>
 </Where>
</Mpeg7Query>
```

Fig. 15 MP7QL query stating "I want the multimedia objects where a goal is scored by Team X against Team Y", using XML syntax



```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
 <Description xsi:type="UserDescriptionType">
  <UserPreferences>
   <FilteringAndSearchPreferences>
    <CreationPreferences>
     <Keyword>X</Keyword>
     <Keyword>Y</Keyword>
     <Keyword>goal</Keyword>
    </CreationPreferences>
   </FilteringAndSearchPreferences>
  </UserPreferences>
 </Description>
</Mpeg7>
```

Fig. 16 MPEG-7 FASP stating "I want the multimedia objects where"

resulting in false drops. Other examples of query types that cannot be expressed using the MPEG-7 FASPs are *Give me the multimedia objects showing the red cards of the players of Team X in the game Team X–Team Y*, *Give me the multimedia objects showing the goals scored by Team Y from a penalty kick against Team X* etc. All these query types can be expressed in MP7QL.

We plan to perform in the future additional extensive evaluation of the effectiveness of the MP7QL in other application domains, including news and cultural heritage.

8 Conclusions—future work

We have presented in this paper the MPEG-7 Query Language (MP7QL), a powerful query language that we have developed for querying MPEG-7 descriptions The MP7QL has the MPEG-7 as data model and allows for querying every aspect of an MPEG-7 multimedia content description, including semantics, low-level visual features and media-related aspects. It also allows for the exploitation of domain knowledge encoded using pure MPEG-7 constructs. In addition, it allows the explicit specification of boolean operators and/or preference values. The MP7QL query results are represented as MPEG-7 documents, guaranteeing the closure of the results within the MPEG-7 space.

The MP7QL queries may utilize the user Filtering and Search Preferences (FASP) and Usage History as context, thus allowing for personalized multimedia content retrieval. The MP7QL FASP model has the model of the standard MPEG-7 FASPs as a special case

Both the MP7QL and its compatible FASP model allow for the exploitation of domain knowledge encoded using pure MPEG-7 constructs. In addition, they allow the explicit specification of boolean operators and/or preference values in order to allow the combination of the query conditions according to the user intentions and the expression of the importance of the individual conditions for the users.

The MP7QL has been expressed using both XML Schema and OWL syntax. An implementation of the MP7QL, on top of an XML Native Database is currently in progress.

Our future research in this area includes the specification of the MP7QL query server capabilities description, using a profiling mechanism similar to the one that is available for MPEG-7 [16]. In addition, the XML syntax of the MP7QL will be provided in the binary format of XML (BiM) [14] in order to support bandwidth-efficiency, in the same way that the MPEG-7 descriptors and description schemes are expressed in BiM. Further experimentation will take place for the evaluation of the MP7QL and the MP7QL FASP model.

References

1. Agius, H., Angelides, M.: Modelling and filtering of MPEG-7-compliant meta-data for digital video. In: Proceedings of the ACM SAC, 2004, pp. 1248–1252 (2004)
2. Agius, H., Angelides, M.: Enriching MPEG-7 user models with content metadata. In: Proceedings of SMAP 2006, pp. 151–156 (2006)
3. Akrivas, G., Ioannou, S., Karakoulakis, E., Karpouzis, K., Avrithis, Y., Delopoulos, A., Kollias, S., Varlamis, I., Vaziriannis, M.: An intelligent system for retrieval and mining of audiovisual material based on the MPEG-7 description schemes. In: Proceedings of the Eunite 2001, Tenerife, Spain (2001)
4. Bertini, M., Del Bimbo, A., Nunziati, W.: Video clip matching using MPEG-7 descriptors and edit distance. In: Proceedings of the CIVR 2006, pp. 133–142 (2006)
5. Date, C.J.: An introduction to database systems, 6th edn. Addison-Wesley, New York (1995)
6. Eidenberger, H., Breiteneder, C.: VizIR: a framework for visual information retrieval. J. Vis. Lang. Comput. **14**(5), 443–469 (2003)
7. Fallside, D., Walmsley, P. (eds.): XML Schema Part 0: Primer. W3C Recommendation. http://www.w3.org/TR/xmlschema-0/ (2001)
8. Graves, A., Lalmas, M.: Video retrieval using an MPEG-7 based inference network. In: Proceedings of the ACM SIGIR 2002, pp. 339–346 (2002)
9. Hammiche, S., Lopez, B., Benbernou, S., Hacid, M.S., Vakali, A.: Domain knowledge based queries for multimedia data retrieval. In: Proceedings of the Workshop on Multimedia Semantics 2006 (WMS 2006), pp. 94–103, 19–21 June 2006, Chania, Crete (2006)
10. Hunter, J.: Adding multimedia to the semantic Web—building an MPEG-7 ontology. In: Proceedings of the International Semantic Web Working Symposium (SWWS) (2001)
11. ISO/IEC: 15938–3:2001: Information Technology—Multimedia content description interface. Part 3 Visual, Version 1 (2001)
12. ISO/IEC: 15938–4:2001: Information Technology—Multimedia content description interface. Part 4 Audio, Version 1 (2001)
13. ISO/IEC: 15938–5:2003: Information Technology—Multimedia content description interface. Part 5: Multimedia description schemes, 1st edn (2003)
14. ISO/IEC 23001–1: Information technology—MPEG system technologies. Part 1: Binary MPEG Format for XML
15. ISO/IEC JTC1/SC29/WG11. N8219—MPEG-7 Query Format Requirements version 1.1. Klagenfurt, Austria (2006)
16. ISO/IEC JTC1/SC29/WG11. N6263—Study of MPEG-7 Profiles Part 9 Committee Draft (2003)
17. Karanastasi, A., Christodoulakis, S.: An ontology-based natural language interface generator for multimedia repositories. DELOS Conference, Tirrenia, Italy, March 2007 (2007)
18. Kay, M. (ed.): XSL Transformations (XSLT) Version 2.0. W3C Recommendation, 23 January 2007. http://www.w3.org/TR/xslt20/ (2005)
19. Lee, M.H., Kang, J.H., Myaeng, S.H., Hyun, S.J., Yoo, J.M., Ko, E.J., Jang, J.W., Lim, J.H.: A Multimedia Digital Library System based on MPEG-7 and XQuery. In: Proceedings of the ICADL (2003)
20. McGuinness, D.L., van Harmelen, F. (eds.): OWL Web Ontology Language: overview. W3C Recommendation. 10 Feb. 2004. http://www.w3.org/TR/owl-features (2004)
21. Prud'hommeaux, E., Seaborne, A. (eds.): SPARQL Query Language for RDF. W3C Draft, 26 March 2007. http://www.w3.org/TR/rdf-sparql-query/ (2007)
22. Rogers, D., Hunter, J., Kosovic, D.: The TV-Trawler Project. Special Issue of the International Journal of Imaging Systems and Technology on Multimedia Content Description and Video Compression (2003)
23. Siméon, J., Chamberlin, D., Fernández, M., Boag, S., Florescu, D., Robie, J. (eds): XQuery 1.0: an XML Query Language. W3C Recommendation, 23 Jan. 2007. http://www.w3.org/TR/xquery/ (2007)
24. Salembier, P.: MPEG-7 Multimedia description schemes. IEEE TA Circuits Syst. Video Technol. **11**(6), 748–759 (2001)
25. Tseng, B., Lin, C.Y., Smith, J.: Using MPEG-7 and MPEG-21 for personalizing video. IEEE Multimed. **11**(1), 42–52 (2004)
26. Tsinaraki, C., Christodoulakis, S.: A Multimedia user preference model that supports semantics and its application to MPEG 7/21. In: Proceedings of the Multimedia Modeling 2006 Conference (MMM 2006), pp. 35–42, January 2006, Beijing, China (2006)
27. Tsinaraki, C., Fatourou, E., Christodoulakis, S.: An ontology-driven framework for the management of semantic metadata describing audiovisual information. In: Proceedings of the Conference of Advanced Information Systems Engineering (CaiSE), Velden, Austria, 2003, pp. 340–356 (2003)
28. Tsinaraki, C., Polydoros, P., Christodoulakis, S.: Interoperability support between MPEG-7/21 and OWL in DS-MIRF. Transactions on Knowledge and Data Engineering (TKDE), Special Issue on the Semantic Web Era, 2007, pp. 219–232 (2007)
29. Vallet, D., Fernandez, M., Castellis, P., Mylonas, P., Avrithis, Y.: A contextual personalization approach based on ontological knowledge. In: Proceedings the IJCAI 2006 (2006)
30. Wang, Q., Balke, W.T., Kiessling, W., Huhn, A.: P-News: Deeply Personalized News Dissemination for MPEG-7 based Digital Libraries. In: Proceedings of ECDL 2004 (2004)