MP7QF: An MPEG-7 Query Format

Matthias Gruhne
Fraunhofer IDMT
Metadata Department
Langewiesener Str. 22, 98693 Ilmenau, Germany
{ghe}@idmt.fhg.de

Ruben Tous and Jaime Delgado
Universitat Politècnica de Catalunya (UPC)
Dpt. d'Arquitectura de Computadors
Jordi Girona, 1-3. E-08034 Barcelona, Spain
{rtous, jaime.delgado}@ac.upc.edu

Mario Doeller and Harald Kosch
Dpt. Distributed Information Technology
University of Passau
Innstrasse 43, Passau, Germany
{mario.doeller, harald.kosch}@uni-passau.de

Abstract

Due to the growing number of digital audiovisual media during the last years, the amount of metadata to describe the content increased significantly. In order to improve the search and retrieve of such data, the metadata standard MPEG-7 has been established. This international standard facilitates many application domains and is probably the richest multimedia metadata set available today. However, it does not yet exist a common query format, which communicates between client and databases and which supports crossmodal and cross-medial retrieval. Therefore the MPEG committee decided to contribute to this issue. This paper describes the current status of the standardization effort and shows the already implemented technology more in detail. Furthermore, examples are described in order to highlight the concept of the proposed query language.

1. Introduction

The ever-growing availability of digital audiovisual material to the end user via new media and content distribution methods resulted in an increasing need to automatically categorize digital media. Descriptive information about digital data which is delivered together with the actual content represents one way to facilitate this search immensely. Due to the fact, that these information are mostly disordered and a search turned out to be very challenging, the metadata standard MPEG-7 has been established. One of the missing parts is a query format that describes an interface between user and database to easily distribute a combined crossmodal and crossmedia query to a database. Therefore the MPEG committee started to close the missing gap. The current status of the standardization efforts is committee draft (CD), meaning that all descriptors explained in this paper are stable and might be part of the future standard. However, some technology might be added after careful selection, testing and approval. The remainder of this paper is organized as follows: Section 2 describes the existing query language approaches and multimedia databases. Then, Section 3 explains the actual query format core in the CD and depicts the input format and the output format more in detail. Section 4 shows an example in order to exhibit the concept of the query language. The document finishes with a conclusion.

2. Related work

A number query languages for multimedia data have been published, e.g. SQL/MM [9] or MOQL [6] which are out of scope of this paper because they have limits to handle XML data. Today, these kinds of works are based on MPEG-7 descriptors and the MPEG-7 data model. Some simply defend the use of XQuery or some extensions. Others define a more high-level and user-oriented approach. With respect to the previous works, the language presented here is based on MPEG-7. It outperforms XQuery-based approaches like [5],[10],[2],[13],[3] because, while offer-

0-7695-3030-3/07 $25.00 © 2007 IEEE
DOI 10.1109/AXMEDIS.2007.40

ing the same level of expressiveness, it offers multiple content-based search functionalities (query-by-example (QBE) or query-by-freetext) and other information retrieval (IR) -like features (e.g. paging or relevance feedback). Besides, XQuery does not provide means for querying multiple databases in one request and does not support multimodal or spatial/temporal queries. Nevertheless, there is ongoing work in this direction. For instance, the authors in [13] describe an XQuery extension for MPEG-7 vector-based feature queries. Furthermore, the authors in [3] adapted XQuery for the retrieval of MPEG-7 descriptions based on semantic views. Its adaptation, called Semantic Views Query Language (SVQL) is specialized for retrieving MPEG-7 descriptions in TV news retrieval applications and is not intended to be a general query language. The language presented here differs from other XML and MPEG-7 based approaches like [12, 8] because it keeps working over the data model defined by MPEG-7, and does not attempt to define a higher-level data model. The authors in [8] propose an XML query language with multimedia query constructs called MMDOC-QL. MMDOC-QL bases on a logical formalism path predicate calculus [7] which supports multimedia content retrieval based on described spatial, temporal and visual data types and relationships. MMDOC-QL has several drawbacks such as simultaneous searches in multiple databases or the integration of user preferences and usage history which are not considered in MMDOC-QL. The authors in [12] introduced *PTDOM* as a native schema aware XML database system for MPEG-7 media descriptions. Their system provides an MPEG-7 schema compliant schema catalogue whose main goal is besides document validation, an appropriate typed representation of document content (elements and attributes) supporting enhanced indexing and query optimization of non-textual document content.

3. Query Format

The in this paper described query format (MP7QF) transmits the query by using the eXtended Mark-up Language protocol (XML) [11]. XML is a general-purpose mark-up language with the primary purpose of sharing data across different information systems, particularly via the Internet. This language has been chosen for the query format, because MPEG-7 is based on XML in order to be compliant with previous versions of this standard. The schema description of the proposed format is based on the MPEG-7 Description and Definition Language (DDL) [4], which inherits XML schema [4]. Furthermore, the query format is fully compatible to XML schemas. The query format is destined to

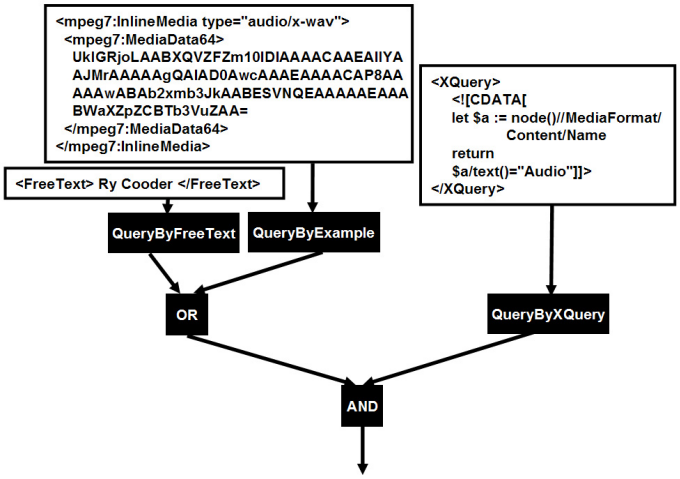

Figure 1. Example combination of query types

transmit MPEG-7 descriptors and description schemes. But since the MPEG-7 schema is already very complex and some user might want to use his own schema, a mechanism has been established to allow transmitting information from other schemas than MPEG-7 as well. The root node of the query format schema consists of an input type and an output type. The input format describes the query interface from the user to the database (see section 3.1), whilst the output query format (see section 3.2) describes the response from the server to the client. A further attribute has been implemented in order to specify an ID value for the query. This number could be used e.g. in an asynchronous mode, where the server not directly responds to the client and closes the connection. After a certain time, the client could ask again for the result with the previous ID.

3.1. Input Query Format

The input query format has been established in order to enable the communication between client and server. In this part of the query formulation, different operations and operators are handled, descriptions and content are defined and the expected result set is transmitted. The input query format node is named *InputType* and consists of three types: *QFDeclarationType*, *OutputDescriptionType* and *QueryConditionType*. These three types have been placed into a sequence in order to allow transferring a number of queries at the same time. *QFDeclarationType* allows to reference the actual content (e.g. MPEG-7 descriptions or audiovisual content), established to be able to describe the content once, but usable within the actual

query declaration as reference a couple of times without writing the same descriptor every time. Since the size of MPEG-7 descriptions can be large, the size of the query does not increase significantly when increasing complexity. The proposed type consists of a sequence of the type *ResourceType* and an attribute with an id as referencing number. The *ResourceType* can contain a choice of an MPEG-7 descriptor, a *TextAnnotation*, a *MediaResource* or an *AnyDescription*, so every kind or desired resource is supported for referencing. Especially to mention is the *AnyDescription* type. This type enables the use of own descriptors of arbitrary XML schemas. The *OutputDescriptionType* signals to the server, what result the client expects as answer.

The *ResultItemType* is the definition of the expected output in order to give the user the opportunity to specify the format of the output he expects, when submitting the query. The expected output can be a resource locator, freetext or metadata from any namespace. The *QueryConditionType* provides means for expressing restrictions and conditions in order to shrink the result set to the user's needs. The node consists of a sequence of *ConditionType* elements, which is an abstract base class. Inherited from *ConditionType* are the classes *OperatorType*, which describes the operators link the actions and *ConditionType* that describes the actual condition. An example for this kind of type is the *QueryByExampleType*, which looks for similar or exact items of the qiven example. At the moment five different types have been implemented into the CD. Besides the *QueryByExampleType* a *QueryByFreeTextType* has been defined in order to allow a freetext search. Furthermore an *XQueryType* has been implemented to search with XQuery expressions. XQuery is an expression language, defined by the W3C consortium to query within XML documents. The last defined type is the *QueryByRelevanceFeedbackType*, which describes a query operation that takes into consideration the result of the previous retrieval. The query operation enables a user to identify good and/or bad examples within a previous result set and to indicate to the retrieval system that it should retrieve e.g., "more examples like this one." The *QueryOperatorType* is an abstract base type and serves as root type for all operator classes such as Boolean, Comparison or Arithmetic types. Currently, only Boolean operators have been defined. Among these are AND, OR, NOT and XOR operators.

3.2. Output Query Format

The OQF specifies the expected output from the server to the client and the node within the XML schema is called *OutputType*. This node consists of a sequence of three different elements, named *GlobalFreeText*, *SystemMessage* and *ResultItem*. The *GlobalFreeText* element consists a string and specifies a text message that the server may want to reply to the client. The *SystemMessage* element consists of the type *SystemMessageType*, which replies a message to the client. This can be a choice between status messages, error messages and warnings. The *ResultItemType* specifies the actual result and consists of a sequence of *Resource*, *FreeText* and *Description* elements. The *Resource* element contains a URI of the object, The *FreeText* element contains a textual description and the *Description* element contains the namespace "'any"' in order to be able to reply descriptors from other schemas as MPEG-7.

Code 1 Example usage of the *QueryByExample* QueryType

```
<Mpeg7Query>
 <Input>
  <QFDeclaration>
   <Resource id="id1">
    <MPEG7DescriptionElement
                  xsi:type="mpeg7:MediaInstanceType">
     <mpeg7:InstanceIdentifier/>
     <mpeg7:MediaLocator>
      <mpeg7:InlineMedia type="audio/x-wav">
       <mpeg7:MediaData64>UklGRjoLAABXQVZFZm10IDIAAAACA
        AEAIlYAAJMrAAAAAgQAIADOAwcAAAEAAAACAP8AAAAAwABA
        b2xmb3JkAABESVNQEAAAAAEAAABWaXZpZCBTb3VuZAA=
       </mpeg7:MediaData64>
      </mpeg7:InlineMedia>
     </mpeg7:MediaLocator>
    </MPEG7DescriptionElement>
   </Resource>
  </QFDeclaration>
  <QueryCondition>
   <Condition xsi:type="QueryByExample">
     <ResouceREF>id1</ResouceREF>
   </Condition>
  </QueryCondition>
 </Input>
</Mpeg7Query>
```

4. Query Examples

The following examples illustrate some key aspects of MP7QF. The namespaces declaration of the *Mpeg7Query* element has been removed in all the XML fragments to ease readability. The first example (see Figure 1) shows graphically how Boolean operators can be used to combine conditions expressed using different query types like the *QueryByFreeText*, the *QueryByExample* or the *QueryByXQuery*. The presented query specifies the need to find audio contents similar to a given example WAV file (embedded within the query

in base64 encoding) and related to the free-text "Ry Cooder". The XML code of this example is not included to save space. Example in Code 1 shows the usage of the *QueryByExample* QueryType. The request includes an example audio file (in WAV format), which is directly embedded within a *Resource* element using base64 encoding. Instead of inlining the example within the *Condition* element (which is also possible), it is defined within the *QFDeclaration* section. This would allow, for example, to reuse the same data for other conditions just by referring to the same *id*. It is up to the server to decide which similarity measure and search algorithm to apply. The Example in Code 2 illustrates the usage of the *OutputDescription* element. It specifies the MPEG-7 namespace for the output formatting. Because the example does not specify values for the attributes *maxPageEntries* and *maxItemCount* it is up to the server how to paginate the results. The example neither specify values for the attributes *resourceUse* and *freeTextUse* which let's to the server the decision to include the fields *Resource* and *FreeText* in the result items. The example specifies the field *Title* from the MPEG-7 schema, which should appear in the result items if available.

Code 2 Example usage of the *OutputDescription* element

```
<Mpeg7Query>
 <Input>
  <OutputDescription
     outputNameSpace="urn:mpeg:mpeg7:schema:2004">
   <Field typeName="CreationType">/Title</Field>
  </OutputDescription>
  <QueryCondition>
   <Condition xsi:type="QueryByFreeText">
   <FreeText>Ry Cooder</FreeText></Condition>
  </QueryCondition>
 </Input>
</Mpeg7Query>
```

5. Conclusion and further work

This paper presented the work on the MPEG-7 query format. The standard document has already the status CD at the of the MPEG standardization committee. Therefore the core of the framework can be considered as stable. The root node consists of an input format for transmitting the query from the client to the server and an output format to establish the reply. There are, however, still some parts missing, which will be contributed in the future. One of the most important issues, that are not covered yet, are the Query Management Tools. They describe a management layer in order to set important operative parameters as timeout or operation mode. An outline of possible management tools can be found in [1]. The input format could be further extended in order to define other operators as e.g. arithmetic operators.

6 Acknowledgments

This work has been partly supported by the Spanish government (DRM-MM project, TSI 2005-05277), the Europ. Network of Excellence VISNET-II (IST-2005-2.41.6) and PHAROS (IST-2005-2.6.3), funded under the EC IST 6th Framework Program. The authors want to thank furthermore all participants of the MPEG Query Format Ad-Hoc group for their excellent and fruitful collaboration.

References

[1] M. Doeller, I. Wolf, M. Gruhne, and H. Kosch. Towards an MPEG-7 Query Language. In *Proceedings of the International Conference on Signal-Image Technology and InternetBased Systems (IEEE/ACM SITIS'2006)*, pages 382–384, Hammamet, Tunesia, 2006.

[2] N. Fatemi, O. A. Khaled, and G. Coray. An XQuery Adaptation for MPEG-7 Documents Retrieval. In *Proceedings of the XML Conference and Exposition*, Philadelphia, PA, USA, 2003.

[3] N. Fatemi, M. Lalmas, and T. Rlleke. How to retrieve multimedia documents described by MPEG-7. In *Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop, ACM Press,*, New York, NY, USA, 2004.

[4] ISO. ISO/IEC 15938-2:2001. *ISO, http://www.iso.org*, 2001.

[5] J. Kang and al. An xquery engine for digital library systems. In *3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, Houston, Texas, May 2003.

[6] J. Z. Li, M. T. Ozsu, D. Szafron, and V. Oria. MOQL: A Multimedia Object Query Language. In *Proceedings of the third International Workshop on Multimedia Information Systems*, pages 19–28, Como Italy, 1997.

[7] P. Lui, A. Charkraborty, and L. H. Hsu. Path Predicate Calculus: Towards a Logic Formalism for Multimedia XML Query Language. In *Proceedings of the Extreme Markup Languages*, Montreal, Canada, 2000.

[8] P. Lui, A. Charkraborty, and L. H. Hsu. A Logic Approach for MPEG-7 XML Document Queries. In *Proceedings of the Extreme Markup Languages*, Montreal, Canada, 2001.

[9] J. Melton and A. Eisenberg. SQL Multimedia Application packages (SQL/MM). *ACM SIGMOD Record*, 30(4):97–102, December 2001.

[10] D. Tjondronegoro and Y. Chen. Content-based indexing and retrieval using mpeg-7 and xquery in video data management systems. *World Wide Web: Internet and Web Information Systems*, pages 207–227, 2002.

[11] W3C. Extensible Markup Language (XML)- Introduction. *W3C, http://www.w3.org/XML/*, 2006.

[12] U. Westermann and W. Klas. PTDOM: a schema-aware XML database system for MPEG-7 media descriptions. *Software: Practice and Experience,*, 36(8):785–834, 2006.

[13] L. Xue, C. Li, Y. Wu, and Z. Xiong. VeXQuery: an XQuery extension for MPEG-7 vector-based feature query . In *Proceedings of the International Conference on Signal-Image Technology and InternetBased Systems (IEEE/ACM SITIS'2006)*, pages 176–185, Hammamet, Tunesia, 2006.