The MPEG Query Format, a New Standard For Querying Digital Content. Usage in Scholarly Literature Search and Retrieval

Ruben Tous[1] and Jaime Delgado[2]

Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya (UPC)
Departament d'Arquitectura de Computadors Universitat Politècnica de Catalunya (UPC) Mòdul D6, Campus Nord C/ Jordi Girona, 1-3, E-08034 Barcelona, Spain
e-mail: [1]rtous@ac.upc.edu; [2]jaime.delgado@ac.upc.edu

Abstract

The initiative of standardization of MPEG Query Format (MPQF) has refueled the research around the definition of a unified query language for digital content. The goal is to provide a standardized interface to multimedia document repositories, including but not limited to multimedia databases, documental databases, digital libraries, spatio-temporal databases and geographical information systems. The initiative is being led by MPEG (i.e. ISO/IEC JTC1/SC29/WG11). This paper presents MPQF as a new approach for retrieving multimedia document instances from very large document databases, and its particular application to scholarly literature search and retrieval. The paper also explores how MPQF can be used in combination with the Open Archives Initiative (OAI) to deploy advanced distributed search and retrieval services. Finally, the issue of rights preservation is discussed.

Keywords: scholarly literature; search, framework; query format, MPQF; Open Archives Initiative; MPEG

1. Introduction

During the last years, the technologies enabling search and retrieval of multimedia digital contents have gained importance due to the large amount of digitally stored multimedia documents. Therefore, members of the MPEG standardization committee (i.e. ISO/IEC JTC1/SC29/WG11) have developed a new standard, the MPEG Query Format (MPQF) [1, 2, 3], which provides a standardized interface to multimedia document repositories, including but not limited to multimedia databases, documental databases, digital libraries, spatio-temporal databases and geographical information systems.

The MPEG Query Format offers a new and powerful alternative to the traditional scholarly communication model. MPQF provides scholarly repositories with the ability to extend access to their metadata and contents via a standard query interface, in the same way as Z39.50 [4], but making use of the newest XML querying tools (based in XPath 2.0 [5] and XQuery 1.0 [6]) in combination with a set of advanced multimedia information retrieval capabilities defined within MPEG. This would allow, for example, querying for journal papers by specifying constraints over their related XML metadata (which is not restricted to a particular format) in combination with similarity search, relevance feedback, query-by-keywords, query-by-example media (using an example image for retrieving papers with similar ones), etc. MPQF has been designed to unify the way digital materials are searched and retrieved. This has important implications in the near future, when scholarly users' information needs will become more complex and will involve searches combining (in the input and the output) documents from different nature (e-prints, still images, audio transcripts, video files, etc.).

Currently, several forums, like [7], are trying to identify the necessary steps that could be taken to improve interoperability across heterogeneous scholarly repositories. The specific goal is to reach a common understanding of a set of core repository interfaces that would allow services to interact with heterogeneous repositories in a consistent manner. Such repository interfaces include interfaces that support locating, identifying, harvesting and retrieving digital objects. There's an open discussion about if the interoperability framework may benefit from the introduction of a search interface service. In general, it is felt that, while such interface is essential, it should not be part of the core, and that it could be implemented as an autonomous service over one or more digital repositories fed through interaction with core repository interfaces for harvesting like the Open Archives Initiative (OAI) [8]. We defend that MPQF could be this search interface service, deployed in the last mile of the value chain, offering powerful and innovative ways to express user information needs.

2. Related work

In general, the preferred method for distributed acquisition to digital content repositories is metadata harvesting. Metadata harvesting consists on collecting the metadata descriptions of digital items (usually in XML format) from a set of digital content repositories and storing them in a central server. Metadata is lighter than content, and it's feasible to store the necessary amount of it in an aggregation server so that real-time access to information about distributed digital content can take place without the burden of initiating a parallel real-time querying of the underlying target content databases.

Nowadays, the preferred harvesting method is the one offered by the Open Archives Initiative (OAI), which defines a mechanism for harvesting XML-formatted metadata from repositories (usually within the scholarly context). The OAI technical framework is intentionally simple with the intent of providing a low barrier for implementers and users. The trade-off is that its query expressiveness and output format description is very limited. In OAI Protocol for Metadata Harvesting (OAI-PMH), metadata consumers or "harvesters" request metadata from updated records from the metadata producers or "repositories" (data providers are required to provide XML metadata at least in Dublin Core format). These requests can be based on a timestamp range, and can be only restricted to named sets defined by the provider. These sets provide a very limited form of selective harvesting, and do not act as a search interface. Consequently some repositories may provide other querying interfaces with richer functionality, usually in addition to OAI. The two principal examples are Z39.50 and SRU-CQL [9, 10].

Regarding OAI, the MPEG Query Format (MPQF) could also be used for harvesting (though in that case a metadata format offering records update timestamps would be needed), overlapping with the OAI functionalities. However, MPQF is a complex language which has been designed for fine-grained retrieval and more advanced filtering capabilities. Because OAI offers a specialised, low-barrier and mature protocol for harvesting, we think that both mechanisms should be used in conjunction.

With respect to Z39.50 and related protocols/languages, MPQF surpasses their expressive power by offering a flexible combination of XML-based query capabilities with a broad set of multimedia information retrieval capabilities. A major difference with respect to the Z39.50 approach is that MPQF does not define abstract data structures to which the queries refer, instead, MPQF queries use generic XPath and XQuery expressions written in terms of the expected metadata format of the target databases. We envisage the usage of MPQF and its expressive power directly between user agents and service providers, while the OAI will be probably used through the rest of the value chain.

Regarding other multimedia query formats, there exist several languages explicitly for multimedia data such as SQL/MM [11], MOQL [12] or POQLMM [13], which are out of scope of this paper based on its limitation in handling XML data. Today, these kind of works use to be based on MPEG-7 descriptors and the MPEG-7 data model. Some simply defend the use of XQuery or some extensions of it. Others define a more high-level and user-oriented approach. MPQF outperforms XQuery-based approaches like [14, 15, 16] because, while offering the same level of expressiveness, it offers multiple content-based search functionalities (QBE, query-by-freetext) and other 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.

3. MPEG Query Format

3.1. Concepts and benefits

Formally, MPQF is Part 12 of ISO/IEC 15938-12, "Information Technology - Multimedia Content Description Interface" better known as MPEG-7 [17]. The standardization process started in July 2006 with the release of a "Call for Proposals on MPEG-7 Query Format" [18]. However, the query format was technically decoupled from MPEG-7 during the 81st MPEG meeting in July 2007, and its name changed to "MPEG Query Format" or simply "MPQF". The standardization process has proceeded and it is expected that MPQF will become an ISO/IEC final standard after the 85th MPEG meeting in July 2008.

Basically, MPQF is an XML-based query language that defines the format of queries and replies to be interchanged between clients and servers in a distributed multimedia information search-and-retrieval context. The two main benefits of standardizing such kind of language are 1) interoperability between parties (e.g. content providers, aggregators and user agents) and 2) platform independence; developers can write their applications involving multimedia queries independently of the database used, which fosters software reusability and maintainability. The major advantage of having MPEG rather than industry forums leading this initiative is that MPEG specifies international, open standards targeting all possible application domains and which, therefore, are not conditioned by partial interests or restrictions.

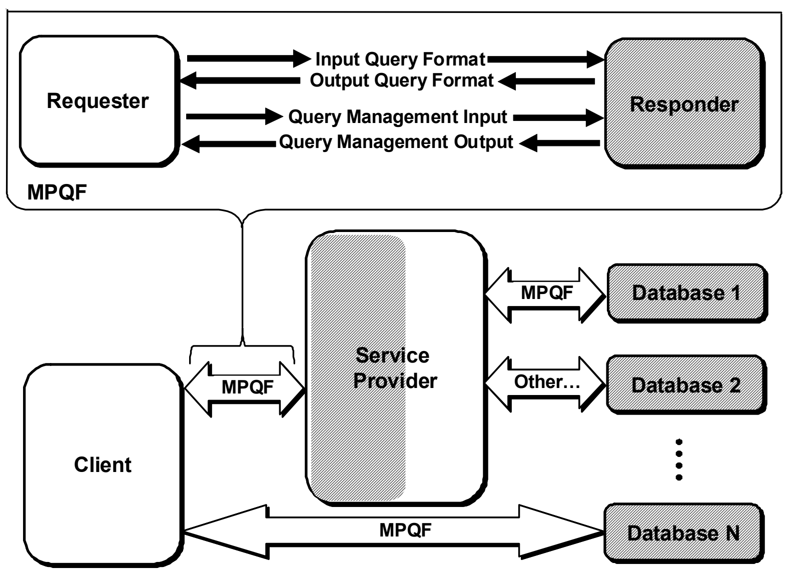

Figure 1. MPEG Query Format diagram

MPQF defines a request-reply XML-based interface between a requester and a responder. Figure 1 shows a diagram outlining the basic MPQF scenario. In the simplest case, the requester may be a user's agent and the responder might be a retrieval system. However, MPQF has been specially designed for more complex scenarios, in which users interact, for instance, with a content aggregator. The content aggregator acts at the same time as responder (from the point-of-view of the user) and as a requester to a number of underlying content providers to which the user query is forwarded.

3.2. Multimedia information retrieval vs. (XML) data retrieval

One of the novel features of MPQF is that it allows the expression of queries combining both the expressive style of information and XML Data Retrieval systems. Thus, MPQF allows combining e.g. keywords and query-by-example with e.g. XQuery allowing the fulfillment of a broad range of users' multimedia information needs. Both approaches to data retrieval aim to facilitate users' access to information, but from different points-of-view. On one hand, given a query expressed in a user-oriented manner (e.g. an image of a bird), an Information Retrieval system aims to retrieve information that might be relevant even though the query is not formalized. In contrast, a Data Retrieval system (e.g. an XQuery-based database) deals with a well defined data model and aims to determine which objects of the collection satisfy clearly defined conditions (e.g. the title of a movie, the size of a video file or the fundamental frequency of an audio signal). Regarding Information Retrieval, MPQF offers a broad range of possibilities that include but are not limited to query-by-example-media, query-byexample-description, query-by-keywords, query-by-feature-range, query-by-spatial-relationships, query-by-temporalrelationships and query-by-relevance-feedback. For Data Retrieval, MPQF offers its own XML query algebra for expressing conditions over the multimedia related XML metadata (e.g. MPEG-7, Dublin Core or any other XMLbased metadata format) but also offers the possibility to embed XQuery expressions (see Figure 2).

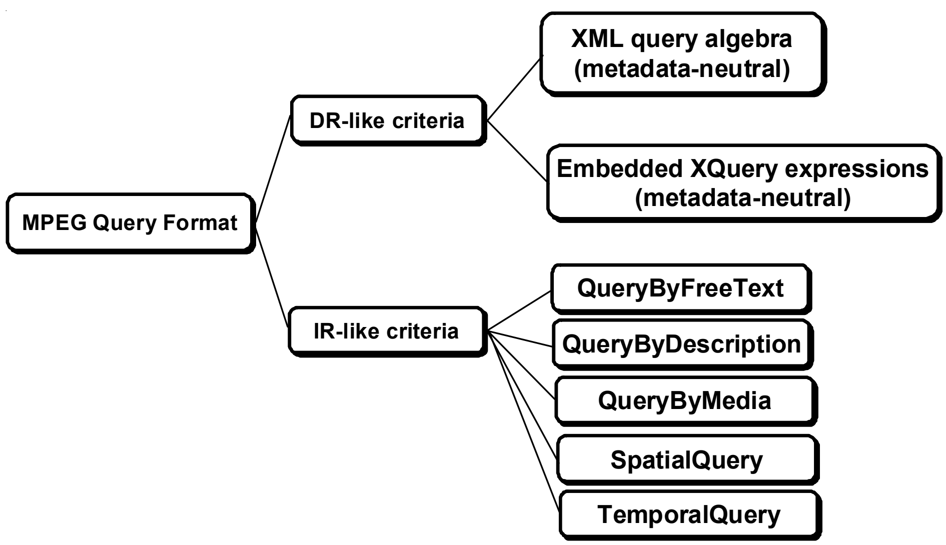

Figure 2. MPQF IR and DR capabilities

3.3. Language parts

MPQF instances are XML documents that can be validated against the MPQF XML schema. Any MPQF instance includes always the *MpegQuery* element as the root element. Below the root element, an MPQF document includes the *Query* element or the *Management* element. MPQF instances with the *Query* element are the usual requests and responses of a digital content search process. The *Query* element can include the *Input* element or the *Output* element, depending if the document is a request or a response. The part of the language describing the contents of the *Input* element is named the Input Query Format (IQF) in the MPQF standard. The part of the language describing the *Output* element is named the Output Query Format (OQF) in the standard. IQF and OQF are just used to facilitate understanding, but do not have representation in the schema. Alternatively, below the root element, an MPQF document can include the *Management* element. Management messages (which in turn can be requests and responses) provide means for requesting service-level functionalities like discovering databases or other kind of service providers, interrogating the capabilities of a service, or configuring service parameters.

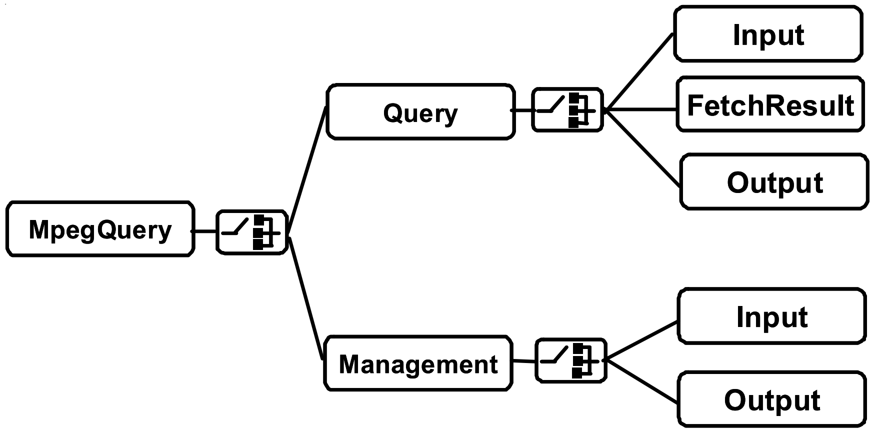

Figure 3. MPQF Schema root elements

3.4 Input Query Format (IQF)

The MPQF's Input Query Format (IQF) mainly allows specifying the search condition tree, which represents the user's search criteria, and also the structure and desired contents of the resultset. The condition tree is the main component of MPQF, and can be built combining different kids of expressions and query types. When analyzing an MPQF condition tree, one must consider that it will be evaluated against an unordered set of Multimedia Content (MC). The concept of Multimedia Content [17] is analogous to the concept of Digital Object from the Digital Libraries area, and refers to the combination of multimedia data and its associated metadata. MPQF allows search and retrieval of complete or partial MC data and metadata. Conditions within the condition tree operate on one evaluation-item (EI) at a given time (or two EIs if a Join operation is used). By default, an

Evaluation Item (EI) is a multimedia content in the database, but other types of EIs are also possible (spatial or time regions, metadata fragments, etc.).

Figure 4 outlines the main elements of the IQF part of the MPQF schema. The condition tree is placed within the *QueryCondition* element, and is constructed combining boolean operators (AND, OR, etc.), simple conditions over the XML metadata fields and query types (*QueryByFreeText*, *QueryByMedia*, etc.). Example in Code 1 shows the MPQF query asking for PDF research papers related to the keywords "Open Access" with a Dublin Core *date* element greater or equal to *2008-01-15*. Note that the query expects the target repository exposing Dublin Core descriptors. Exposing Dublin Core metadata is not required for an MPQF-compliant server, therefore the requester must previously ask the repository which metadata formats support.

3.5 Output Query Format (OQF)

The MPQF's Output Query Format (OQF) allows specifying the structure of the resultset. By default, the resultset includes some fields like the resource locator (*MediaResource* element in MPQF) but MPQF allows also selecting specific XML elements from one or more target namespaces. MPQF allows sorting and grouping result records, but it is deliberately rigid in the way records are presented. Unlike XQuery, which allows defining any possible structure for a result, MPQF records always share the same structure at the top levels. As shown in Figure 5, for each record a *ResultItem* element is returned. Within each

ResultItem, generic information about the record is placed within the *Comment*, *TextResult* and *MediaResource* elements, while the *Description* element is reserved for encapsulating the XML fields which have been selected in the query.

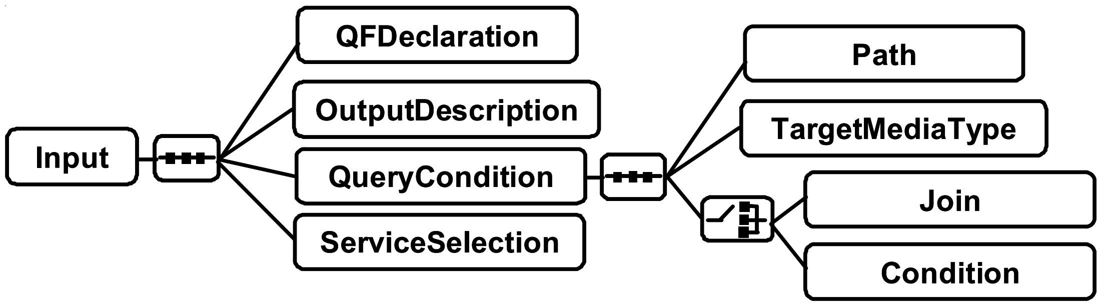

Figure 4. Input Query Format (IQF)

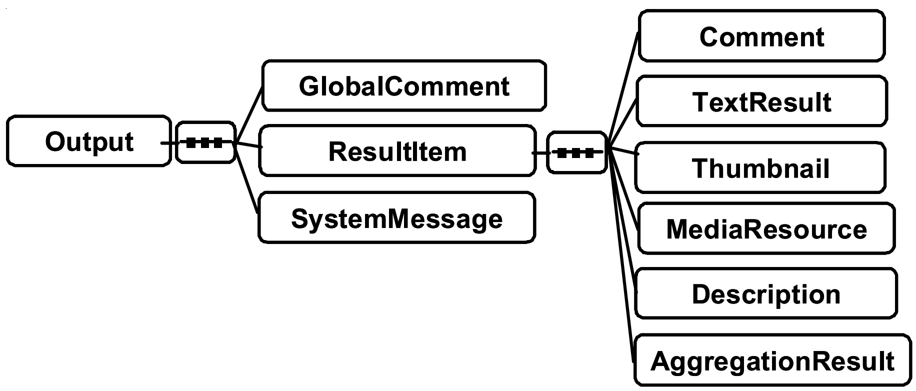

Figure 5. Output Query Format (OQF)

Example in Code 2, gives an idea of how the result of the query in Code 1 could look like. The resultset consists on two records which match the query conditions, and include the Dublin Core elements which have been selected (*title*, *creator*, *publisher* and *date*).

```
<MpegQuery>
  <Query>
    <Input>
      <OutputDescription outputNameSpace="//purl.org/dc/elements/1.1/">
        <ReqField>title</ReqField>
        <ReqField>creator</ReqField>
        <ReqField>publisher</ReqField>
        <ReqField>date</ReqField>
      </OutputDescription>
      <QueryCondition>
        <TargetMediaType>application/pdf</TargetMediaType>
        <Condition xsi:type="AND" preferenceValue="10">
          <Condition xsi:type="QueryByFreeText">
            <FreeText>Open Access</FreeText>
          </Condition>
          <Condition xsi:type="GreaterThanEqual">
            <DateTimeField>date</DateTimeField>
            <DateValue>2008-01-15</DateValue>
          </Condition>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Code 1: Input query example

```
<MpegQuery mpqfID="AB13DGDDE1">
  <Query>
    <Output>
      <ResultItem xsi:type="ResultItemType" recordNumber="1">
        <TextResult>Some advertising here</TextResult>
        <MediaResource>http://www.repository.com/item04.pdf</MediaResource>
        <Description xmlns:dc="http://purl.org/dc/elements/1.1/"
xsi:schemaLocation="http://purl.org/dc/elements/1.1/ dc.xsd">
          <dc:title>Open Access Overview</dc:title>
          <dc:creator>John Smith</dc:creator>
          <dc:publisher>VDM Verlag</dc:publisher>
          <dc:date>2008-02-21</dc:date>
        </Description>
      </ResultItem>
      <ResultItem xsi:type="ResultItemType" recordNumber="2">
        <TextResult>Some advertising here</TextResult>
        <MediaResource>http://www.repository.com/item08.pdf</MediaResource>
        <Description xmlns:dc="http://purl.org/dc/elements/1.1/"
xsi:schemaLocation="http://purl.org/dc/elements/1.1/ dc.xsd">
          <dc:title>Open Access in Germany</dc:title>
          <dc:creator>John Smith</dc:creator>
          <dc:publisher>VDM Verlag</dc:publisher>
          <dc:date>2008-02-01</dc:date>
        </Description>
      </ResultItem>
    </Output>
  </Query>
</MpegQuery>
```

Code 2: Output query example

4. Open Archives and MPQF together. Scholarly objects interchange framework

We envisage that MPQF could be one building block of a future scholarly objects interchange framework, interconnecting heterogeneous scholarly repositories. The framework would be based on the combination of the Open Archives Initiative (OAI) protocol for metadata harvesting (OAI-PMH) with MPQF. Figure 6 outlines graphically the basic elements of the framework in an example scenario.

Required search functionalities amongst the different parties in the framework vary depending on their roles. On one hand, aggregators (e.g. librarians) need collecting metadata descriptions from repositories (e.g. publishers) or between them, and this is usually performed through a harvesting mechanism. On the other hand, content "retailers", which include aggregators and also some repositories (generally medium or large scale ones), should be able to deploy value-added services offering fine-grained access to digital objects, and advanced search and retrieval capabilities. We believe that the MPEG Query Format could be the search interface between "retailers" and users, in the last mile of the value chain, offering expressive ways to represent user information needs.

The scenario from Figure 6 does not cover the real-time distributed usage of MPQF. Our experience in previous projects like [19] and [20] makes us think that real-time distributed search imposes severe limitations in terms of interoperability and performance, and it is not always necessary. However, this scenario is just an example, and nothing limits the distributed usage of MPQF (the standard provides extensive capabilities for that).

5. Advanced examples

5.1 *QueryByMedia* example: Searching research papers with similar images

Example in Code 3 shows the MPQF query asking for PDF research papers including images similar to a given one. An example usage of this query could be the detection of image copyright infringement. For instance, it could have been used in the first 90s, when the Playboy magazine discovered that an image copyrighted by the company in 1972, the Lena Sjooblom's photo (Figure 7), was being widely used in image processing research papers.

The query includes a *Condition* element from the *QueryByMedia* complex type. Query-by-example similarly searches allow to express the user information need with one or more example digital objects (e.g. an image file). Though the usage of low-level features description instead of the example object bit stream is also considered query-by-example, in MPQF these two situations are differentiated, naming query-by-media to the first case (the digital media itself) and query-by-description the second one.

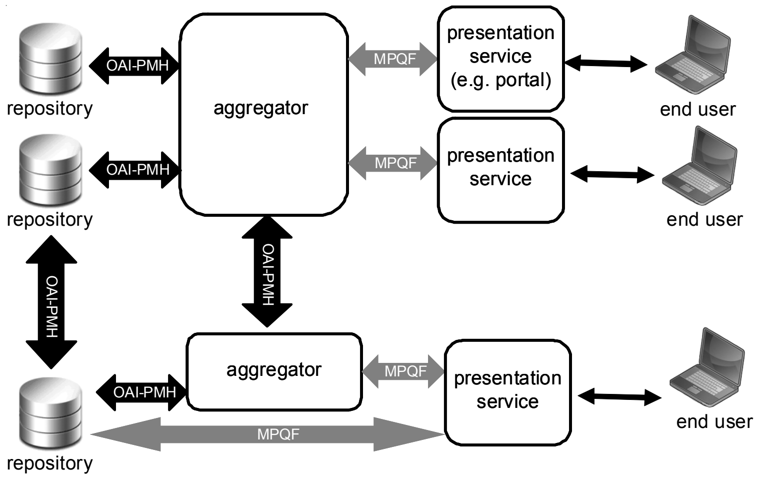

Figure 6. OAI+MPQF Example scenario

Figure 7. Lena Sjooblom's photo from 1972 and a research paper where it appears [21]

```
<MpegQuery>
  <Query>
    <Input>
      <QueryCondition>
          <TargetMediaType>application/pdf</TargetMediaType>
          <Condition xsi:type="QueryByMedia">
            <MediaResource xsi:type="MediaResourceType">
              <MediaResource>
                <InlineMedia type="image/jpeg">
                  <MediaData64>R0lGODlhDwAPAKECAAAAzMzM/////wAAACwAAAAADwA
                  PAAACIISPeQHsrZ5ModrLlN48CXF8m2iQ3YmmKqVlRtW4MLwWACH+H09
                  wdGltaXplZCBieSBVbGVhZCBTbWFydFNhdmVyIQAAOw==</MediaData64>
                </InlineMedia>
              </MediaResource>
            </MediaResource>
         </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Code 3: *QueryByMedia* example

QueryByMedia and *QueryByDescription* are the fundamental operations of MPFQ and represent the query-by-example paradigm. The individual difference lies in the used sample data. The *QueryByMedia* query type uses a media sample such as image as a key for search, whereas *QueryByDescription* allows querying on the basis of an XML-based description. Code 4 shows how an example Dublin Core description can be included in a query. The server should return records corresponding to digital objects with metadata similar to the given ones. It's up to the server deciding which similarity algorithm to apply.

```
<MpegQuery>
  <Query>
    <Input>
      <QueryCondition>
        <TargetMediaType>application/pdf</TargetMediaType>
        <Condition xsi:type="QueryByDescription" matchType="exact">
          <DescriptionResource resourceID="desc07">
            <AnyDescription xmlns:dc="http://purl.org/dc/elements/1.1/">
                <dc:title>Open Access Overview</dc:title>
                <dc:creator>John Smith</dc:creator>
                <dc:publisher>VDM Verlag</dc:publisher>
                <dc:date>2008-02-21</dc:date>
            </AnyDescription>
          </DescriptionResource>
        </Condition>
      </QueryCondition>
    </Input>
  </Query>
</MpegQuery>
```

Code 4: *QueryByDescription* example

5.3 *QueryByRelevanceFeedback* example. Refining the search of a research paper

In Information Retrieval, "relevance feedback" is related to taking the relevance of the results that are initially returned from a given query to improve the results of a new query. MPQF offers the possibility of "explicit" relevance feedback by allowing user to mark specific records as relevant or irrelevant. This is accomplished through the usage of the *QueryByRelevanceFeedback* query type. Let's take again example from Code 1 and Code 2. The user was looking for research papers related to the words "*Open Access*" and submitted a query (Code 1) to the server. The server responded with several records (within a response with id "*AB13DGDDE1*"), some of which are shown in Code 2. Let's imagine that the user found specially interesting the records number 1, 2 and 5. By using the *QueryByRelevanceFeedback* query type, as shown in Code 5, the user cand submit to the server his/her preferences, allowing the server to refine the response.

```
<MpegQuery>
<Query>
  <Input>
    <QueryCondition>
      <Condition xsi:type="QueryByRelevanceFeedback" answerID="AB13DGDDE1">
        <ResultItem>1</ResultItem>
        <ResultItem>2</ResultItem>
        <ResultItem>5</ResultItem>
      </Condition>
    </QueryCondition>
  </Input>
  </Query>
</MpegQuery>
```

Code 5: *QueryByRelevanceFeedback* example

The presented examples just demonstrate a small part of MPQF capabilities, but just pretend to show the particularities of this language in comparison to other existing multimedia querying facilities, and specially in comparison to existing scholarly search interfaces.

6. Conclusions

This paper proposes the usage of a novel standard, the MPEG Query Format, to extend the functionality and to foster the interoperability of scholarly repositories search interfaces. The paper defends that future scholarly digital objects interchange frameworks could be based on the combination of MPQF and the Open Archives protocol. While Open Archives offers a low-barrier mechanism for "wholesale" metadata interchange, MPQF provides scholarly repositories with the ability to extend access to their metadata and contents via a standard query interface, making use of the newest XML querying tools (based in XPath 2.0 and XQuery 1.0) in combination with a set of advanced multimedia information retrieval capabilities defined within MPEG.

The paper describes also how this idea can be applied to the design of a scholarly objects interchange framework. The framework interconnects heterogeneous scholarly repositories and it is based on the combination of two standard technologies such as the OAI-PMH protocol and the MPEG Query Format. The design has been guided by the conclusions of a previous experience, the XAC project [20], from which several lessons were learnt, as the necessary separation between metadata harvesting and real-time search and retrieval, or the necessity to choose a more appropriate query format than XQuery. We are working currently in the first implementation of the framework. It is worth mention that it is planned that from this work, the first known implementation of an MPEG Query Format processor will emerge. Furthermore, parts of the ongoing implementation are being contributed to the MPEG standardisation process in the form of Reference Software modules.

Finally, it is also relevant to indicate that, in fact, we are working with a third standard, the MPEG-21 Rights Expression Language [22] and its extensions, in order to also cover rights management issues. Although it has not been the focus of this paper, we have also considered in our framework the possibility of having licenses associated to the content being distributed. Those licenses specify rights and conditions that apply to a resource for a specific user, and may be used, through an authorization process, to enforce these rights and conditions during the consumption of protected content.

In [22] we have already developed some tools to create licenses, to verify them, to decide if a specific consumption is to be authorised and to distribute information about all events happening on the content. Apart from this, we have participated in the development of a system [23] that allows controlling the rights related to the whole life cycle of intellectual property, from its creation to the final usage. We are currently considering adapting our system to specifically handle scholarly content, which would allow authors to register their work, before sending for reviewing or publication, to decide about the rights they want to give to their creations, and to control over the events related to them.

7. Acknowledgments

This work has been partly supported by the Spanish government (DRM-MM project TSI 2005-05277) and the European Network of Excellence (VISNET-II IST-1-038398), funded under the European Commission IST 6th Framework Program.

8. References

[1] ISO/IEC FDIS 15938-12:2008 "Information Technology — Multimedia Content Description Interface — Part 12: Query Format".

[2] Gruhne, Matthias; Tous, Ruben; Doeller Mario; Delgado, Jaime and Kosch, Harald (2007). MP7QF: An MPEG-7 Query Format. 3rd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS 2007), Barcelona, November 2007. IEEE

Computer Society Press. ISBN 0-7695-3030-3. p. 15-18
[3] Kevin Adistambha et al. (2007). The MPEG-7 Query Format: A New Standard in Progress for Multimedia Query by Content. 7th International Symposium on Communications and Information Technologies (ISCIT 2007), Sydney, Australia, October 16-19, 2007. IEEE Computer Society Press.
[4] ISO 23950. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification.
[5] XQuery 1.0: An XML Query Language. W3C Recommendation 23 January 2007. See http://www.w3.org/TR/xquery/.
[6] XML Path Language (XPath) 2.0. W3C Recommendation 23 January 2007. See http://www.w3.org/TR/xpath20/.
[7] Tony Hey, Herbert Van de Sompel, Don Waters, Cliff Lynch, Carl Lagoze Augmenting interoperability across scholarly repositories JCDL '06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries June 2006
[8] Open Archives Initiative. http://www.openarchives.org/.
[9] Library of Congress, 2004. SRU: SRU (Search Retrieve via URL). See http://www.loc.gov/standards/sru/sru-spec.html.
[10] Common query language. See http://www.loc.gov/z3950/agency/zing/cql/cqlsyntax.html.
[11] J. Melton and A. Eisenberg. SQL Multimedia Application packages (SQL/MM). ACM SIGMOD Record, 30(4):97–102, December 2001.
[12] 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.
[13] A. Henrich and G. Robbert. POQLMM: A Query Language for Structured Multimedia Documents. In Proceedings 1st International Workshop on Multimedia Data and Document Engineering (MDDE'01), pages 17–26, July 2001.
[14] 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.
[15] 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.
[16] 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.
[17] ISO/IEC 15938 Version 2 "Information Technology - Multimedia Content Description Interface" (MPEG-7).
[18] ISO/IEC JTC1/SC29/WG11 N8220. July 2006 "Call for Proposals on MPEG-7 Query Format".
[19] Ruben Tous, Jaime Delgado. Advanced Meta-Search of News in the Web , ELPUB2002 - Technology Interactions. Proceedings of the 6th International ICCC/IFIP Conference on Electronic Publishing held in Karlovy Vary, Czech Republic, 6–8 November 2002. VWF Berlin, 2002. ISBN 3-89700-357-0.
[20] J. Delgado, S. Llorente, E. Peig, and A. Carreras. A multimedia content interchange framework for TV producers. 3rd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS 2007), Barcelona, November 2007. IEEE Computer Society Press. ISBN 0-7695-3030-3. p. 206-213. 2007.
[21] Jose A. Rodrigo, Tatiana Alieva, Maria L. Calvo, Applications of gyrator transform for image processing, Optics Communications. Volume 278, Issue 2, 15 October 2007, Pages 279-284.
[22] ISO/IEC, Information Technology – Multimedia framework (MPEG-21) – Part 5:Rights Expression Language, ISO/IEC 21000- 5:2004, March 2004.
[23] IPOS-DS (Intellectual Property Operations System – Digital Shadow), exploited by NetPortedItems, S.L. http://www.digitalmediavalues.com.