ARCHITECTURE FOR AN MPEG-7 WEB BROWSER

Òscar Celma
Music Technology Group
Universitat Pompeu Fabra

ABSTRACT

The MPEG-7 standard provides description mechanisms and taxonomy management for multimedia documents. There are several approaches to design a multimedia database system using MPEG-7 descriptors. We discuss two of them: relational databases and native XML databases. We have implemented a search and retrieval application for MPEG-7 descriptions based on the latter.

1. INTRODUCTION

Multimedia metadata information represents —from the service providers' point of view— a nice way to add value to multimedia resources (i.e audiovisual files) distribution. Managing audiovisual essence implies to structure its associated metadata and content-based descriptors; using description schemes, taxonomies and ontologies, to organize a meaningful data knowledge representation.

MPEG-7 standard provides content description for audiovisual content, defining normative elements as Descriptors, Description Schemes and a Description Definition Language (DDL). The main use of MPEG-7 standard is, then, to describe all the information of multimedia assets. In that context, we usually talk about metadata (that is data about data) and, in that particular case, data about the multimedia information that is described. Thus, creating MPEG-7 documents allow a user to query and retrieve (parts of) multimedia and audiovisual information.

For many years, database management systems (DBMS) have been used to implement efficient text based information retrieval systems. Nevertheless, in the area of multimedia, and in particular in the Music Information Retrieval field, there is still a lot of ongoing research and there are open questions concerning architectural and design aspects related to DBMS.

In this paper we will expose and review different approaches to manage MPEG-7 with DBMS. Moreover, a web-based system to search and retrieve MPEG-7 descriptions, will be shown. The work presented in this paper has been carried on under the frame of the OpenDrama European IST project.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2004 Universitat Pompeu Fabra.

2. OPENDRAMA PROJECT

OpenDrama European IST project aims the definition, development and integration of a novel platform to author and to deliver rich cross-media digital objects of lyric opera. MPEG-7 has been used in OpenDrama as base technology for a music information retrieval system.

The most important services derived from OpenDrama are: *(i)* a search and retrieval system to browse mutimedia content and *(ii)* the streaming of metadata synchronized with multimedia data. System architecture is based on figure 1. For both OpenDrama's services, the use of the MPEG-7 standard fits the needs. In addition, the use of standards in such a project is a key point to make it a real open, interoperable, maintainable and durable system.

3. OVERVIEW OF MPEG-7 STANDARD

MPEG-7, formally named Multimedia Content Description Interface, aims to create a standard for the description of the multimedia content data. The main goal of the MPEG-7 standard is to provide structural and semantic description mechanism for multimedia content [1].

MPEG-7 descriptors are designed for describing different types of information; from low-level audiovisual features, to high-level semantic objects. Ideally, most descriptors corresponding to low-level would be extracted automatically, whereas some human intervention would be required for producing high-level descriptors.

XML has been adopted as the format to represent MPEG-7 descriptors. Also, MPEG-7 DDL is an extension of the XML Schema (published by W3C [1]). XML Schema provides the means for defining the structure of XML documents, that is; simple and complex data types, type derivation and inheritance, element ocurrence constraints and, finally, namespace-aware for elements and attributes declarations.

4. MPEG-7 AND MULTIMEDIA DATABASE SYSTEMS

Kosch [2] defines an architecture for a multimedia database management system that includes MPEG-7 descriptions, as well as the streaming service of the associated audiovisual files. An architecture of a multimedia database, inspired in [2], is depicted in figure 1.

[1] http://www.w3.org/XML/Schema

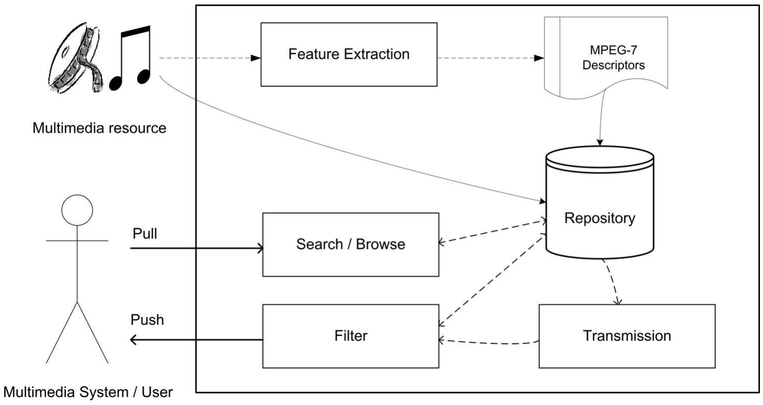

Figure 1. Multimedia database architecture using MPEG-7.

Starting from the feature extraction and annotation process of a multimedia asset, the MPEG-7 descriptors are generated and stored in a repository. Typically, in a multimedia database system one can distinguish two query scenarios: *pull* and *push*. In a *pull* scenario, a user submits queries to the system and receives a set of descriptions satisfying the constraints of the query. On the other hand, in a *push* scenario, a software agent selects MPEG-7 descriptions and performs a set of actions afterwards. One of these actions could be, for instance, proposing to users media information and its description based on their preferences. Hence the user agent is filtering audiovisual information according to metadata description.

There are several approaches and paradigms for structuring MPEG-7 data into a database system. In this paper we point out two general solutions: *(i)* to model MPEG-7 data into a relational database system and *(ii)* to use a native XML database (XML:DB) [3]. The former is based on the classic concept of a relation, while the latter has XML documents as its fundamental unit of (logical) storage in the database.

The next two sections explain both approaches.

4.1. Relational Databases

The work that has been previously done for structuring MPEG-7 data into a database system is based, mostly, on the classical relational model —plus some extensions to adapt the XML information into the relations. For instance, Jacob [4] has implemented a database to manage descriptions of sound objects —in MPEG-7— using a PostgreSQL database. A set of extensions (mainly programming triggers) has been designed to take into account insertions, updates and deletions of elements in an MPEG-7 document. An extraction rule engine allows to generate the MPEG-7 data. Yet, it is not clear, from a database user point of view, how to query (*select*) the MPEG-7 data inside the DB. Döller and Kosch [5] have designed an *MPEG-7 Multimedia Data Cartridge*. This system is an extension of object-relational DBMS Oracle 9i, providing a multimedia query language, access to media, and indexing capacities. Descriptors in the MPEG-7 schema are mapped to object types and tables, thus allowing to express queries in an hybrid SQL and XPath language.

Both systems permit to validate XML elements with the XML Schema (i.e validating MPEG-7 descriptions using the MPEG-7 DDL), providing a way to assure data integrity. However, due to the fact that the MPEG-7 DDL is tightly associated to the XML Schema definition, and the difficulties of reverse-engineering the model, managing MPEG-7 descriptors is equivalent to managing XML documents [6]. Thus, in this approach there is a big of effort in transducing the whole MPEG-7 Schema within a set of relations (i.e tables). Even so, possible changes on the MPEG-7 standard would imply to redo part of the database schema, which might be unfeasible when a system is being exploited. To cope this problem in a general sense, there has been considerable research concerning the automatic mapping between schema definitions of XML documents and relational database schema ([7], [8]), but most of the work is focused on DTD definitions instead of XML Schema [2], so they do not suffice for the management of MPEG-7 data.

4.2. Native XML Databases

Our approximation to structuring MPEG-7 descriptors is to use a native XML:DB. According to Bourret [9], there are more than 35 (open source and commercial) native XML database systems. Native XML:DB define a (logical) model for an XML document, and stores and retrieves documents according to that model. Native XML:DB make use of collections as internal folders for repositories of XML documents.

Westerman [6] has reviewed the existing database systems that can manage MPEG-7 media descriptions. Their study includes a set of native XML database. A table comparision between systems unveils a lack of data integrity validation by most of them. Data integrity is a key point in any database system. In a native XML:DB integrity validation is done by parsing an XML document through its

[2] The critical difference between DTDs and XML Schema is that XML Schema uses an XML-based syntax, whereas DTDs have a unique syntax held over from SGML DTDs

schema definition. Data integrity should be verified after an insertion of a new XML document to the DB, or after a modification of an already existent document. None of the native XML:DB presented in [6] allow full schema validation of MPEG-7 descriptors through MPEG-7 DDL. To solve this issue, a proposal of schema validation applied to native XML:DB is presented in [10].

As native XML:DB are still reaching maturity, another important aspect is to define languages that allow to query, insert, update and delete elements in the document. The most used languages —by XML:DB implementations— to query and to retrieve (part of) documents are the W3C *XPath 2.0*[3] and *XQuery 1.0*[4] recommendations. *XQuery* is a functional, strongly typed language that satisfies the requirements of a database query language. Updating XML data is possible with *XUpdate* initiative[5]. *XUpdate* is a simple XML update language. It can be used to modify XML content by simply declaring, in an XML syntax, what changes should be made.

Next examples show the power of *XQuery* expressions. Example 1 shows an *XQuery* expression to retrieve all the MPEG-7 audiovisual segments containing media information. Example 2 shows an *XQuery* expression to retrieve all MPEG-7 person agents, whose role is ’Singer’, and the characters they play.

```
for $segment in //AudioVisualSegment
let $title:=
 $segment/CreationInformation/
 Creation/Title/text()
order by $title
return
 for $media in $segment/MediaInformation/
             MediaProfile
 let $file:=$media/MediaInstance/
             MediaLocator/MediaUri/text()
 let $type:=$media/MediaFormat/
             Content/Name/text()
 return
 <a href="{$file}"> {
   $title , " [" , $type , "]"
 } </a>
```

Example 1. *XQuery* expression for retrieving a listing of multimedia items —title and support type.

```
for $creator in
 /Mpeg7/Description/MultimediaContent
 /*/CreationInformation/Creation/Creator
where
  $creator/Role[@href="OpenDramaSinger%"]
 and
  $creator/Agent[@xsi:type="PersonType"]
order by
 $creator/Agent/Name/FamilyName
return
<agent>
{
 let $completeName:= $creator/Agent/Name
 let $name:= $completeName
             /GivenName/text()
 let $surname:= $completeName
             /FamilyName/text()
 return
  <singer>
  {
    $name, " ", $surname
  }
  </singer>
}
{
 let $completeName:= $creator/Character
 let $name:= $completeName
             /GivenName/text()
 let $surname:= $completeName
             /FamilyName/text()
 return
  <character>
  {
    $name, " ", $surname
  }
  </character>
}
</agent>
```

Example 2. *XQuery* example to display the singers and the characters they play.

Both examples reveal the potential of the *XQuery* language. When combined with XHTML, *XQuery* allows to effortlessly create web applications. Direct access to the XML data, and then formating it to be displayed by the web browser is pretty straightforward.

5. MPEG-7 WEB BROWSER ARCHITECTURE

We have implemented a search and retrieval web system, based on MPEG-7. We follow the *pull* application scenario depicted in figure 1 —a client requests audiovisual information and its associated metadata. In our system, metadata information is related to the creation and production process —including media information—, as well as the structural (temporal) decomposition of an opera.

Figure 2 depicts the whole system architecture. A web browser queries the XML:DB according to MPEG-7 descriptors. In order to access to the database, the server application transforms the MPEG-7-based query to an *XQuery* expression. Next step is to search into the database for segments matching the criteria, to process the XML results, and to return HTML documents to the browser (as well as the multimedia files retrieved from the streaming server). Finally, *eXist*[6], a native open source XML:DB, is used to manage MPEG-7 documents. *eXist* implements both *XQuery* and *XUpdate* languages.

[3] http://www.w3.org/TR/2004/WD-xpath20-20040723

[4] http://www.w3.org/XML/Query

[5] http://www.xmldatabases.org

[6] http://exist-db.org

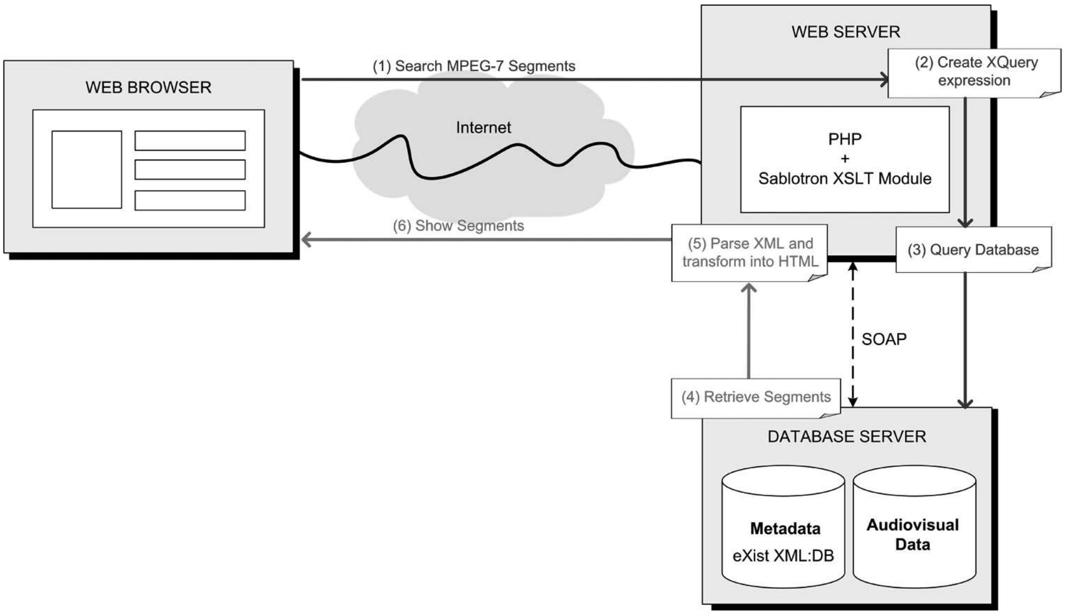

Figure 2. MPEG-7 Web Browser architecture.

6. CONCLUSIONS

MPEG-7 provides description mechanisms for multimedia content; however, existing systems based on MPEG-7 standard (specially in the audio field) are still immature.

There are several approaches to model database management of MPEG-7 descriptions. Relational or native XML:DB approach might depend on the application requeriments. Beforehand, native XML:DB seems a natural way to organise MPEG-7 descriptors and MPEG-7 DDL as it is tigthly associated to XML structure. Even though, there is still some work pending on the design of XML:DB —as well as the standarization of *XQuery* 1.0 language— to provide full funcionality for an MPEG-7 based application. Using a relational database to manage MPEG-7 can derive to an "artificial" or complex database schema, as well as an overhead of programming (to validate data integrity, for instance).

7. REFERENCES

[1] Manjunath, B. S., Salembier, P. and Sikora, T. *Introduction to MPEG 7: Multimedia Content Description Language*, Ed. Wiley, 2002.

[2] Kosch, H. *Distributed Multimedia Database Systems supported by MPEG-7 and MPEG-21*, CRC Press, 2004.

[3] XML:DB. "XML:DB Initiative for XML Databases". Available online: http://xmldb-org.sourceforge.net/index.html

[4] Jacob, M. "Managing large sound databases using MPEG-7", *Proceedings AES 25th Internationl Conference* , London, UK, 2004 July 17–19.

[5] Döller, M. and Kosch, H. "An MPEG-7 Multimedia Data Cartridge", *Proceedings of the SPIE Conference on Multimedia Computing and Networking 2003 (MMCN03)* , Santa Clara, CA, 2003 January 29–31.

[6] Westermann, U., Klas, W. "An analysis of XML database solutions for the management of MPEG-7 media descriptions", ACM Computing Surveys (CSUR), Volume 35, Issue 4. December 2003.

[7] Shanmugasundaram, J., Tufte, K., He, G., Zhang, C., DeWitt D., Naughton, J. "Relational Databases for Querying XML Documents: Limitations and Opportunities", *Proceedings of 25th International Conference on Very Large Data Bases*, Edinburgh, Scotland, UK, 1999 September 7–10.

[8] Tian, F., DeWitt, D., Chen, J., Zhang, C. "The Design and Performance Evaluation of Alternative XML Storage Strategies", *ACM Sigmod Record*, Volume 31, Number 1, March 2002.

[9] Bourret, R. "XML Database Products: Native XML Databases". Available online: http://www.rpbourret.com/xml/XMLDatabaseProds.htm

[10] Hu, G., Li, Q. "Schema Validation Applied to Native XML Databases", *International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet* , L'Aquila, Italy, 2002 July 29–August 4.