Milos: A Multimedia Content Management System*

Giuseppe Amato, Claudio Gennaro, Fausto Rabitti, Pasquale Savino

ISTI-CNR
Pisa, Italy
{giuseppe.amato, claudio.gennaro, fausto.rabitti, pasquale.savino}@isti.cnr.it
(Extended abstract)

Abstract. This paper describes the architecture of the MILOS Content Management System. MILOS supports the storage and content based retrieval of any XML document, as well as multimedia documents whose descriptions are provided by using heterogenous metadata models represented in XML. MILOS is **flexible** in the management of documents containing different types of data and content descriptions; it is **efficient** and **scalable** in the storage and content based retrieval of these documents. The paper illustrates the solutions adopted to support the management of different metadata descriptions of multimedia documents in the same repository, and it illustrates the experiments performed by using the MILOS system to archive documents belonging to three different and heterogenous collections which contain news agencies, scientific papers, and audiovideo documentaries.

1 Introduction

A large part of corporate information like office documents, legal papers, technical references, regulations, customer relationship information, etc. should be in the scope of the Content Management System (CMS). In order to effectively manage this kind of data, a CMS should be able to manage not only formatted data, but also textual data, using Information Retrieval technology, semi-structured data, typically in XML (W3C standard), mixed-mode data, like structured presentations, and multimedia data, like images and audio/video.

These requirements are especially essential for the next generation Digital Library (DL). Accessing multimedia data is becoming more and more essential for digital libraries since numeric data and text documents account for only 0.003% of the total amount of digital information produced today [10]. There is an extensive research work towards the extension of DL technology to support

* This work was partially supported by the ECD project [3], funded by the italian government, and by DELOS NoE, funded by the European Commission under FP6 (Sixth Framework Programme). We would like to thank Paolo Bolettieri, Franca Debole, Francesco Furfari, and , and Bertrand Le Saux, for their valuable contribution to the MILOS implementation.

the management of other media: images, audio, video, etc. [12] [1]. At the same time, the first multimedia DLs are becoming commercially available [2].

We have identified these requirements for a Multimedia Content Management System: (i) *Flexibility*, in structuring both multimedia object and their metadata; (ii) *Scalability* and *distribution* as fundamental design objectives of the system architecture; (iii) *Operational model* suitable to the user required functions.

Identifying the characteristics of the *operational model* is essential in order to design a system with suitable performance for the operations considered most critical by the user in the desired application settings. These operational requirements result quite different from these typical database systems.

a) It should be possible to insert new XML data and metadata without prior intervention of a database administrator. In databases, no data/metadata can be inserted if its schema has not been previously defined. b) It should be possible to define different metadata corresponding to the same multimedia basic object(s). An XML metadata object can span over many multimedia basic objects. c) The access to the content should be based on all its components, that is, the query formulation should combine conditions on formatted data/metadata attributes, textual and multimedia components (via associated features). The query results should be ranked taking into account all components in the answer. d) Searching must be very efficient. Since metadata are coded in XML, the system must efficiently support standard XML query languages such as XPath [7] and XQuery [8], with extensions for querying text and multimedia components. e) Since update operations are quite rare compared to search operations, it is not necessary to enforce a database-like transactional mechanism. Rather, a concurrency control mechanism to support the editing process of complex multimedia objects and associated metadata is necessary, based on some sort of check-out/check-in protocol. f) The system must be efficient in supporting the continuous insertion of new multimedia objects and associated metadata.

Based on these requirements, we have designed the architecture of MILOS (Multimedia dIgital Library Object Server), an XML-based Multimedia Content Management system component that can be used to build efficient, flexible and scalable multimedia digital libraries.

2 Architecture

MILOS is intended to be a general purpose content management system, which can be used to build applications dealing with any digital content. Therefore, if we make an analogy to the database field, our system is the analogous of a database management system in the domain of document intensive applications, as for instance digital libraries. It is a software tool specialized to support applications where documents, embodied in different digital media, and their metadata are efficiently and effectively handled.

Until now, many Digital Library Systems have been developed and experimented. However, their limit is that they are tailored to specific types of documents and to a specific metadata description model. This limits their applicability to different applications environments and the interoperability between different metadata description models. This is mainly due to the lack of standard general purpose basic building components, as the content management system that we propose.

2.1 Overall architecture description

MILOS mainly operates over multimedia documents and their descriptive metadata, by supporting their *storage and preservation*, their *efficient and effective retrieval*, and their *efficient and effective management*.

It has been developed by using the Web Service technology, which in many cases (e.g. .NET, EJB, CORBA, etc.) already provides very complex support for "standard" operations such as authentication, authorization management, encryption, replication, distribution, load balancing, etc. Thus, we do not further elaborate on these topics, but we will mainly concentrate the presentation to the main innovative aspects required by a general purpose content management system. Specifically it should have the capability of (1) managing arbitrary heterogeneous metadata schemas, (2) providing applications with independent views on the metadata schema actually handled, (3) managing different documents embodied in different media and stored with different strategies.

Point (1) is important since different organizations may have already their own metadata schemas, and hardly want to modify them to be compatible with a specific system. In addition, there are also cases where the same application might need to deal with different metadata at the same time.

Point (2) makes it possible that the metadata schema seen by the users of an application is different from the metadata schemas actually stored in the repository of the content management system. As an example, we can suppose that the user wants to interact with a digital library using Dublin Core, independently from the specific metadata actually managed by the content management system. This feature is especially useful in case of heterogeneous metadata available at the same time in the metadata repository.

Point (3) requires that no assumptions should be taken on the types of media and encoding used to represent documents, and especially on the specific strategy used to store them. This allows applications to be unaware of the technical details related to multimedia document management. For instance, textual documents can be stored in the file system and served to the users using a normal web server. However, video documents might need to be stored in a video server that uses various storage devices (e.g. digital tapes, optical disks, etc.

According to the issues illustrated above, the architecture of the MILOS content management system is composed of three main components as depicted in Figure 1: the Metadata Storage and Retrieval (MSR) component, the Multi Media Server (MMS) component, and the Repository Server Logic (RSL) component. All these components are implemented as Web Services and interact by using SOAP.

The MSR manages the metadata of the digital library. It relies on our technology for native XML databases, and guarantees the requirements illustrated

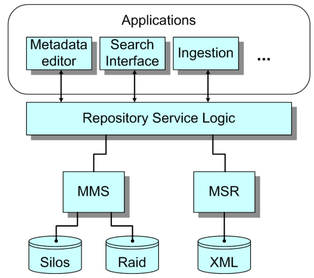

Fig. 1. General Architecture of MILOS

at point (1) above. Our native XML database, supports the management of standard XML query languages such as XQuery and XPath, but also offers advanced search and indexing functionality on arbitrary XML documents. Thus we implemented an XML database that supports high performance search and retrieval on heavily structured XML documents [4,13], but also provides full text search [11], automatic classification [9], and feature similarity search [6] functionality. So, for instance, the tag name `<abstract>` can be associated with a full text index and with an automatic text classifier. On the other hand, the MPEG-7 `<VisualDescriptor>` tag might be associated with a similarity search index structure and with an automatic visual content classifier. Performance of search operations on our XML database is much higher than those obtained by transforming the XML structures in relational tables and using a traditional relational database.

The MMS manages the documents used by the application. All documents are associated with a unique identifier independent from the strategy adopted to store and to serve the document. MMS guarantees the requirements of point (3) above. The task of the MMS is to make the programmers of document intensive applications free of all storage strategy related issues. When an application requires to retrieve a document it will send the request to the MMS that will act as a gateway to the actual repository that maintains the document. A similar approach is used when a document has to be inserted in the repository. The system administrator can define rules, based on distinctions among mime types, to tell the MMS how to deal with documents being inserted. Therefore, for example, an MPEG-2 video document will be stored in a tape, while an image will be stored in an array of disks.

The RSL provides the application developers with a uniform and integrated way for accessing MMS and MRS. In addition, it supports the mapping of different metadata schemas as required at point (2) above. The RSL also manages the accesses to the underlying databases. In particular, it manages query processing by integrating unaligned information stored in the two databases. It performs reconciliation of retrieved data through ranking. When a new XML schema is introduced, the designer must configure the RSL in order to specify which metadata fields are important for the search, and how to map the metadata fields, for the retrieval of the documents.

3 Metadata Mapping

A significant features of MILOS is the possibility to support different metadata schemas. This task is accomplished by using a simple description file, which defines a schema mapping used by the RSL search service for retrieving the XML documents. The main purpose of this mapping it to define rules for translating application requests into XQuery, so that the applications do not have to use XQuery directly. Moreover, this mechanism allows the RSL to translate abstract names of fields (such as Title, Author, etc.) used by the applications, into XPath of the XML documents to retrieve, without the need of knowing the specific schema model adopted.

The configuration file, called *metadata.properties*, is composed of a set of entries, each one corresponding to a mapping directive. In particular, each entry specifies how to translate an abstract name of a metadata field into the corresponding XML path for addressing the element, which actually contains the content of the field. A generic entry of the metadata.properties file has the following structure:

metadataType[Entity][.Name] = <XML Path Expression>*

- The *metadataType* field identifies the metadata model we would like to refer to, e.g., Dublin Core, SCORM, MPEG-7, etc.
- The *Entity* field is the name of the entity of the model we would like to refer to e.g., Book, Manifest, Lom, etc. Typically, it is the root of the XML document that represents the instance of the entity. If this field is left empty, it means that we refer to all the entity instances of the repository.
- The *Name* field allows to specify the name of the attribute to search for, e.g., Title, Author, etc. If empty it means that we refer to all the attributes, in a given entity.

As an example, let us suppose the repository contains a set of SCORM objects. We can have the following simple metadata.properties file composed of only two entries:

scorm_lom.title = lom/general/title/langstring
scorm_lom.dc.creator.title = lom/general/title/langstring

They specify that both the abstract metadata fields ‘title’ and ‘dc.creator.title’ of the conceptual entity *lom* of the SCORM model, must be addressed by means of the XPath string lom/general/title/langstring. Note that, the *title* XML element does not contain the title text of the document, but the element *langstring*, which in turn contains the text.

Let us now explain how the directives contained in the configuration file are used by the RSL to map the metadata fields of the query to those used in the documents. As an example, let us consider a specific method of the search service of RSL to retrieve the metadata by means of an exact match search on the document or metadata fields.

findExactMatch(**string** MetadaType, **vector of string** field, **vector of string** value, **string** returnField),

This method searches for a set of documents or metadata, identified by the *MetadaType* parameter (for instance, ECHO, MPEG-7, SCORM, etc.). The *field* parameter, is a vector of abstract names of metadata fields to search for. The *value* parameter specifies the values that the fields must match (the different fields are searched by using the boolean connective **AND**). Finally, the *returnField* parameter specifies the field to be returned by the matching documents.

In particular, the method translates the call into XQuery as explained in the following:

1. for each triple <*MetadaType, value*$_i$*, field*$_i$> specified by the *findExactMacth*, RSL searches in the *metadata.properties* the XPath strings Q_i.
2. for the pair <*MetadaType, returnField*>, specified by the *findExactMacth*, the RSL searches in the *metadata.properties* the matching XPath string R.
3. finally, the XPath strings Q_i and R are combined in order to form the resulting XQuery to submit to the MSR as in the following:

for *$a* **in** *MetadaType*
where \$a$Q_1 = value_1$ **and** \$a$Q_2 = value_2$ **and** ... **and** \$a$Q_n = value_n$
return *$aR*

4 Field Trials

In order to verify and demonstrate the flexibility of MILOS in the management of different metadata models, we imported four different data-set, containing documents and metadata of a very different nature:

– the Reuters repository which contains 800,000 XML document from the Reuters collection (2.6GB) composed of news agencies,
– the ACM Sigmod Record metadata data-set, composed of 46 XML files (1MB),
– the DBLP data-set, composed of an XML document of 187MB, and
– the ECHO [1] audio/video data-set, composed of 50 hours of video documentaries.

For each of these data sets we provided a bulk import tool and we implemented a user interface to express queries and to visualize retrieved items.

Actually, ECHO is the most significant example of the capability of MILOS to support the management of different metadata schemas, even the most complex

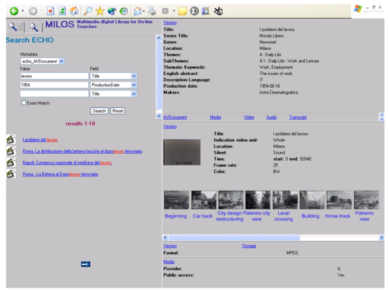

Fig. 2. The ECHO retrieval interface implemented in MILOS

ones. Indeed, the metadata model adopted in ECHO, based on IFLA/FRBR, is rather complex and strongly structured. It is used for representing the audio-visual contents of the archive. Obviously, XML is used as an internal representation of video metadata, as well as an interchange format with other systems. The schema of the ECHO system is composed of seven entities.

We have imported all XML metadata of the ECHO collection, which comprise, the description in English and in the original language of the video (Title, Producer, year, etc.), the boundaries of the scene detected (associated with a textual description), the audio segmentation (in noise, music, speech, etc.), and the Speech Transcripts. The multimedia collection comprises, the mpeg video and keyframes of the detected scenes. The collection is composed of about 8,000 documents described by 43,000 XML files. The multimedia repository consists of about 21GB of mpeg files.

Figure 2 shows the search interface of the ECHO collection. The interface provides two distinct search interfaces, a fielded search (shown in figure) and a full text search. The former allows a user to retrieve documents by means of their fields (Title, Genre, ProductionDate, etc.). The latter allows one to retrieve the documents by means of a standard full text search on fields such as Title, Description, Transcript, etc. Selecting one of the retrieved items, it is possible to navigate inside its metadata structure; for example, it is possible to examine the keyframes and the title associated with the scene detected in the video.

5 Conclusion

This paper described the architecture of the MILOS Content Management System and the solutions adopted to obtain a system that is flexible in the management of documents with different types of content and descriptions, and that is efficient and scalable in the storage and content based retrieval of these documents. In particular, we described the approach adopted to support the management of different metadata descriptions of multimedia documents in the same repository. This goes towards the solution of the challenging problems of interoperability among different metadata descriptions. The proposed solution, based on the use of a mapping mechanism between the metadata fields of the different models, has been practically experimented by using the MILOS system to archive documents belonging to four different and heterogenous collections which contain news agencies, scientific papers, and audiovideo documentaries. The archiving of these documents was straightforward and it only required the creation of the mapping file and the development of the user interfaces to archive and search the documents.

Evolutions of this activity are foreseen in several directions: on one side we are working to improve the retrieval capabilities of the Metadata Storage and Retrieval component in order to support similarity search of multimedia documents; on the other side, we are working with partners of the ECD [3] project on the automatization of the mapping between different metadata schemas, by using thesaurus and cross-language vocabularies [5].

References

1. European chronicles on-line. Available at http://pc-erato2.iei.pi.cnr.it/echo/.
2. Virage web site. http://www.virage.com/.
3. ECD - Enhanced Content Delivery, 2002. http://ecd.isti.cnr.it/.
4. G. Amato, F. Debole, F. Rabitti, and P. Zezula. YAPI: Yet another path index for XML searching. In *ECDL 2003, 7th European Conference on Research and Advanced Technology for Digital Libraries, Trondheim, Norway, August 17-22, 2003*, 2003.
5. D. Beneventano, S. Bergamaschi, S. Castano, V. D. Antonellis, A. Ferrara, F. Guerra, F. Mandreoli, G. C. Ornetti, and M. Vincini. Semantic integration and query optimization of heterogeneous data sources. In *OOIS Workshops*, pages 154–165, 2002.
6. C. Böhm, S. Berchtold, and D. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3):322–373, September 2001.
7. W. W. W. Consortium. XML path language (XPath), version 1.0, W3C. Recommendation, November 1999.
8. W. W. W. Consortium. XQuery 1.0: An XML query language. W3C Working Draft, November 2002. http://www.w3.org/TR/xquery.
9. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambidge University Press, 2000.
10. DELOS. Digital libraries: Future directions for a european research programme. Report, June 2001. http://delos-noe.iei.pi.cnr.it/activities/researchforum/Brainstorming/brainstorming-report.pdf.
11. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.
12. H. Wactlar. The informedia digital video library. Available at http://www.informedia.cs.cmu.edu/.
13. P. Zezula, G. Amato, F. Debole, and F. Rabitti. Tree signatures for xml querying and navigation. In Z. Bellahsene, A. B. Chaudhri, E. Rahm, M. Rys, and R. Unland, editors, *Database and XML Technologies, First International XML Database Symposium, XSym 2003*, volume 2824 of *Lecture Notes in Computer Science*, pages 149–163. Springer, 2003.