Design and Implementation of XML-Based Digital Video Library System

Jacky C.K. Ma and Michael R. Lyu
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong

ABSTRACT
In this paper, we describe the design and implementation of an XML-Based Digital Video Library (XDVL) System. The system includes automatic processes from video creation through video delivery, which involves storing, searching, indexing and retrieval of video contents. It also allows users to search for videos based on various kinds of information using a multi-lingual approach. We adopt a multi-tier model of four components in our system, namely Video Server, Indexing Server, Query Server and Client Application, resulting in a more extensible and usable system. For data management, we employed XML as the enabling technology for metadata structure repository and component's messaging interface. With XDVL, we can maintain the compatibility between different digital library developers, and its open-standard solution will contribute to the popularity of digital video libraries for their creation, integration and subscription.

Keywords
digital video library, data management, XML.

1. INTRODUCTION

Advances in media processing technology and growth of the Internet open up a wide range of application areas where video is heavily used. Digital Video Library (DVL) system allows users to search among a huge collection of digital video by their contents.

A DVL system includes automatic processes from video creation through video delivery, and provides other additional functionality that makes it a powerful information resource. Under a multi-tier model that consists of Video Server, Indexing Server, Query Server, and Client Application as the major components, we can increase the usability, extensibility and reliability by clustering and redundancy, possibly with a heterogeneous set of servers and networks. To facilitate this, we employ XML as the enabling technology to identify, describe and manage the multimedia content, thus naming our system as XML-based Digital Video Library (XDVL). XML maintains the interoperability between different developers' implementations by providing the standardized data management framework. Consequently, we can seamlessly introduce new or enhanced services into existing system. This encourages third-parties developers to develop various value-added components for the DVL systems, and allow many innovative ideas to be realized.

2. SYSTEM ARCHITECTURE

From the view of functional division, the XDVL system may be divided into four primary components: Video Server, Indexing Server, Query Server and Client Applications. While this is a conceptual division, the implementation design may vary from single DVL-Workstation to distributed system over the Internet. The general workflow of a DVL system is shown in Figure 1. The first process is digitizing the raw video and storing them in the Video Server. Then the Indexing Server will process the videos to extract features and information to build the indexing structure. The above processes are offline and can be carried out in batch mode. On the other hand, user query and video playback are online processes that the respond time is an important aspect for system design.

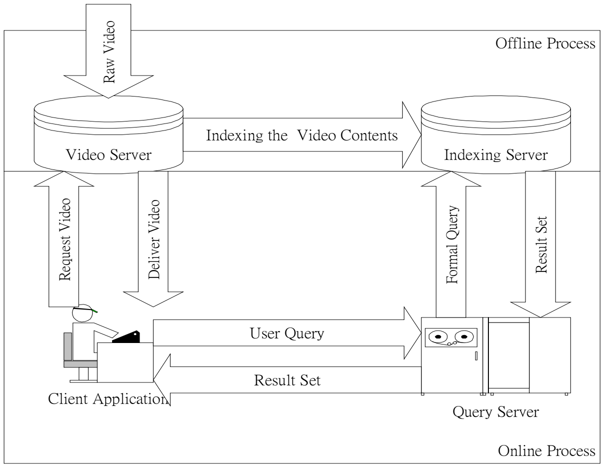

Figure 1. Overview of a DVL System

In its simplest form, all of the system components remain in the same computer appearing as one piece of program. Digital video is stored as MPEG files in the hard-drive or CD-ROM, indexing structure is kept in a desktop database engine, while user query and video playback may be featured in a single program. This simple application models the CD-ROM Encyclopedias. By distributing tasks to different servers, we construct a Digital Video Library Network as shown in Figure 2, where each server gather information from multiple sources and serve multiple clients to maximize system capacity and availability. Moreover, through the use of middleware, DVL may work with legacy systems such as search engines, meta-search engines and web browsers, and appears as part of an integrated system in the Internet.

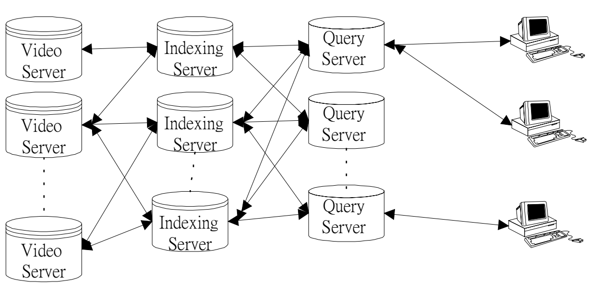

Figure 2. Digital Video Library Network

2.1 Video Server

Video Server specializes in capturing, storing and delivering video contents. Video sources may come in various physical forms such as VHS tapes, laser disc, or TV broadcasting, and the video contents may also fall into different categories such as interviews, lectures, News, MTVs, movies and etc. Consequently, the processing techniques, segment lengths and quality of video delivery will vary accordingly. For this technology requirement and for copyright issues, it is likely that video content providers establish and manage their own video servers.

Storing of video requires the input video signal to be segment smaller pieces. To accomplish this, low-level features (e.g. color composition change for scene breaks, or blank screen with silence during scene transitions) and content-specific features (e.g. specific sound pieces or screen changes for news reports and commercials) will be captured by a matching technique. For video delivery, two technologies will be employed: Quality of Services(QoS) and streaming video delivery. These two technologies can greatly increase the perceived services by the users. Since there are lots proposed reference implementation currently, we are not going to cover them here.

2.2 Indexing Server

Indexing Server stores the major indexing structure of video for query and retrieval. Information to be extracted and indexed mainly falls into three types: raw textual information, physical information and semantic information. The extracted information will later be queried on a text-basis or content-basis.

Raw textual information includes all kind of textual data associated with a video that is not extracted from the video. This applies to information provided by Video Server, annotations, summaries and viewers' feedback. Textual information is the simplest form among three and do not involve processing of the video, and obviously they will only be queried on text-basis. Physical information, on the other hand, involves analysis of the video, and may be queried on both text-basis and content-basis. Features like color, texture, shape, motion, and spatiotemporal structures of video scenes are typical examples. To extract these features, videos are decomposed into shots of consistent background scene; the foreground video objects are extracted by comparing successive frames within a shot. Features will be stored as color histogram, frequency histogram, shape and motion descriptors that can be use in content-based query process.

Semantic information is the most difficult to extract as semantic analysis needs the help of real world knowledge, which is difficult to model in terms of complexity as well as capacity. However, there are still degrees of achievement in areas like speech recognition with the use of sound wave dictionary, vocabulary dictionary, and language's grammar. In fact, automatically generated script is very useful for content-based searching. For visual features, the underlying principle is quite the same: visual features of video objects are extracted and used to "look up" for the possible semantic objects (i.e. dictionary look up). By calculating the co-occurrence probability of all video objects in a shot, we can determine which combination of objects is most likely to happen (i.e. grammar checking). However, there is no existing video object dictionary and video object grammar, we can only create one by using labeled multimedia training data. By using a hidden Markov Model (HMM), the determining features of video objects and the co-occurrence probability of different video objects can be estimated. However, the training data should be constrained within a limited scope to generate reasonable results; practical usage of video object understanding is still a long way to go.

2.3 Query Server

Query Server accepts user queries from Client Applications, constructs queries to Indexing Servers, collects and ranks the results, and finally sends them back to the Client Application. More than a simple search engine, the Query Server is equipped with knowledge about the Indexing Servers that registered on it, therefore queries can be send to a selected group of Indexing Servers that contains relevant information. Duplicated entries will be merged and results will be clustered by categories to give a concise and easy-navigating report.

There are various information retrieval models including Boolean Model, Vector Model and Probabilistic Model that are well defined in textual domain, but now they are extended to handle content-based in Query Server. With content-based query, user can choose a color from the color panel and draw a shape, than apply an AND query; or he may select frames from several videos, and query for videos that have similar video objects. The flexible query method will enhance the DVL exploration experience.

As a portal subsystem, personal customization is one of the many value-added services. The portal may filter and deliver the news interesting to the user everyday and notify the user whenever a new movie performed by a particular movie star is available. For the most intelligent portal, the behavior and interest pattern of the user will be learned, and suggest potential video whenever appropriate.

2.4 Client Application

Client Application is the presentation subsystem of XDVL. In essence, it only handles query submission, query result presentation and video playback. In our earlier client prototype (Figure 3), we adopt a Java Applet thin client that allows user to query by title, keyword or script. After submitting the query, the returned results are shown in thumbnails with their titles and short description for user to locate their interested video faster and easier. When the user clicks on a thumbnail, the selected video will start playing in stream with synchronous running highlights on the script.

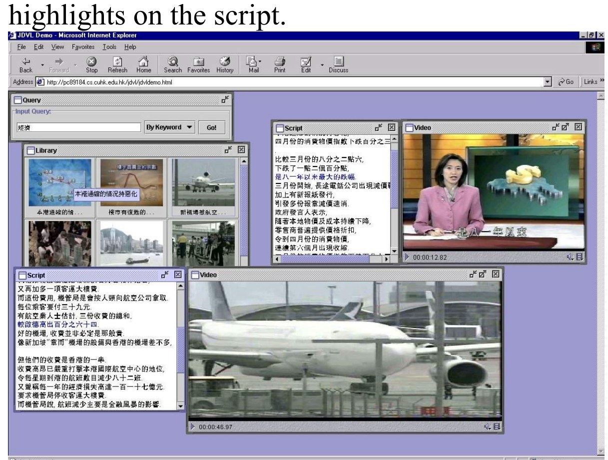

Figure 3. A Java Applet Client

As more functions are provided in other DVL modules, the Client Application may be enhanced accordingly. Say, if image query is supported, the Client Application should allow user to import a picture and provide drawing tools for sketching the query; if sound query is supported, the Client Application should allow the user to input with the microphone. Besides, a Client Application may provide additional features such as video manipulation and multimedia document browsing (instead of video) that provides extra value to the user.

3. DATA MANAGEMENT

To facilitate the system architecture mentioned, and for the sake of flexibility and interoperability, we employ XML for both data storage and data exchange. The semi-structure allows some degree of irregularities, in terms increase flexibility in the meta-data framework and capable to meet various system requirements. As a medium for system integration, the descriptive features must be understood by the applications. Attributes that we store in XML format are:

- Production Information: information from the producer (title, date, keywords, producer, performer, language, etc.)
- Usage Information: transactional information (copyright and cost of usage)
- Processing Information: information for process controls and parameters for the defining the quality of service (file size, bit-rate, resolution, codecs, etc.)
- Physical Information: low level video features (color, shape, segments)
- Conceptual Information: high level video features (people, place, events, narratives)
- Category: semi-auto video classification (news, commercials, MTV, conferences, etc.)
- References: pointers to other resources.
- Summaries and Feedback: comments by professionals and feedback from viewers.

XML also benefits from its simple plain text form, which allow it to be transmitted through most type of networks. On the other hand, the plain text form also address a need to reference binary objects in XML, and to secure the text data by encryption, such that it will be capable to handle various kind of requirement in data management.

4. CONCLUSION AND FUTURE WORK

In this paper, we introduce the system architecture of XDVL and describe its components. We also describe how to manage the data by using XML as an enabling technology. Currently, most technologies associated with DVL are developed for English-only system. By using XML technology, the problem in character coding is solved, but we still need more effort on the multi-lingual issues. Besides, we will try to exploit the possibilities in XML data management, and target to build a model that provides generic XDVL functions in the future.

5. ACKNOWLEDGEMENT

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. CUHK4193/00E).

6. REFERENCES

[1] Jennifer Widom. *Data Management for XML Research Directions*. IEEE Data Engineering Bulletin, Special Issue on XML, 22(3):44-52, September 1999.

[2] Michael Christel and David Martin. *Information Visualization within a Digital Video Library*. Journal of Intelligent Information System, June 1998.

[3] Surjanto, B.; Ritter, N.; Loeser, H. *XML content management based on object-relational database technology*. Proceedings of the First International Conference on Web Information Systems Engineering, Volume: 1, 2000. Page(s): 70 -79 vol.1.

[4] Szentivanyi, G. *The role of XML in generic and distributed multimedia management*. Proceedings. IEEE 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (WET ICE '99), 1999. Page(s): 317 –318.

[5] Wactlar, H. *Informedia - Search and Summarization in the Video Medium*. Proceedings of Imagina 2000 Conference, Monaco, January 31 - February 2, 2000.