Architecture & Data Management

of

XML-Based Digital Video Library

Master of Philosophy

Research Project Second-Semester Report

Supervisor

Professor Michael Lyu

Prepared by

Ma Chak Kei (00315340)

Department of Computer Science and Engineering

The Chinese University of Hong Kong

Abstract

Tremendous growth of the Internet population creates a large demand on new applications, and advances in Internet technologies make it feasible to develop new exciting application base on video and broadband network. One of the hottest topics nowadays is the Digital Video Library Systems.

In this term paper, my work in last year will be described. This includes an overview of XML-Based Digital Video Library, which I have done in the first semester; and also a description of Data Management in Digital Video Library using XML, which I focused in the second semester. After these two sections, the future research plan will be given. The major interest of the research plan is on exploring the XML schemas, to study the usage of multiple DTDs, and to introduce more complicated datatypes than those in the current DTD specification.

Contents

Introduction

Advances in multimedia technology and Internet technology open up the area of Digital Video Library (DVL) system which address the need of entertainment, infotainment, education, education, cultural services, shopping, professional services, etc. Currently, many elements exist as infrastructure for multimedia delivery and consumption. Access to information through network is convenient. However, there is no 'big picture' to describe how these elements, either in existence or under development, relate to each other. The aim for XML-Based Digital Video Library is to have a generic model of DVL that various elements fit together elegantly.

A DVL system includes automatic processes from content creation through video delivery and consumption, and provides other additional functionality that makes it a powerful information resource. Under a modular framework, we can figure out what are the major subsystems, and ready to increase the reliability and performance by redundancy or clustering, possibly with a heterogeneous set of servers and networks. To facilitate this, the multimedia content has to be identified, described and managed in a good way to ensure interoperability between different developers.

XML, as its openness and simplicity, is a promising solution for multimedia content description and for component messaging interface in the multimedia framework. In fact, MPEP-7 and MPEG-21 are proposed standards that try to address the need of multimedia applications and, one of the most promising solutions is to implement the standards as XML schemas.

However, as there are several insufficiencies in the classical Data Type Definition that limit the expressive power of schemas. As stated in the proposed research plan, we will look into ways that enhance the DTD to release the power of XML.

XML-Based Digital Video Library

Introduction

Advances in media processing technology and growth of the Internet opens up a wide range of application areas including entertainment, infotainment, education, cultural services, shopping, professional services, etc. Digital Video Library (DVL) system is the technology to address these purposes. In a DVL system, users may search among a huge collection of digital video files for almost any topics and categories. Movies, MTV, news records and conferences are just a few examples of those available video archives.

A DVL system includes automatic processes from video creation through video delivery, and provides other additional functionality that makes it a powerful information resource. In order to achieve high usability, extensibility and reliability, we adopt a multi-tier model that consists of Video Server, Indexing Server, Query Server, and Client Application as the major components. Each kind of the components forms a subsystem of the DVL system with specific functionality and can be clustered to provide higher capacity and better fault tolerance. For the metadata structures and component interfaces, we employ XML as the enabling technology, thus naming our system as XML-based Digital Video Library (XDVL). XML provides a standardized framework for data sharing and action signaling among components and maintains the interoperability between different developers' implementation. This encourages development of a hybrid system on different software and hardware platforms, and provides abundant heterogeneous features. While users practice open standard to favor sharing of resources, they are also aware of the need to code and deliver proprietary information securely to protect innovative and original idea; this allows companies to differentiate their customers by providing different class services with different level

of information details.

Besides, combining the power of both component-base system and XML interface, we can seamlessly introduce new or enhanced services into existing system. This encourages third-parties developers to devote into development of components that provide value-added services. This is especially essential in a field like DVL system that the related technology is undergoing rapid development and new services will likely to be emerged anytime.

System Architecture

Our multi-tier XDVL system consists of four primary components: Video Server, Indexing Server, Query Server and Client Applications. Based on this setting, the system may be modeled as a single DVL-Workstation or as a distributed system over the Internet. The general workflow of a DVL system is shown in Figure 1. The process is started by the input of the raw video into the Video Server: this step involves digitizing and storing of video. Then the videos are processed by Indexing Server to extract features and information that can help searching in a later stage and to build indexing structure. The above processes are called offline processes since they are invisible to the end-users and can be carried out in a batch mode at anytime. On the other hand, there are online processes whose response time affects the users directly. This includes processes involved in query action and video playback.

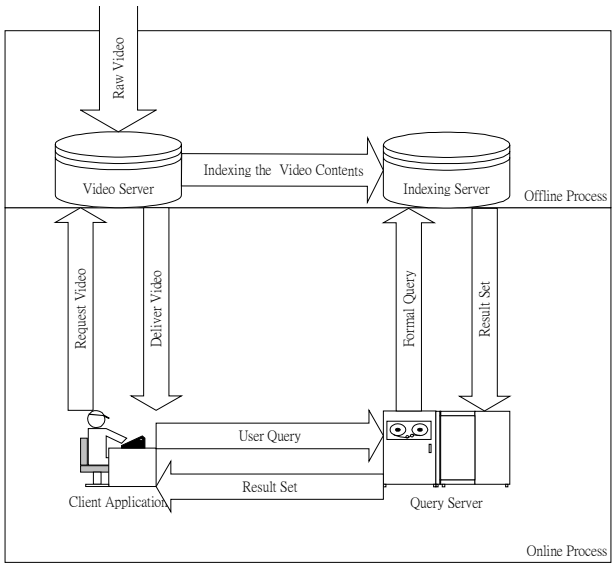

Figure 1. Overview of a DVL System

It should be understand that the four components are functionally decomposed in concept, but in implementation they may be further divided or combined. For example, a Client Application may combine with Query Server (i.e., perform the function of Query Server by itself), or a Query Server may be divided into query preprocessing and query optimization parts. In the simplest form, all of the system components remain in the same computer appearing as one piece of program. Digital videos are stored as MPEG files in the hard-drive or CD-ROM, their indexing structure is kept in a desktop database engine, and user query and video playback functions may be featured in a single program. In this case, video files and indexes are prepared before the system is deployed, but the simplicity of this scenario still demonstrates a number of useful applications like encyclopedia on CD-ROM and info-kiosk in museums.

By distributing tasks to different servers, we can construct a Digital Video Library Network as shown in figure 2, where each server may gather information from multiple sources and serve multiple clients as well. Thus video resources can be shared among a large server group, maximizing the system capacity and availability. Moreover, through the use of a specialized server as “middleware”, XDVL may even work with legacy systems such as search engines, meta-search

engines and web browsers, and may appear as part of an integrated system in the Internet.

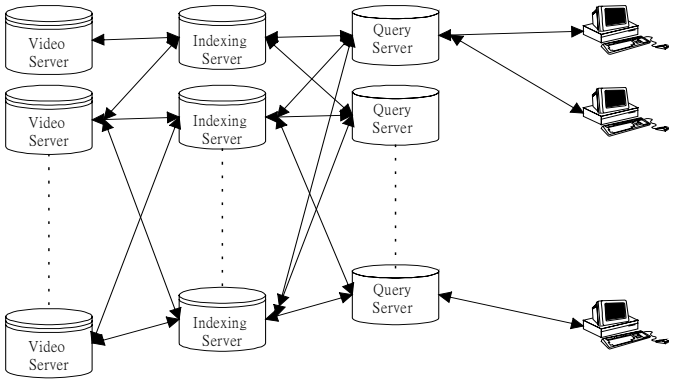

Figure 2. Digital Video Library Network

Video Server

Video Server forms the storage subsystem of XDVL and specializes in capturing, storing and delivering video contents. Video sources may come in various physical forms such as VHS tapes, laser disc, or air broadcasting. Since they are not always in digital form, conversion from source video into digital format for storage and delivery is needed. This conversion may require special hardware. The video contents may fall into different categories such as interviews, lectures, News, MTVs, movies, etc. In addition, the processing techniques, segment lengths and quality of video delivery may also vary. Consequently, a dedicated video server cannot handle all kinds of video sources. For this technology requirement and also for copyright issues, it is likely that some video content providers may establish their own video servers and charge for their video delivery services.

For storing of videos in digital equipment, it is necessary to segment the video input into smaller pieces for easier management. Several different methods are used on video segmentation. First, we use low-level features to segment the video; for example, the change of scene usually involves a

short period of blank screen together with silence, which is quite easy to detect. The differences between successive frames are used to identify scene changes in more complicated scenarios. Second, besides the low-level features, we make use of some content-specific features such as the break between news report and commercials, or the shout of “GOAL!” in a soccer match. These features can be detected using image or sound matching algorithms and are not complicated as image understanding and speech recognition, but the Video Server is required to have the knowledge about the kind of video it is processing as these features may not be universal to all kinds of videos.

As video delivery is an online process, its performance will directly affect the quality of service perceived by the end-user: therefore, it is important to maximize both the throughput and the latency of the system. The major constraint in this issue is the computing power and the network bandwidth, but employing load-balancing and multicasting can maximize the system utility in a given hardware configuration. On the other hand, common connection-oriented protocols like HTTP and FTP will create a large workload and overhead to the system. Consequently, they should be the least preference choices. Instead, we employ streaming protocols designed for real-time media, these protocols exploit fault-tolerance feature for media data stream and allow lost of information to some extent.

Indexing Server

Indexing Server forms the meta-data subsystem of XDVL. It stores the major indexing structure of video for query and retrieval. Basically, it makes use of the information provided by the video server such as the title or video category (news, commercials, etc) to help the extraction of video features. 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 either text-basis or content-basis.

Raw textual information includes all kinds of textual data associated with a video that is immediately available. This applies to the information provided by Video Server, annotations, summaries and viewers' feedbacks. Textual information is the simplest form among extracted information. This is because it does not involve extra process of the video, and obviously it will only be queried in a text-basis nature. 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 spatio-temporal structures of video scenes are examples of physical information. To extract these physical features, video sequences are first decomposed into separate shots such that each shot has a consistent background scene. The foreground video objects is then determined by comparing successive frames within the shot and they will be analyzed to extract their physical features. Features will be stored as color histogram, frequency histogram, shape and motion description as well as representative image, which can be used in content-based query process. Although these features are coded in human un-readable forms, text-based query is still possible through the use of association dictionary that translates human language into these feature patterns. For example, "red" will be mapped to a color histogram of red color, "ball" will be mapped to a circular shape description, etc.

Semantic information is the most difficult to extract and index among the three types of information. This is because extraction of semantic features needs the help of real-world knowledge, which is difficult to model in the computer system. However, there are still some degrees of achievement. Speech recognition is one of the most successful parts in semantic feature extraction. The recognition process involves using a sound wave dictionary, a vocabulary dictionary, and rules for the language's grammar. In fact, the script of video is generated by speech recognition tools automatically. Despite the accuracy of recognition never reaches 100%, the extracted information is still extremely useful. Although the complexity of visual features is dramatically higher than that of

speech recognition, the underlying principle is quite the same. Visual features of video objects are first extracted as described in the previous paragraph, then we use the attributes to “look up” for the possible objects (i.e., dictionary look-up). By calculating the probability of co-occurrence of all video objects in the same shot, we can determine which combination of objects is most likely to happen (i.e. grammar checking). What makes the difference between speech recognition and video object recognition is that we have the dictionaries and grammar for speech but do not have a general video object dictionary with well-defined video object grammar. This can be partly solved by using labeled multimedia training data. We can segment the video, extract the objects and then label them accordingly. By using a hidden Markov Model (HMM), we investigate what combinations of features can determine a video object, and also the probability of co-occurrence for different video objects. However, the training data should be constrained within a limited scope in order to generate reasonable results.

Due to the difference in requirement and processing techniques, each Indexing Server may only serve to extract some of the video features. Advanced Indexing Server may be able to extract complicated structural information and conceptual information, such as video script, video caption, company logo, human face and even auto-summary of the video narratives. It is even possible to include other multimedia links (references to/by documents, websites, etc.) and audience feedback as an indicator for degree of interest, and to provide a categorization structure. Since each Indexing Server only indexes a limited number of video features, it is useful for a video to be indexed by multiple Indexing Servers. In other words, each Video Server will subscribe to multiple Indexing Servers, and vice versa.

Indexing Server, in conclusion, is a front-end for Video Server. Only authenticated Indexing Server can make use of information provided by Video Server to help the indexing process. This in turns brings a benefit that the video object would not be transferred to many servers for processing like the case of web crawlers and web search engines. Consequently, a remarkable amount of

computing power and network traffics can be saved. Since XML is used to maintain interoperability between the components, the implementation of indexing structure inside the Indexing Server does not matter as long as it provides a consistent XML interface.

Query Server

Query Server forms the information portal subsystem of XDVL. It accepts user queries from Client Application, constructs optimized queries and sends them to Indexing Servers. Then it collects the results, ranks them and sends them back to Client Application. The role of Query Server is analog to the search engines in the World Wide Web. Query Server is more than a simple search engine. With the benefit of XML, Query Server is equipped with the knowledge about the Indexing Servers which have registered on it. Therefore, queries can be sent to a selected group of Indexing Servers depending on a category. For example, when a user query on a movie, the query will only sent to those Indexing Servers containing relevant information. Moreover, duplicated entries can be removed easily with the help of resource location indicator in XML messages and the information given in different Indexing Server can be combined. Results will also be clustered according to their categories and keywords for a easy navigation.

Besides simple keyword matching, Query Server also supports various query models including Boolean Model, Vector Model and Probabilistic Model. These powerful search mechanisms are well-defined in the textual domain with traditional information retrieval theory, but the concept is modified and extended to handle content-based queries as well. With content-based query, users can choose a color from the color panel and draw a shape, than apply an AND query; or they may select a key-frame from a video, and query for videos that have similar video objects. The flexible query method will allow the users to easily locate the video they want.

Despite of the power of searching methods offered, there are casual video viewers who just want to

explore a video library for something new to them and do not have particular keywords in mind for searching. Learning from the web search engines, we offer a category directory for the video collections. Contrast to the human-intensive classification of webpages, the video category directory can be built automatically with the help of cues in the XML data.

As a portal subsystem, personal customization is one of the many value-added services. The portal may filter and deliver the news interesting to a 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 the portal can offer video of high potential interest to the user actively.

User Applications

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

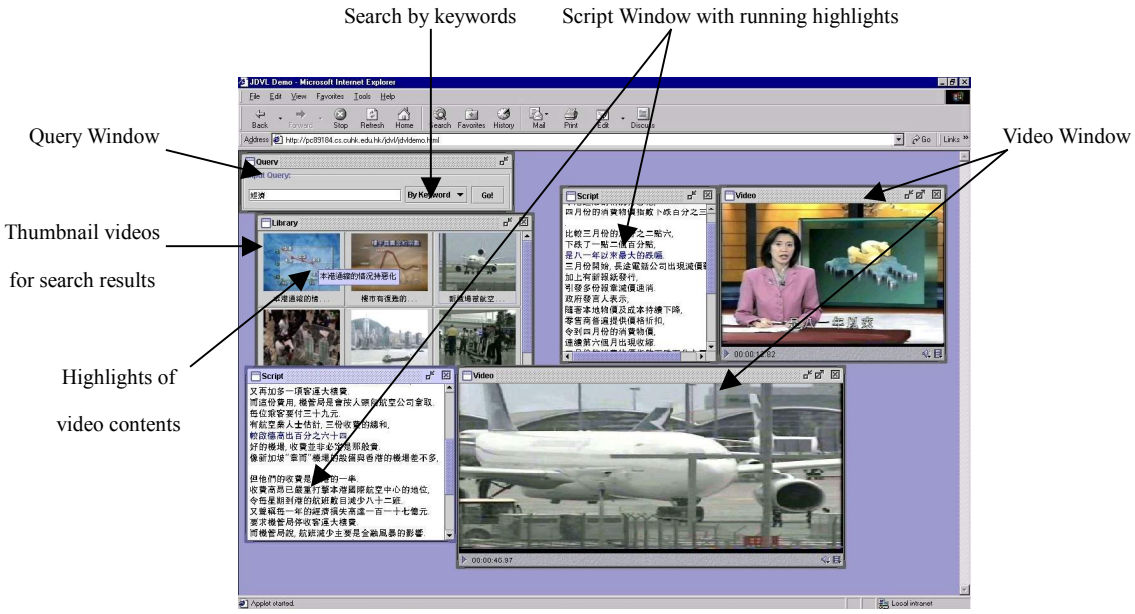

Figure 3. A Java Applet Client

To maximize the power of XDVL system, the design of Client Application must address the features provided by Query Server. If Query Server supports image query, Client Application should allow the user to import a picture in local hard disk and provide drawing tools for user to sketch the query. If Query Server supports sound query, Client Application should allow the user to use microphone as an input method. Besides, Client Application may provide other complicated features that are independent to other parts of XDVL system such as video manipulation and multimedia document browsing. Beyond the visible features, there are some invisible but important issues such as video decoder, QoS negotiation, and video caching. These issues directly affect the stability and performance of Client Application and are as important as the visible features.

Data Management

Data management is an important issue in multimedia applications. Actually, identifying and managing media content is not limited to the database retrieval process in DVL, but extends to areas like broadcast channel selection, multimedia editing, and multimedia directory services. A good data management schema enables applications to have “quality access” to content, and implies good storage solutions, high-performance content identification, highly automated system integration, and personalized filtering, searching and retrieval.

To facilitate the system architecture mentioned in the previous chapter, we employ XML for both data storage and data exchange. XML is capable to manage the data as its semi-structured data will brings more flexibility to the system while maintains the interoperability.

XML also benefits from its simple plain text format, which allow it to be transmitted through most type of networks easily. On the other hand, the plain text format raises another issues: a need to reference or include binary objects in XML, and to secure the XML data by encryption to protect the proprietary rights. When these issues are properly solved, it will be capable to handle various requirement in different kinds of systems.

As a medium for system integration, the descriptive features must be understood by the applications. They may be different for different user domains and different applications. This implies that the same material can be described using different types of features, tuned to the area of application. These features can mainly classified as Video Description and System Meta-Data, and they will also be covered in the following sections.

MPEG-7: Multimedia Content Description Interface

Accessing audio and video used to be a simple matter because of the simplicity of the access mechanisms and because of the poverty of the sources. But as audiovisual information is becoming available in digital form, in digital archives, on the World Wide Web, in broadcast data streams and in personal and professional databases, the amount of information is growing in a high and higher speed. Users will be confronted with such a large number of contents provided by multiple sources that efficient and accurate access to this almost infinite amount of content will be even more difficult. The value of information often depends on how easy it can be found, retrieved, accessed and filtered and managed.

Moreover, as the volume of media content increases, identifying and managing them efficiently is becoming more difficult too. To address this problem, we need an all-purpose multimedia content descriptor to facilitate automatic system integration and personalization in searching and retrieval. In fact, more and more audio and visual information nowadays consumed by computational systems rather than human being, in form of creation, exchange, retrieval and re-use. MPEG-7 is a standard proposed to be the solution.

MPEG-7 is an ISO/IEC standard being developed by MPEG (Moving Picture Experts Group). It is aimed at providing a rich set of standardized tools to describe multimedia content. MPEG-7 will be standard for describing the multimedia content data that will support various operational requirements. Both human users and automatic systems that process audiovisual information are considered.

The main elements of the MPEG-7's standard are:

- Descriptors (D): representations of Features, that define the syntax and the semantics of each

feature representation,

- Description Schemes (DS), that specify the structure and semantics of the relationships between their components. These components may be both Descriptors and Description Schemes,
- A Description Definition Language (DDL) to allow the creation of new Description Schemes and, possibly, Descriptors and to allows the extension and modification of existing Description Schemes,
- System tools, to support multiplexing of descriptions, synchronization of descriptions with content, transmission mechanisms, coded representations (both textual and binary formats) for efficient storage and transmission, management and protection of intellectual property in MPEG-7 descriptions, etc.

MPEG-7 Descriptors (D)

Audiovisual data content included by MPEG-7 may include: still pictures, graphics, 3D models, audio, speech, video, and the relations of each element in a multimedia presentation. Special cases of these general data types may include facial expressions and personal characteristics. It will also allow different granularity in its descriptions to offer operation in different levels of details.

As the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications. This implies an object will be described by a set of features, each tuned to the area of application. To take the example of visual material: a lower abstraction level description will include shape, size, texture, color, displacement and position. And for audio: key, mood, tempo, tempo changes, position in sound space. For the highest level abstraction, we will have semantic information like: ‘This is a scene of Beckham

scoring 1-0 in the soccer match United Manchester vs. Liverpool'. Table 4 shows some of the contents descriptions.

Descriptors	*Functionality*
Production Information	Searchable information describing the creation and production processes of the content (director, title, date, keywords, producer, performer, language)
Usage Information	Information related to the usage of the content (copyright pointers, usage history, broadcast schedule, cost of usage)
Physical Information	Information related to the form of media (storage format, encoding, bit-rate, resolution)
Structural Feature	Information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking)
Low Level Feature	Information about low level features in the content (colors, textures, sound timbres, melody description)
Conceptual Information	High level features in terms of reality concepts captured from the content (objects and events, interactions among objects)
Presentation Information	Information about how to browse the content in an efficient way (summaries, variations, spatial and frequency subbands)
References	Pointer to information about collections of objects related to the content (webpage, document, and abstracts)
Interaction Information	Information about the interaction of the user with the content (user preferences, usage history)

Table 4: Content Descriptors

On the other hand, more than a description of the content, other information about the multimedia

data is also included as shown in Table 5.

Descriptors	*Functionality*
Form	An example of the form is the coding scheme used (e.g. JPEG, MPEG-2), or the overall data size. This information helps determining whether the material can be 'read' by the user
Conditions for accessing the material	This includes links to a registry with intellectual property rights information, and price
Classification	This includes parental rating, and content classification into a number of pre-defined categories
Links to other relevant material	The information may help the user speeding up the search
The context	In the case of recorded non-fiction content, it is very important to know the occasion of the recording (e.g. Olympic Games 1996, final of 200 meter hurdles, men)

Table 5: Non-content Descriptors

The MPEG-7 Descriptors are designed for describing the multimedia features includes: low-level audio-visual features such as color, texture, motion, audio energy, and so forth; high-level features of semantic objects, events and abstract concepts; content management processes; information about the storage media, and so forth. It is expected that most Descriptors corresponding to low-level features will be extracted automatically, whereas human intervention will be required for producing the high-level Descriptors.

MPEG-7 Description Scheme (DS)

The MPEG-7 DSs expand on the MPEG-7 Descriptors by combining individual Descriptors as well as other DSs within more complex structures and by defining the relationships among the

constituent Descriptors and DSs. In MPEG-7, the DSs are categorized as pertaining specifically to the audio or visual domain, or generically to the description of multimedia. For example, typically, the generic DSs correspond to meta-data related to the creation, production, usage and management of multimedia as well as to describing the content directly at a number of levels including signal structure, features, models and semantics. Typically, the Multimedia DSs refer to all kinds of multimedia consisting of audio, visual, and textual data, whereas the domain specific Descriptors, such as for color, texture, shape, melody, and so forth, refer specifically to the audio or visual domain.

The MPEG-7 Multimedia DS specification has also defined a number of basic elements that are used repeatedly as fundamental constructs throughout the definition of the MPEG-7 DSs. Many of the basic elements provide specific data-types and mathematical structures, such as vectors and matrices, which are important for audio-visual content description. Also included as basic elements are constructs for linking media files and localizing segments, regions, and so forth. Many of the basic elements address specific needs of audio-visual content description, such as the description of time, places, persons, individuals, groups, organizations, and other textual annotation.

In MPEG-7, there are generic as well as multimedia entities. Generic entities are features, which are used in audio, visual, and text descriptions, and therefore “generic” to all media. These are, for instance, “vector”, “time”, etc. Apart from this set of generic description tools, more complex description tools are standardized. They are used whenever more than one medium needs to be described (e.g. audio and video.) These description tools can be grouped into 5 different classes according to their functionality:

- Content description: describe the Structure (regions, video frames, and audio segments) and Semantics (objects, events, abstract notions)
- Content management: describe different aspects of creation and production, media coding,

storage and file formats and content usage

- Content organization: represent the analysis and classification of several AV contents, organizes collections of audio-visual content, segments, events, and/or objects
- Navigation and access: facilitating browsing and retrieval of audio-visual content by defining summaries, partitions and decompositions, and variations of the audio-visual material
- User interaction: describe user preferences and usage history pertaining to the consumption of the multimedia material

MPEG-7 Description Definition Language (DLL)

The Description Definition Language provides the solid descriptive foundation by which users can create their own Description Schemes and Descriptors and forms a core part of MPEG-7 standard. The DDL defines the syntactic rules to express and combine Description Schemes and Descriptors. According to the definition in the MPEG-7 Requirements Document the DDL is 'a language that allows the creation of new Description Schemes and, possibly, Descriptors. It also allows the extension and modification of existing Description Schemes.'

The DDL is not a modeling language such as Unified Modeling Language (UML) but a schema language to represent the results of modeling audiovisual data, i.e. DSs and Ds. It must satisfy the MPEG-7 DDL requirements: to be able to express spatial, temporal, structural, and conceptual relationships between the elements of a DS, and between DSs. It must provide a rich model for links and references between one or more descriptions and the data that it describes. In addition, it must be platform and application independent and human- and machine-readable. The general consensus within MPEG-7 is that it should be based on XML syntax. Figure 6 shows how the framework put all the things together.

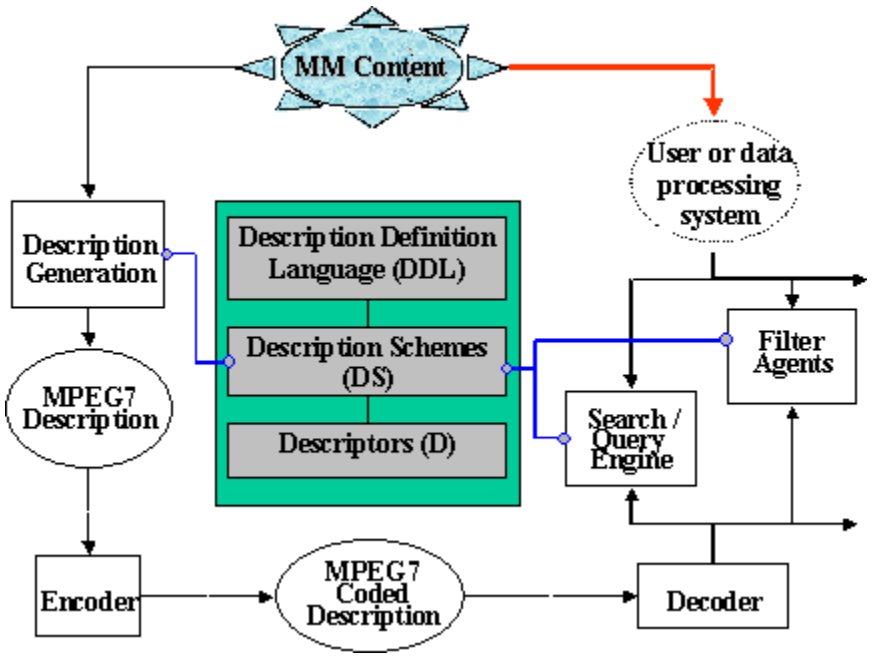

Figure 6: Abstract representation a MPEG-7 system

MPEG-21: Multimedia Framework

Currently, multimedia technology provides services to different players in the multimedia industry from content creators to end-users. Access to information and services can be easily provided with plenty of networks and terminals. However, no complete solutions exist that allow different parties, each with their own models, rules, procedures, interests and content formats, to interact efficiently using this complex infrastructure. A common multimedia framework will facilitate co-operation between different parties and support a more efficient implementation and integration of the different models, rules, procedures, interests and content formats.

The multimedia content delivery chain encompasses content creation, production, delivery and consumption. To support this, the content has to be identified, described, managed and protected. The transport and delivery of content will occur over a heterogeneous set of terminals and networks within which events will occur and require reporting. Such reporting will include reliable delivery,

the management of personal data and preferences taking user privacy into account and the management of (financial) transactions.

As in MPEG-7, MPEG-21 is also an ISO/IEC standard being developed by MPEG (Moving Picture Experts Group). The MPEG-21 multimedia framework will identify and define the key elements needed to support the multimedia delivery chain as described above, the relationships between and the operations supported by them. Within the parts of MPEG-21, MPEG will elaborate the elements by defining the syntax and semantics of their characteristics, such as interfaces to the elements. MPEG-21 will also address the necessary framework functionality, such as the protocols associated with the interfaces, and mechanisms to provide a repository, composition, conformance, etc.

The seven key elements defined in MPEG-21 is shown in table 7

Element	*Description*
Digital Item Declaration	a uniform and flexible abstraction and interoperable schema for declaring Digital Items
Digital Item Identification and Description	a framework for identification and description of any entity regardless of its nature, type or granularity
Content Handling and Usage	provide interfaces and protocols that enable creation, manipulation, search, access, storage, delivery, and (re)use of content across the content distribution and consumption value chain
Intellectual Property Management and Protection	the means to enable content to be persistently and reliably managed and protected across a wide range of networks and devices
Terminals and Networks	the ability to provide interoperable and transparent access to content across networks and

	terminals
Content Representation	how the media resources are represented
Event Reporting	the metrics and interfaces that enable Users to understand precisely the performance of all reportable events within the framework

Table 7: Key elements defined in MPEG-21

MPEG-21 recommendations will be determined by interoperability requirements, and their level of detail may vary for each framework element. The actual instantiation and implementation of the framework elements below the abstraction level required to achieve interoperability, will not be specified.

XML Schema Overview

Extensible Markup Language (XML) is a low-level syntax for representing structured data and can be used to support a wide variety of applications. This idea is put across in a simplistic way in the figure 8, which shows how XML now underpins a number of Web markup languages and applications.

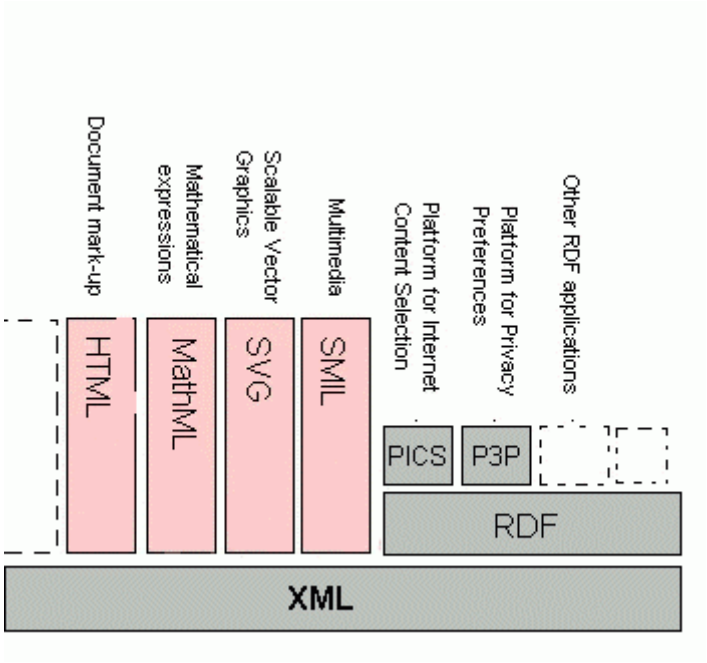

Figure 8: Application and Markup languages building on XML

XML is a subset of SGML and it has become a widely used format for encoding data (including metadata and control data) for exchange between loosely coupled applications. Such exchange is currently hampered by the difficulty of fully describing the exchange data model in terms of XML DTDs; exchange data model versioning issues further complicate such interactions. When the exchange data model is represented by the more expressive XML Schema definitions, the task of mapping the exchange data model to and from application internal data models will be simplified.

The purpose of a schema is to define and describe a class of XML documents by using these constructs to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content, attributes and their values, entities and their contents and notations. Schema constructs may also provide for the specification of implicit information such as default values. Schemas document their own meaning, usage, and function.

From an application developer's perspective, a schema sets constraints on what is allowable in an XML document. With the schema defined, the processing across different system modules and even

across different developer's work will be interoperable.

In an XML document, a schema is a description of the way that the document is marked up: its grammar, vocabulary, structure, datatype, etc. Data Type Definition (DTD) is the classical schema in the XML specification, but there are some disadvantages:

- Not extensible
- Not describing XML as data well,
- Inheritance between DTDs is not supported
- Not provide support for namespaces

In fact, some of these disadvantages are due to the lack of hierarchical abstraction in DTD, such that there is no way to extend a class unless re-write the DTD, and no way to inherit a class to form a sub-class. We are looking into methods to address this problem.

In a DVL system, flexibility and extensibility is essential to the system quality and we need to define how the elements in the XML related to each other and also, the way they are assembled together. XML appears to be the promising solution. However, to apply XML in real application, we need to have a generalized plan to govern their action. This can be done by XML schemas, which define the characteristics of classes of objects and the way that data is marked up by applying particular constructs to constrain their structure.

Research Plan

For building a DVL, we face a problem in designing the messaging model. More specifically, we will like to have an extensible dispatcher that we can plug-in/remove any function modules easily, and the question will be: ‘How can a module get relevant information effectively from a bulk XML message or XML data stream, given a schema for each module?’

Following this idea, we will like to investigate if there can be multiple DTDs for a single XML document, and what will be the effect of multiple DTD on data creation and data consumption. Furthermore, we may also investigate on datatypes that is essential to facilitate video description and system integration but not yet in the DTD specification.

Multiple DTDs, more than composite one large DTD from multiple small DTD, but also involve object hierarchy. This not only involves the definition of DTDs, but also the mechanism to process the document when only part of the DTD is given.

Data types present in DTD specification are rather primitive. Lack of high-level data type results in more difficult in expression. We will like to explore the usage of arrays, trees and objects in DTD.

The following related XML Schema Proposals will also be studied as comparison:

- XML-Data (XDR)
- Document Content Description (DCD)
- Schema for Object-oriented XML (SOX)
- Document Definition Markup Language (DDML)
- Schematron

- Datatypes for DTDs (DT4DTD)
- REgular LAnguage description for XML (RELAX)
- Document Structure Description (DSD)
- TREX (Tree Regular Expressions for XML)

Conclusion

As a conclusion for the work in my first year, I have studied the framework of Digital Video Library systems and XML as a data management tools.

In the first semester, I have surveyed and learned the technologies associated with a Digital Video Library. This includes general indexing, searching and retrieval methods in an information system. More effort is spent on the content-based indexing of video contents and a modular architecture of DVL is proposed and studied.

In the second semester, I have focused on the data representation and messaging method in a multi-tier DVL architecture. XML is studied in various aspects for its feasibility to solve the problem. A research plan is also proposed on enhancing the DTD for extensibility and expressive power.

Lastly, I want to mention that this project of DVL development is under the VIEW technologies project funded by the ITF. As this is not just a research project, the data management model (as well as other system modules) should be concrete to the standard of other industrial strength product. I also hope that the research result will contribute in this project.

Reference

[1] A. Hampapur. Semantic Video Indexing : Approach and Issues. *ACM Sigmod Record*. 1996.

[2] A. Hauptmann, M. Witbrock. Story segmentation and detection of commercials in broadcast news video. *Proceedings of Advances in Digital Libraries Conference*, Santa Barbara, April 1998.

[3] A. Jaimes, F-F Chang. Model based image classification for content-based retrieval. *SPIE Conference on Storage and Retrieval for Image and Video Database*, San Jose, CA, January 1999.

[4] A. Levy. http://www.cs.washington.edu/homes/alon/widom-response.html. *More on data management for XML*, 1999.

[5] A. Merlino, D. Morey, D. Maybury. Broadcast news navigation using story segmentation. *Proceedings of ACM Multimedia*, November 1997.

[6] A.M. Pejtersen. Semantic information retrieval. *Communications of the ACM*, 41(4):90--92. Petrie, C. J. (1998). The XML files. IEEE Internet Computing, pages 4--5.

[7] A.P. de Vries and H.M. Blanken. Database technology and the management of multimedia data in Mirror. In Multimedia Storage and Archiving Systems III, volume 3527 of *Proceedings of SPIE*, Boston MA, November 1998.

[8] A. Vogel et al., "Distributed Multimedia and QOS: A survey, " *IEEE Multimedia*, vol. 2, no. 2, Summer 1995, pp. 10-19.

[9] B. Shahraray, D. Gibbon. Efficient archiving and content-based retrieval of video information on the Web. *AAAI Symposium on Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*, Stanford, CA, March 1997, pp 133-136

[10] B. Surjanto, N. Ritter and H. Loeser. XML Content Management based on Object-Relational Database Technology. *Proc. of the 1st Int. Conf. On Web Information Systems Engineering* (WISE), Hong Kong, June 2000.

[11] B. Yates, R Neto, *Modern Information Retrieval*, Addison Wesley.

[12] Bridle, J. S., Brown, M. D. and Chamberlain, R. M., An algorithm for connected word recognition. *In Automatic speech analysis and recognition (J.P. Haton, editor)*, pp.191-204. Dordrecht, Holland: D. Reidel.

[13] H.D. Wactlar, T. Kanade, M. A. Smith and S. M. Stevens. "Intelligent Access to Digital Video: Informedia Project". *IEEE Computer*, Vol.29, No.3, pp.46-52, May 1996.

[14] J. Widom. Data management for XML: Research directions. *IEEE Data Engineering Bulletin*, 22(3):44--52, Sept. 1999.

[15] M.G. Christel, D. Martin. Information Visualization Within a Digital Video Library. *JIIS* 11(3): 235-257 (1998).

[16] M.A. Smith and M. Christel. Automating the creation of a digital video library. In *Proceedings of Third ACM International Conference on Multimedia*, pages 357-358, Anaheim, CA, November 1995.

[17] M. A. A. Tatham, An Integrated Knowledge Base for Speech Synthesis and Automatic Speech Recognition, *Journal of Phonetics (1985) 13*, pp. 175-188, Academic Press Inc. (London) Limited.

[18] Moore, R. K., Overview of speech input. *In Proceedings of the 1st international conference on speech technology (J. N. Holmes, editor)*, pp. 25-38. Bedford: IFS (Publications) Ltd. Amsterdam: North Holland.

[19] Nalin K. Sharda, *Multimedia Information Networking*, Prentice Hall, 1999

[20] Q. Huang, Z. Liu, A. Rosenberg. Automated semantic structure reconstruction and representation generation for broadcast news. *Proceedings SPIE, Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, January 1999.

[21] R. Kahn and R. Wilensky. A framework for distributed digital object services. *Technical Report cnru.dlib/tn95-01, CNRI*, May 1995. http://www.cnri.reston.va.us/k-w.html.

[22] S.F. Chang, Q. Huang, A. Puri, B. Shahraray and T. Huang. *Multimedia search and retrieval. Multimedia Systems, Standards, and Networks*, MARCEL DEKKER, 2000.

[23] Stephen W. K. Fu, C. H. Lee, Orville L. Clubb, A Survey on Chinese Speech Recognition, 23 November, 1995.