Component-Based Design and Integration of a Distributed Multimedia Management System[*]

Shu-Ching Chen[1], Mei-Ling Shyu[2], Na Zhao[1] and Chengcui Zhang[1]
[1]Distributed Multimedia Information System Laboratory, School of Computer Science
Florida International University, Miami, FL 33199, USA
[2]Department of Electrical & Computer Engineering
University of Miami, Coral Gables, FL 33124, USA
Email: [1]{chens, nzhao002, czhang02}@cs.fiu.edu, [2]shyu@miami.edu

Abstract − *In order to fulfill the functional require-ments of multimedia management systems while achiev-ing the efficiency and reliability at the same time, we adopt the component-based design to develop a compli-cated distributed multimedia management system called* DMMManager *in this paper.* DMMManager *integrates a set of core components for managing multimedia data, such as content-based retrieval for image and video data, and multimedia presentation design and render-ing. Component-based design makes the reuse and share of the components easy and reliable, and thus avoids the waste of both time and man power in the development cycle. Moreover, new applications can be easily inte-grated into this system as a new component and utilize the existing components. Besides component reuse, the system is also designed to facilitate information reuse. For example, the output of the image/video retrieval component can be taken as the input for the multime-dia presentation component. In addition, the use of subnetworks in our multimedia presentation design also enables the modulized reuse of the existing presentation models that are previously created.*

Keywords: Multimedia retrieval, multimedia author-ing, multimedia presentation, component-based design.

1 Introduction

The advances in multimedia capture and multime-dia processing technologies lead to the desire to store, analyze, organize, retrieve, and display digital multime-dia data. As a result, various multimedia data man-agement systems have been developed to fulfill these requirements with sharing and distributing multimedia data through computer networks. However, lots of time and man power have been spent on implementing the same or similar functions including user interfaces, the commonly used algorithms for media analysis, etc. The

purpose of our research is to develop such a component-based multimedia management system which tries to cover a full scope of functionalities for multimedia man-agement systems, and can be extended and maintained easily. For such a purpose, a set of core components are developed and can be reused conveniently to avoid resource wasting.

A distributed multimedia management system, *DM-MManager* is introduced in this paper. The DMMMan-ager system supports a set of core functionalities includ-ing multimedia data analysis, indexing, organization, content-based retrieval, real-time capturing, multimedia presentation design and rendering, etc. By modeling the subsystems into components, the whole system is com-posed of a set of reusable and extensible components, making the system easy to maintain and to extend in accordance with the rapid advances in multimedia tech-nologies.

The design and implementation of three major com-ponents: (1) image management component, (2) video management component, and (3) multimedia presenta-tion component are discussed. The image and video management components provide the users the great convenience in obtaining their desired media by issu-ing database queries. Then, a multimedia presenta-tion component is developed to help the users to in-corporate different types of media data into the mul-timedia presentation design, and to render multimedia presentations based on the created presentation models. Compared with other existing work, the proposed mul-timedia presentation component adopts the Multimedia Augmented Transition Network (MATN) model, which is an abstract semantic model for multimedia presen-tations and has several advantages over other existing models in terms of representation power and informa-tion reuse [3].

The paper is organized as follows. In the next section, related work is reviewed and summarized. In Section 3, three major components as well as their subcompo-

---

nents are discussed in details. Section 4 describes the distributed system architecture and system integration. The final remarks are given in the Section 5.

2 Related Work

In this section, a review of other existing multimedia management systems and the current trends in multimedia research are presented, and then the benefits of adopting component-based design in the development of multimedia management systems are summarized.

The content-based retrieval techniques for multimedia databases are discussed in [16], where two principle ways for the representation of queries: query by subject/object and query by example are summarized. Most of the current content-based image retrieval systems may focus on only one aspect of the requirements for managing multimedia data, such as new indexing data structures, sketching capabilities, or providing relevance feedback, etc.

Another major research areas in multimedia research is multimedia presentation modeling which also leads to the need of a multimedia presentation design system that is powerful in functionalities and easy to use for general users. For example, an object-based knowledge representation system called AROM is proposed in [8], which provides a multimedia presentation authoring tool called V-STROM. The multimedia presentation scenario can be modeled by UML diagram class-like descriptions. As another example, a standard reference model (SRM) is proposed in [2] to describe the intelligent multimedia presentation system (IMMPS). A generic architecture is presented to analyze and to compare the existing IMMPSs.

In addition, how to distribute multimedia data through the computer network so that the users can access or handle remote multimedia data conveniently is also important in multimedia research. Tavanapong et al. [14] describes the design and implementation of a video browsing system on the Internet, in which a normal playback and browsing mechanism is supported. In [11], the authors proposed a real time multimedia presentation system for media-on-demand through the Internet by using the RTP protocol.

Recently, component-based design emerges as a new paradigm aiming at software reuse from the software architecture's view [1] [10]. The components are able to be plugged into and to play with other components in the system. The work in [9] demonstrates a scalable distributed multimedia system framework using component-based design.

3 Component-Based Design

Figure 1 shows the four main components as well as their subcomponents in the *DMMManager* system. The four components are *Image Management Component*, *Video Management Component*, *Multimedia Pre-*
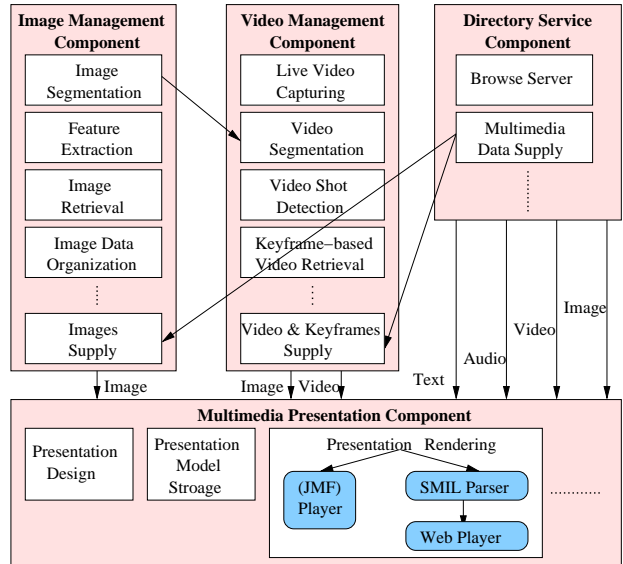

Figure 1: An overview of DMMManager components

sentation Component, and *Directory Service Component*. These components have the ability to communicate with each other, exchange data, support different functionalities of each other, and share and reuse some of their subcomponents. The multimedia information reuse is also realized via the data channel between image/video retrieval components and multimedia presentation component. DMMManager provides the functionality to browse all the directories in the server database. Any multimedia data which are stored in the server side file systems can be browsed and downloaded directly. This component shares the file supply subcomponent with the above three components. Since the functionality of this component is clear and straightforward, we will not include it in the detailed discussions.

3.1 Image Management Component

In addition to keyword-based Internet image search engines, much more efforts are put on the research in developing effective content-based image retrieval algorithms and systems. Based on the surveys in image retrieval systems [15] [16], we found that most of the existing image retrieval systems share the following common features. First, the features of all the images, including low- and/or mid-level features such as color, texture, shape, and object locations, are extracted and stored. Then the users can issue database queries by specifying an query image. The feature vector of this query sample is used to fetch the similar images from the database based on some similarity measures. Finally the query results are displayed to the users.

Figure 2 illustrates the details of our image management component. This component conducts the oper-

ations on the image data, such as image segmentation, low- or mid-level feature extraction, image data organization, and image supply, where content-based image retrieval is the main functionality. At current stage of its implementation, the image feature extraction phase considers both the low-level features (color features in HSV color space) and the mid-level features (the spatial locations of objects). Compared to the color features, object information is much more difficult to obtain. In our system, we implemented an unsupervised image segmentation method named *Simultaneous Partition and Parameter Estimation (SPCPE)* [4] to get the object location information.
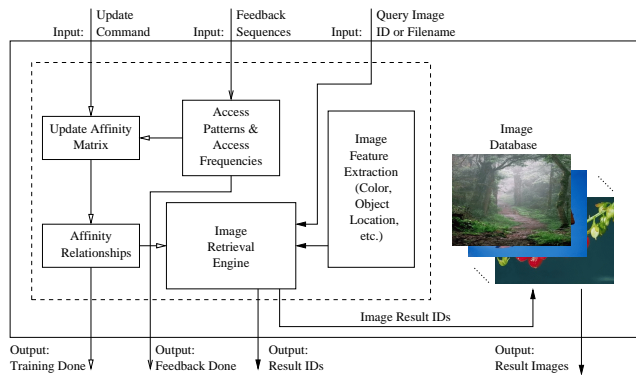

Figure 2: Image management component

Like some advanced image retrieval systems, we also provide the user feedback functionality so that users' perceptions can be captured and utilized to refine the query results. Relevance feedback (RF) [12] is a popular solution when dealing with the semantic gap between low-level image features and high-level user concepts. Upon one single query image, the user needs to select the similar images to offer the feedback. The feedback is then used to update the similarity metric, and the updated retrieval results are sent back to the user for next iteration of feedback. Using this way, the system tries to learn the high-level user concepts via several iterations of feedback.

The feedback facility is integrated into our system. However, different from the traditional relevance feedback CBIR systems, we adopt a novel image retrieval and learning mechanism with better training performance and plug it into our DMMManager system. In this subsystem, the Markov Model Mediator (MMM) mechanism [13] is adopted to learn the high-level semantic concepts [7]. Multiple users are involved in the training processing. Instead of training for one query image, the feedbacks are provided for a set of query images. After collecting the feedback sequences and sending them to the server engine, the corresponding access patterns and access frequencies are stored in the database. When a certain amount of feedbacks are collected, the user

may choose to update the affinity matrix, or let the system automatically update the affinity matrix. The affinity matrix represents the relative affinity relationships among all the images in the database. Whenever two images are accessed together in a user query, their relative affinity relationship increases. Unlike the RF systems, the sparse mistakes made by individual users will not affect the overall performance of the semantic concepts learned by using the MMM mechanism since it is trained by using multiple users' feedbacks, and the low-level image features are combined with the relative affinities for image retrieval.
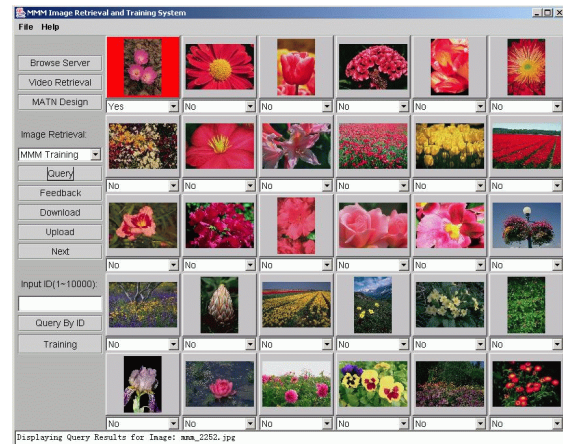

Figure 3: Content-based image retrieval interface

The operation flow of the MMM-based image retrieval and training system is illustrated in Figure 2. In addition, a friendly user interface as shown in Figure 3 is designed and implemented for image retrieval, user feedback collecting, and system training. By utilizing the MMM mechanism, the users do not need to take the heavy responsibilities to train the system just for one query image. Also, we provide a functionality that, when a user finds the desired image(s), he/she can download it from the server to the client side, and then utilize it for future presentation design. It is worth mentioning that the user interface is an independent component. Although there can be more than one underlying retrieval mechanisms, the user interface is only responsible for collecting user's request/feedbacks and displaying the results. Hence, it is a reusable component. In fact, whatever the image retrieval mechanism is, it can be implemented as a black box by specifying some standard input and output interfaces. In other words, the CBIR components can be encapsulated and have the advantages of replaceability and extensibility.

3.2 Video Management Component

Figure 4 shows the architecture of the video management system. Basically, there are three subcomponents, namely *live video capture*, *video segmentation*,

and *keyframe-based video retrieval*. The live video capture subcomponent can supply the live raw videos captured via a web camera. The video segmentation subcomponent takes the raw video data, and segment it into a sequence of video shots, where a video shot is an unbroken sequence of video frames from one camera. The output of video segmentation, such as the key frames (representing each shot) and video shot boundaries, are then used in the keyframe-based video retrieval subcomponent, where the users are allowed to issue video database queries based on the keyframes, and to browse the returned video shots. The three subcomponents works together to help retrieve and provide the desired video data for future multimedia presentation designs. Since the video data can also be regarded as a composition of a set of continuous images/frames, the image segmentation subcomponent can be reused here to analyze video frames and to help detect video shot boundaries.
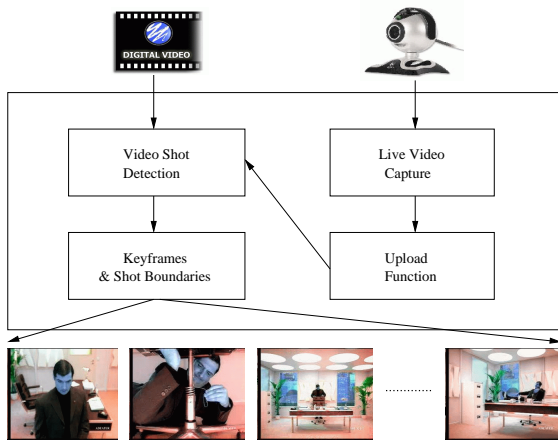

Figure 4: Video management component

(1) Live Video Capture: This component has the ability to capture, encode, decode, and display live video data as well. It supports the real-time encoding and displaying of live videos of AVI and MPEG formats. The captured raw videos are stored in the client side at the beginning. Then by using the 'upload' function in video management component, the captured video data are transferred to the server side for future use. Also, the video segmentation algorithm can be activated to process the captured AVI or MPEG video files.

(2) Video Segmentation: In our system, there are two input sources for video data. One is via live video capture subcomponent, and the other one is via user-supplied videos downloaded from web or digitized by themselves. Video data can be regarded as a composition of a set of consecutive images/frames. The consecutive video frames within one camera motion form a video shot. For this subcomponent, we implement an unsupervised video shot detection method previously proposed by us in [5]. This method incorporates multiple techniques for shot detection, such as histogram comparison, pixel-level comparison, and object segmentation/tracking. Since our system is component-based, we can utilize the histogram extraction and image segmentation subcomponents from the image management component for shot detection purpose.

For simplicity, the first frame of each detected shot is considered as the keyframe representing the whole shot. In fact, our system has the ability to reach the finer granularities of keyframe representation by considering the relative spatial-temporal relationships between video objects [6]. After the video is processed, it has been cut into several smaller video clips based on shot boundaries and stored together with the corresponding keyframes. The stored shot boundary and keyframe information are valuable for future video indexing and retrieval. Both AVI and MPEG formats are supported in this processing.

(3) Keyframe-Based Video Retrieval: Our video retrieval subcomponent supports the keyframe-based video retrieval. By specifying a keyframe, the user can retrieve and browse the desired video shot represented by that keyframe. As shown in Figure 5, a set of representative keyframes for a video sequence is displayed to the user, giving an overview of the contents for that video. The corresponding shot can be displayed by double-clicking the keyframe of interest. Both the whole video and the segmented video clips can be downloaded for future use (i.e., multimedia presentation design). This keyframe-based video retrieval is actually a special case of general image retrieval, such that the video retrieval component can be connected to image retrieval component and fully utilize all the underlying retrieval facilities.
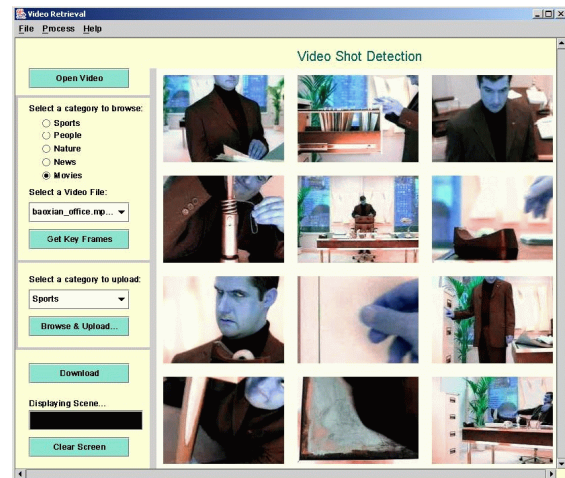

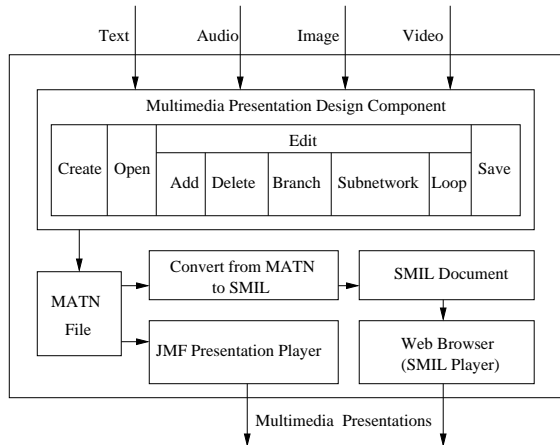Figure 5: Keyframe-based video retrieval interface

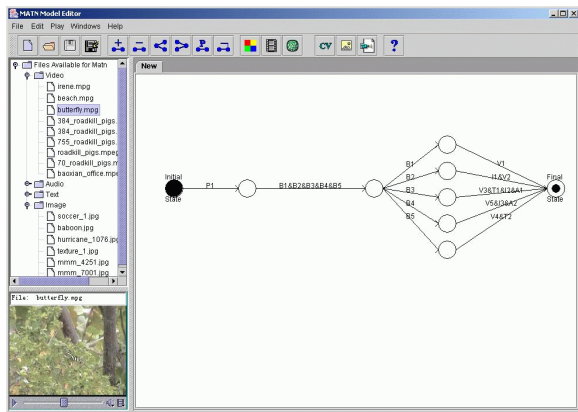Figure 6: Multimedia presentation design and rendering component

Figure 7: Multimedia presentation design interface

3.3 Multimedia Presentation Component

A multimedia presentation is a temporal composition of various kinds of multimedia data, in which a set of multimedia data with various temporal relations are arranged to appear within one single context. Nowadays, the multimedia authoring and presentations have become very important in many applications of different areas such as remote education.

We implemented the multimedia presentation component which provides easy-to-use utilities for the users to easily create, open, edit, save, and display a multimedia presentation. Generally speaking, the multimedia presentation component can be split into two major modules: One is for multimedia presentation authoring, the other is for multimedia presentation rendering. Figure 6 gives an overview of the embedded multimedia presentation component. Various types of multimedia data can be gathered and archived from the other components described above. In the presentation de-

sign component, a visualization interface is developed so that the user can get familiar with all the design tools quickly. Some editing tools are provided to the users for creating multimedia presentation models. The created presentation models can be saved and stored for future use. The Multimedia Augmented Transition Network (MATN) model [3] is adopted as the abstract semantic model as well as the internal data structures for multimedia presentations. User interaction and information reuse are supported by some specific features of the MATN model, e.g., branching, loop, subnetwork, etc.

In the presentation rendering layer, two options are provided to fulfill the needs of different environments and different requirements. The designed multimedia presentation model can be rendered as a stand-alone application via Java Media Framework (JMF) [17], or it can be translated into a SMIL [18] document by a SMIL parser subcomponent. The resulted SMIL document describes the same presentation structure and can be rendered via web browsers.

(1) MATN Design Layer: Compared to the traditional unimedia presentations, the multimedia presentations are far more complicated and hence difficult to describe. Various semantic models for multimedia presentation are proposed to deal with this issue, which can be categorized into timeline-based models, script-based models, graph-based models, and structure-based models. Some presentation models do not support the complete spatial and temporal relationships between media streams, some models do not provide support to model user interactions during presentation, while some other models are too complex to understand by general users. In our DMMManager system, an abstract semantic model called Multimedia Augmented Transition Network model (MATN) for multimedia presentation and authoring modeling is incorporated [3].

An MATN model consists of a group of states connected by the directed arcs. It adopts the multimedia input string [3] as its input, which is a regular expression like representation of the spatio-temporal relationships between media streams. In other words, it describes the spatio-temporal combination of media streams and how they will be displayed.

The MATN model has several advantages. First, the major advantage of the MATN model is its support for user interactions in the multimedia presentations by adopting the concepts of branch and loop. As shown in Figure 7, a branching presentation structure is designed so that the user can determine which direction this presentation will go and be rendered at run time. Second, the MATN model can express a multimedia presentation in a hierarchical way by introducing the subnetwork concept. Third, the quality of service control can be achieved by using the condition/action table in which the appropriate actions are specified with respect

to different network conditions. These actions may affect and change the rendering order and quality of a multimedia presentation. Our implementation of this component supports all the featured properties of the MATN model, such as the user interaction, loops, subnetwork, and quality of service (QoS) control.

As an important feature of the MATN model, a subnetwork can be used to represent an already-existing and well-defined multimedia scenario which is referred by any other multimedia presentations. When a subnetwork is encountered during the interpretation of an MATN model, the control of the main presentation is passed to the subnetwork level in the way that the presentation flow within the subnetwork is inserted seamlessly into the main presentation. Figure 7 shows a presentation with a subnetwork (P1) on the first arc. When the presentation of the subnetwork finishes, the control goes back to the main presentation level and continues its presentation from the next state of the previously exit state. This capability makes it possible for an MATN to illustrate a complicated scenario without losing the integrated view. The subnetwork is employed to manifest a multimedia presentation in a hierarchical manner, which can clarify the complicated scenario involving a lot of spatial/temporal relations. Therefore, subnetwork enables the reuse of any existing multimedia presentations, which definitely offers great convenience to the user.

(2) Rendering Layer via JMF: After a user creates a multimedia presentation in the design layer, the presentation itself is still a presentation model based on MATN. Thus the presentation rendering subcomponent is needed to turn it into a visual multimedia presentation perceivable to the users. The Java Media Framework provides a good solution for rendering this kind of presentation in a runtime environment [17].
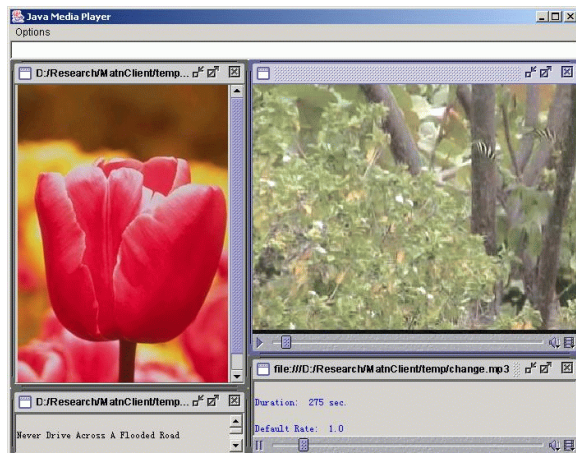

Figure 8: Multimedia presentation display interfaces

The demonstrated MATN presentation model has a

clean structure while containing many useful information. For example, each arc has its own time duration specification. For all the media objects involved in this presentation, a feature set is utilized to store the corresponding features such as file name, specified or fixed duration, display window position, and display window size. These hidden information is needed when rendering the multimedia presentation. Any part of the presentation can be rendered based on the user's preference. An presentation rendering example via Java Media Framework is shown in Figure 8. Four multimedia objects of different types (image, video, audio, text) gathered from other components are displayed concurrently.

(3) Rendering Layer via SMIL: SMIL language is a good example of the script-based presentation models, which provides powerful functionalities to create synchronized multimedia presentations. However, it's not easy to learn a new programming language for general users. In contrast to this, our system provides the users the facility to design a MATN-based multimedia presentation in a graphical environment, then by using a SMIL parser, the created MATN model can be automatically translated into a SMIL script upon users' request.
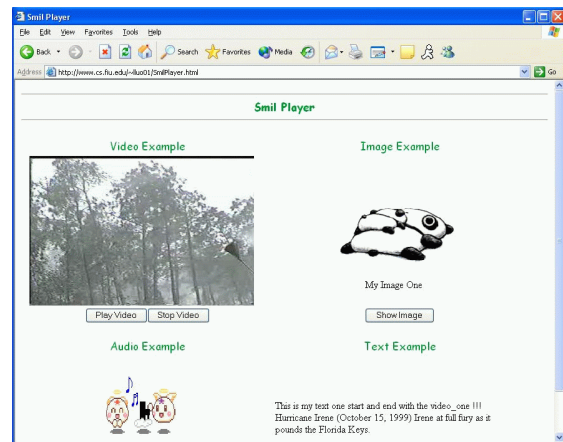

Figure 9: Multimedia presentation rendering via web browser

The implementation of the SMIL parser guarantees the correctness of keeping the same conceptual structure, temporal relations, and synchronization controls as those in the original MATN model. The multimedia file locations and durations can be easily fetched from the well-structured MATN model and are employed to convert the MATN presentation model into HTML+SMIL. With the easy-to-use MATN design interface, the heavy duty to learn a programming language is avoided. Also, the MATN structure serves as a reusable data model component, providing more choices for the users to render the multimedia presentation. Figure 9 demonstrates a SMIL presentation which

is converted from an existing MATN model.

4 System Integration

4.1 Distributed System Architecture

One of the targets for distributed architecture design is to make sure that the system has the ability to carry user requests or multimedia data and to deliver them to any kind of client or server in an efficient way. By adopting a multi-threaded client/server architecture, our DMMManager system supports simultaneous accesses of multiple users. The server engine can assign a new thread to handle a new request received. The UDP protocol is utilized in the server and client communication, and the TCP protocol will take its place when the complete multimedia contents are required for the playback of multimedia streams. When handling the presentation playback, the RTP protocol will be used.

4.2 The Client Side Integration

As demonstrated in Figure 10, several user interfaces implemented by Java are embedded into the DMMManager system, such as content-based image retrieval interface, live video capture interface, video retrieval interface, browse server interface, presentation design interface, and presentation rendering interface. These interfaces are used to collect the user requests and to display the results to the users as well. Users can select and download the requested multimedia data for the purpose of presentation design and rendering. Thus the connection between the presentation module and the media management modules is established. Upon the 'download' request, the requested media file are added into the tree view of the available multimedia materials, such that the user can select and use it in the presentation design phase. The MATN presentation models are then used to render the multimedia presentations by using either JMF player or web browsers.

Since the users may issue diverse requests for different multimedia data at the same time, the user requests are categorized based on the domain, media type, and request content during packaging. Meanwhile, the result packages that need to be sent back to the users are also categorized to ensure the correct deliveries.

4.3 The Server Side Integration

In addition to storing and organizing the huge amount of multimedia data on the server side, all the computation intensive functions are also deployed on the server side in order to make the full usage of the server's storage space and computation power. Such algorithms include image segmentation, image matching and retrieval, video segmentation, etc. Three major components are embedded in the server side, namely image management system, video management system, and directory service component. In the meanwhile, some other applications are integrated as well, such as the

control of data file transmission and the server side directory browsing. Upon receiving the users' requests, the server first analyzes the message and identify what kind of operation the user did and which application needs to be run next. The corresponding calculations or operations are then activated accordingly.
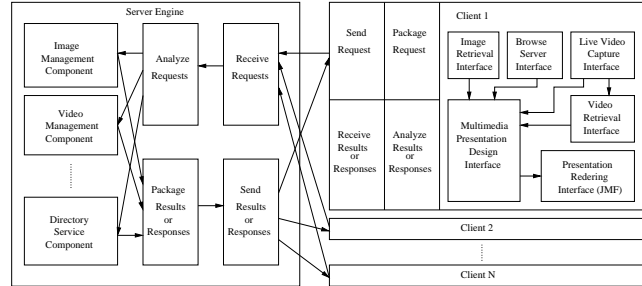

Figure 10: System architecture

5 Conclusions

In this research, we seek to build up a component-based multimedia management system to efficiently organize, analyze, retrieve, and display multimedia data, as well as to realize the multimedia presentation design and rendering. A set of key components are developed to handle various multimedia data management issues, such as image retrieval, video retrieval, and multimedia presentation design and rendering. These well-designed components are flexible, extensible, and easy to maintain, which guarantees the reusability of the system components. In addition, the information reuse is realized in our system via several ways: 1) the information of user access patterns and access frequencies is stored and used for further refinement of the underlying image retrieval mechanism; 2) the retrieval results from image/video management components can be directly reused by the multimedia presentation component; 3) the employment of the MATN model and its subnetworks enables the modulization and reuse of the existing presentation models.

A distributed system architecture is also developed, where the multimedia database is maintained on the server side. Besides, all the computation intensive functions are deployed on the server side as well. User-friendly interfaces are developed and connected with each other so that the user can switch between them easily. Another advantage of the proposed architecture is that it allows some level of openness in its architecture, such that a new multimedia application can be plugged into this architecture easily, which is also another proof of the extensibility of this system. In brief, the DMMManager system exploits the possible solutions (component-based design) for integrating various functionalities in multimedia management systems

while also enabling the information reuse at the maximum level.

Acknowledgment

References

[1] M. Aoyama, "New age of software development: How component-based software engineering changes the way of software development", Proc. 20th International Conference on Software Engineering, pp. 30-37. IEEE Computer Society Press / ACM Press, 1998.

[2] M. Bordegoni, G. Faconti, S. Feiner, M. Maybury, T. Rist, S. Ruggieri, P. Trashanias, M. Wilson, "A Standard Reference Model for Intelligent Multimedia Presentation System", *Computer Standards and Interfaces*, vol. 18(7), 1998, pp. 477-496.

[3] S.-C. Chen, R.L. Kashyap, and A. Ghafoor, *Semantic Models for Multimedia Database Searching and Browsing*, Kluwer Academic Publishers, ISBN 0-7923-7888-1, September 2000.

[4] S.-C. Chen, M.-L. Shyu, X. Jin, Q. Chen, C. Zhang, and J. Strickrott, "A Flexible Image Retrieval and Multimedia Presentation Management System for Multimedia Databases", Proc. ACM Multimedia 2001 conference, pp. 601-602, September 30 - October 5, 2001, Ottawa, CANADA.

[5] S.-C. Chen, M.-L. Shyu, C. Zhang, and R.L. Kashyap, "Video Scene Change Detection Method Using Unsupervised Segmentation and Object Tracking", Proc. IEEE International Conference on Multimedia and Expo (ICME), pp. 57-60, August 22-25, 2001, Waseda University, Tokyo, Japan.

[6] S.-C. Chen, M.-L. Shyu, C. Zhang, and R.L. Kashyap, "Identifying Overlapped Objects for Video Indexing and Modeling in Multimedia Database Systems", *International Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 715-734, December 2001.

[7] S.-C. Chen, M.-L. Shyu, N. Zhao, and C. Zhang, "An Affinity-Based Image Retrieval System for Multimedia Authoring and Presentation", accepted for publication, The 11th Annual ACM International Conference on Multimedia, November 2-8, 2003, Berkeley, CA, USA.

[8] A. Ketfi, J. Gensel, and H. Martin, "Enhancing a Multimedia Presentation System with Object-Based Knowledge Representation", Proc. Internet and Multimedia Systems and Applications (IMSA 2001), pp. 264-269, Honolulu (HA) USA, August 13-16, 2001.

[9] F. Kon, R. H. Campbell, S. Tan, M. Valdez, Z. Chen, and J. Wong, "A Component-Based Architecture for Scalable Distributed Multimedia", Proc. 14th Intl. Conf. on Advanced Science and Technology, pp. 121-135, Naperville, April 1998.

[10] E. A. Lee and Y. Xiong. System-Level Types for Component-Based Design. Technical Memorandum UCB/ERL M00/8, Electronics Research Lab, University of California, Berkeley, CA 94720, USA, February 29, 2000.

[11] S. Palacharla, A. Karmouch, and S. A. Mahmoud, "Design and Implementation of a Real-time Multimedia Presentation System using RTP", Proc. 21st International Computer Software and Applications Conference (COMPSAC'97), pp. 376-381, Washington. DC, August 1997.

[12] Y. Rui, T. S. Huang, and S. Mehrotra, "Relevance Feedback: A Powerful Tool for Interactive Content-Based Image Retrieval", *IEEE Transactions on Circuit and Video Technology, Special Issue on Segmentation, Description, and Retrieval of Video Content*, 18(5), pp. 644-655, 1998.

[13] M.-L. Shyu, S.-C. Chen, and C. Zhang, *A Stochastic Content-Based Image Retrieval Mechanism*, accepted for publication, Multimedia Systems and Content-based Image Retrieval, Edited by Sagarmay Deb, Idea Group Inc., publisher of the Idea Group Publishing, Information Science Publishing and IRM Press imprints.

[14] W. Tavanapong and K. A. Hua, "Design and implementation of a video browsing system for the internet", *Journal of Software-Practice and Experience*, 31(5): pp. 471-503, April 2001.

[15] R. C. Veltkamp and M. Tanase. Content-Based Image Retrieval Systems: A Survey. Technical Report UU-CS-2000-34, October 2000.

[16] A. Yoshitaka and T. Ichikawa, "A Survey on Content-Based Retrieval for Multimedia Databases", *IEEE Trans. on Knowledge and Data Engineering*, 11(1): pp. 81-93, Jan 1999.

[17] Java Media Framework, http://java.sun.com/products/java-media/jmf/.

[18] Synchronized Multimedia Integration Language (SMIL), http://www.w3.org/TR/smil20/.