ISSN No. 0976-5697

Volume 5, No. 8, Nov-Dec 2014

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

A New Approach for Processing Different Multimedia Formats

Nisha P. Shetty
Department of CS&E
Canara Engineering College
Mangalore, India

Saritha Suvarna
Department of CS&E
Canara Engineering College
Mangalore, India

Supreetha D.R.
Department of CS&E
Canara Engineering College
Mangalore, India

Alok Ranjan
Assistant Professor, Department of CS&E
Canara Engineering College
Mangalore, India

Abstract: Multimedia is a repository of knowledge in forms such as images, text, audio and video. In this paper the methodology presented facilitates the searching and delivery of all media types among users connected via Internet. An efficient means for teleconferencing is provided which is reliable across WANs, supports heterogeneity and is interoperable with other implementations. For faster retrieval of data encapsulation is used where data, object and user implementations pertaining to a single object is grouped together as a single entity. Increase in speed during retrieval, no data loss during transmission makes this system favourable.

Keywords: multimedia; encapsulation; connection management; content based multimedia retrieval; groupware

I. INTRODUCTION

In this digital era, a large amount of multimedia data have been amassed due to the development of the Internet, the World Wide Web and new devices for producing digital content such as digital (video) cameras. Thus, a demand has risen for efficient tools for searching, browsing and effective sharing of the multimedia data [10][11].

Previously, Multimedia Retrieval Systems were described in form of a dataflow diagram (DFD) which was less suitable for productive information retrieval because it separated data and the operations that modify the data. The addition of a new data component or a new operation introduced a number of scattered changes in the diagram, which complicated the development environment. The development and molding of the retrieval system according to the needs of the user and the application was a complicated.

A distributed multimedia retrieval architecture called CMRS, Content Based Multimedia Retrieval by Timo, Hannu *et al.* [1] is instigated, which can easily be extended to include new media types and retrieval operations without modifications to existing features. The principal idea of object orientation such as encapsulation of data and operations into a single unit has been used. The system is cleaved into a media independent platform and modular media specific extensions, which contain encapsulations of data (physical media items and operations to analyze the media items) and queries (a query definition and operations to perform queries with this particular query definition). This division gives a great flexibility in extending the system with new media types and/or operations.

Multimedia Connection Management by Eve and Stephen [2] architecture allows geographically separated individuals to collaborate by combining real–time packet audio and video with shared computer workspaces, sometimes called *groupware*.

II. THE CMRS ARCHITECTURE

CMRS represents the distributed client/server-architecture that comprises of a fully media independent platform and media specific extensions as shown in Figure 1a. The platform provides services such as basic client-server communication and distribution, and storing and retrieving capabilities of different data objects, which makes the platform simple in terms of user interface, search methods and data model. As data and query components are independent and have their own user interface new components can be added without modifications to the existing platform. The Principal components of the CMRS architecture described using UML notation is shown in the following figures.

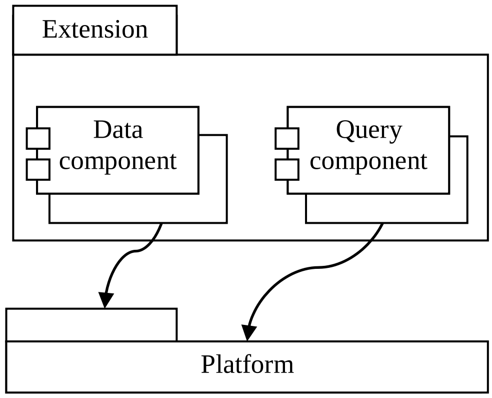

Figure 1: CMRS architecture comprises of a media independent platform and media specific extensions, which contain one or more *Data* and *Query* components

A media specific extension consists of one or more Data and Query components, which are depicted in Figure. 1b. A Data component encapsulates physical media items and operations to analyze the media items to generate the associated metadata, which are stored in the database. A Query component stores the specification of the query and operations to perform the query independently of the database. Both the Data and Query component also have an own user interface, with which they can visualize themselves, hence the general structure of the component lends to the well-known MVC (Model-View-Controller) paradigm. In the case of the Query component the Query Editing User Interface is used to provide Query Definition Data. Runtime view of a typical system configuration is presented in Fig. 1c. Query and Data components denoted by *q* and *d* are transferred between Client and Server using object serialization mechanism provided by Java.

The platform consists of four modules: *Server*, *Client*, *WWW-Gateway* and *WWW-Client*.

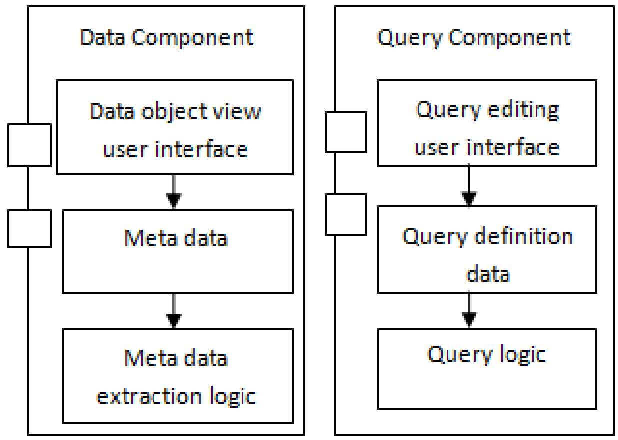

Figure 2: *Data* and *Query* components encapsulate the data, user interface and operations related to physical media items and queries, respectively.

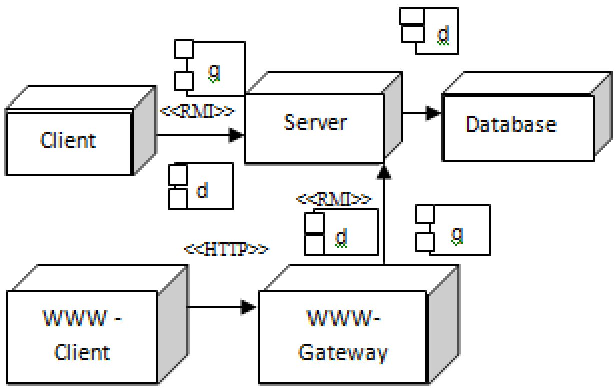

Figure 3: Runtime view of the platform .

The Server authenticates users, process queries and offer database services to Clients. Database Service handles all connections to the database and offers database management interface for Clients and Data components. Query Services is in charge of query processing and offers the query-processing interface to the Client(s) and Query components. Feature Extraction Service controls metadata extraction, so that the Client(s) can automatically distribute the metadata extraction of Data components over available computing resources.

Client consists of four modules: *Database Manager*, *Query Definition Tool*, *Annotation Tool*, and *SOM Browser*. *Database Manager* is a user application that provides user interfaces for browsing and managing the database. *Query Definition Tool* is a user application, which is used to define content-based queries. User can define queries by adding Query components to a query definition tree and then defining Query Definition Data for each Query component with the component specific *Query Editing User Interface.*

SOM Browser is a user application, which provides a hierarchical 2-dimensional view on a database. The view is realized with a Self-Organizing Map [3], which upon unsupervised training derives a nonlinear mapping from the source feature space onto a 2-dimensional lattice preserving the topological ordering of the data. A SOM is represented in form of neurons, which have a fixed location in the 2 dimensional lattice and a weight vector, which corresponds to a location in the original feature space. On training the weight vectors they are tuned so that each neuron represents a portion of the data, and their weight vectors designate the rough location of this portion of data in the original feature space. Given an object in the original feature space, its location on the map can simply be determined by searching for the neuron, which has the most similar weight vector to the feature vector of the object. Since SOM preserves the relative topological ordering of the weight vectors and the neurons, those objects that are located near each other in the original feature space will be located in nearby or same neurons in the map.

Annotation Tool is used to have relevant cues for queries, it is crucial to extract semantically meaningful elements from the database images. They are used outwardly in queries to ensure fast-access to the images holding desired semantic information. A step in the vicinity of the solution is to develop a semi-automatic tool for semantic image annotation that promotes the image analysis as a supporting function to the annotation process.

The CMRS architecture offers retrieval and browsing possibility also in the WWW using HTML and Java based user interfaces. The server side of the WWW interface is implemented using Java servlet technology and a three-tiered architecture (*WWW-Client* <> *WWW-Gateway* <> *Server*).The *WWW-Gateway* implemented by servlets acts like *Client* from *Server's* point of view, converting *Server's* RMI-communication to HTTP understood by *WWW-Clients*. *WWW-Client* is a simplified version of the standard *Client*, supporting only image queries based on example images and sketches drawn by the user.

III. CONNECTION MANAGEMENT ARCHITECTURE

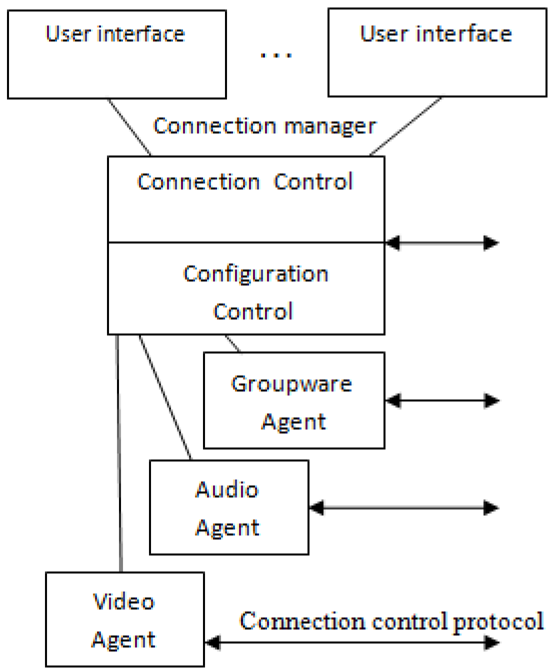

Figure 4: Co-ordinated management of separate service.

The proposed multimedia connection architecture allows the distributed individual components to associate by combining the real-time packet audio and video with shared computers. The connection manager manages the multiparty, multimedia connections. Theoretically, connection manager is separate from user interfaces (UIs) to the system, which is above it offering services up to the user and relaying requests down from the user. The connection manager is also separate from underlying components, called media agents, which handle communication protocol decisions and the devices specific to each type of shared media (audio, video, groupware). This organization allows the connection manager to convey timing information between media agents for inter−media synchronization.

The layered organization is referred as the connection management architecture, since management of connections is the primary focus of the model and since the connection manager acts as the conduit through which control information flows between layers [6-9].

A. The Connection Control Protocol

The Connection Control Protocol (CCP) is an application layer protocol used by connection managers to communicate among themselves. The CCP is a multicast, transaction−based protocol which aims to provide reliable, group communication and to accommodate variability in request−reply response time due to WAN operation and to heterogeneous end system configurations [5].

Conference orchestration is achieved through use of a distributed, peer−to−peer model. The connection manager acts as both client and server, notifying users of requests from other connection managers, and placing requests to other connection managers on the local user's behalf. The connection manager first communicates remotely with peer connection managers during conference setup, via CCP, which then communicates locally with the media agents via well−defined call interfaces to create the underlying voice, video and groupware data flows.

The four−phase conference setup process is the main task during which the initiator is responsible for negotiating a common set of capabilities, requesting participation, initiating media connections and propagating information among peers; it acts as a leader until conference creation completes. The CCP has a built−in mechanism that acts as a transparent conduit for other types of operations between user interfaces and media agents. These operations might occur during or outside of a conference, without CCP having to know the particulars of each operation.

B. Configuration Management

Configuration management is the need to communicate configuration information and to implement service selection so that end systems supporting different combinations of services can still be interoperable. Several mechanisms have been advocated to deal with heterogeneity such as, a configuration language, a distributed resource locator service, and a resource synthesizer. A configuration language describes resources and devices located at each end system. This language will categorize the set of services offered by each system, and must be extensible to accommodate new services and devices as they are added. The system retains a mapping of configuration attributes to media agents, allowing the connection manager module to choose an appropriate agent or agents to meet configuration requirements made by applications. The resource synthesizer will synthesize solutions to configuration requests [4]. The resource synthesizer must find a sequence of communication mechanisms and intermediate translators to construct paths among all the end systems in the connection.

IV. PROPOSED METHODOLOGY

This paper collaborates CMRS architecture, which comprises of a media independent platform providing basic functionality, media specific extensions providing enhanced analysis and retrieval functionality for particular media types with a Connection management architecture which contains a connection control protocol, and configuration management. Combination of both these architectures effectively supports all media types and is extensible to support major functionalities such as search, browse, share, transfer and retrieve.

V. CONCLUSION

The current architecture unfolded in this paper is a combination of two main design principles i.e. CMRS (Content Based Multimedia Retrieval Architecture) and Connection Management Architecture. Query results which are based on the contents are executed at a faster pace. Audio and video data are ferried among the users in productive network environment using connection management architecture so that the users are never detained. This system is completely extensible and supports the transfer of all the commonly used forms of media such as images, audio and video effectively in real time.

VI. ACKNOWLEDGMENT

We would like to thank Assistant Professor Mr. Alok Ranjan for his great supervision as we hurdled through all the obstacles while completing this work. This Research Paper would not have been possible without his guidance and his valuable assistance in the preparation and completion of this study.

VII. REFERENCES

[1] Timo Ojala, Hannu Kauniskangas, Heikki Keränen, Esa Matinmikko, Markus Aittola, Kimmo Hagelberg, Mika RRautiainen and Mikko Häkkinen, "CMRS : Architecture For Content-Based Multimedia Retrieval".

[2] Eve M. Schooler,Stephen L. Casner," An Architecture for Multimedia Connection Management", Proceedings IEEE 4th Comsoc International Workshop on Multimedia Communications,MM '92, pp.271−274, Monterey, CA (April 1992).

[3] Kohonen T," Self-Organizing Maps (3rd edition)". Springer-Verlag, Berlin (2001).

[4] Nicolaou, C., "An Architecture for Real−Time Multimedia Communications Systems", IEEE Journal on Selected Areas in Communications, Vol.8.No.3, pp.391−400 (Apr 1990).

[5] Schooler, E.M., "The Connection Control Protocol: Specification", Version 1.0, USC/Information Sciences Institute, Marina del Rey, CA (Jan 1992).

[6] Bates, P.C., Segal, M.E., "Touring Machine: A Video Telecommunications Software Testbed", Proceedings First International Workshop on Network and Operating System Support for Digital Audio and Video, Berkeley, CA (Nov 1990).

[7] Leung, W.H., Baumgartner, T.J., Hwang, Y.H., Morgan, M.J., Tu, S.C., "A Software Architecture for Workstations Supporting Multimedia Conferencing in Packet Switching

Networks", IEEE Journal on Selected Areas in Communications, Vol.8, No.3, pp.380−390 (Apr 1990).

[8] Vin, H.M., Zellweger, P.T., Swinehart, D.C., Rangan, P.V., "Multimedia Conferencing in the Ether phone Environment", IEEE Computer, Vol.24, No.10, pp.69−79 (Oct 1991).

[9] Garcia−Luna−Aceves, J.J., Craighill, E.J., Lang, R., "An Open−Systems Model for Computer−Supported Collaboration", Proceedings 2nd IEEE Conference on Workstations (Mar 1988).

[10] Hirzalla N and Karmouch A,"A data model and query language for multimedia documents databases". Multimedia Systems 7:338–348 (1999).

[11] Zhang C, Meng W, Zhang Z and Wu Z ," WebSSQL-A query language for multimedia Web documents", Proc. IEEE Advances in Digital Libraries, Bethesda, MD, 58–67 (2000).