Web-based Multimedia Applications Design Using the MOR

Bingyang Zhou
Department. of Maths & Computing
University of Southern QLD

Mao Lin Huang
Department of Computer Systems
The University of Technology, Sydney

Abstract

This article presents a study of the Web-based multimedia applications design using the Multimedia Object Repository (MOR). The basic structure of the MOR was described in [5]; this paper focuses on the structure of the graphics object building block and text object-building subblock. The MOR provides the ability to create sophisticated linkages between multimedia documents from a variety of web applications as well as providing a development framework for creating additional Intranet applications with that capability.

Keywords: web-based information systems, MOR, multimedia objects

1. Introduction

Multimedia systems and applications are very exciting tasks through the Internet and World Wide Web. Multimedia system and application are influencing many areas, including business/industry services, distance education/learning, entertainment, scientific research and various services. Multimedia enhances the overall quality and quantity of information; most areas of information technology are embracing the emerging multimedia platforms, data types, and environments [1]. The Web was a medium for sharing documents among people and computers via the Internet. It is faster becoming a distributed development environment capable of providing information and other application resources to millions of people around the world. Corporations are implementing internal Webs and Web-based multimedia application such as Intranets, to facilitate communications and sharing of information among employees. Web-based multimedia applications are springing up in organisations literally overnight – in many cases changing the way they conduct their businesses [2]. The Multimedia Object Repository (MOR) is a powerful tool, which can be used to design the web-based multimedia systems and applications. The MOR is a data store with a set of applications. The MOR can be accessed via a query language, java language or programming API. It can be accessed using instructions typed at a command line or through interfaces that custom designed for use on a specific platform. You can design your own applications solely for the purpose of querying the MOR and returning specific information about a multimedia object. This ability integrates a database into applications that can be accessed by users using any kind of web systems.

Most organisations maintain a variety of autonomous computer databases that support basic infrastructure needs, such as human resources and classic information systems, such as corporate management information systems. There are several reasons organisations might want to employ the MOR in their web-based systems design. In fact, in most web-based multimedia applications, the MOR can serve as the basic building blocks for web applications design such as, manage the serving of large document-based information repositories to internal and external users of the system; leverage and use legacy database systems, the information they contain, and existing applications; unlock the potential of unused information held within organisational databases; extend the functionality of web server so that user can make information maintain available to the general public and internal users.

2. Multimedia application design and development structure

Web-based multimedia application design entails planning and the conceiving of ideas that address certain needs. In many cases, especially with web systems design, it entails the confluence of many competing issues, the integration of efforts by personnel with multiple technical disciplines and backgrounds, and dealing with the ever-present problems of obtaining sufficient levels of resources to address the tasks at hand [2]. Web application design includes inputs, outputs, environment, processes, workflows, and transformation of inputs to outputs. It necessarily incorporates the contributions and effects of people in the system, including end-users, developers, managers, and technical staff. Additionally, multimedia application design looks at the existing problems or opportunities, develops requirements and goals to address those problems and opportunities, proposes various alternative methods to implement solutions, and then examines the operational, technical, and economic feasibility of proposed alternative solutions. The MOR serves an integral role in many applications being developed for customer use and to improve the services provided to them. Among the uses of the MOR in web-based applications are: technical product information and knowledge bases that enable users to easily search for and obtain multimedia object specifications such as, size, data types, supported softwares, and troubleshooting information.

3. Graphic design unit

It is difficult to provide hard and gast rules for graphic visual design; certainly the site's visual design will be determined by the company's current marketing collateral, issues of personal taste and the limitations of the medium. Corporate web sites must promote and reflect the company's brand identity [3]. If the web site looks one way and the print literature looks entirely different, the company is promoting an inconsistent image. In addition, the graphic design should help communicate the message, not just decorate it. While market branding of the company drives the design of an Internet site, an extranet site may involve the brand of another company as well [4]. The design of an extranet site typically needs to convey a partnership between two or more companies. Consequently, the site may have a look and feel similar to the main company's web site, but also include the logo and company colours of the partner. In this way, the design is driven both by marketing and by the user's desires. An intranet site is similar but may pay even less attention to corporate image. Users of an intranet typically are not interested in graphics, nor does the brand identity of the company need to be continually reiterated in a creative fashion. Rather, the user's desire for simple, fast loading pages will drive the design. The MOR will provide the function to optimise the multimedia graphics and reduce the loading time of the page.

Figure 1 shows one of the building blocks of the MOR, graphics building block. A client program or an interactive user can create, delete, copy, cut, position, resize, and move graphic objects. The graphic building block in the MOR was designed to provide a programming utility that would allow an implementation to create program with functionality of drawing objects.

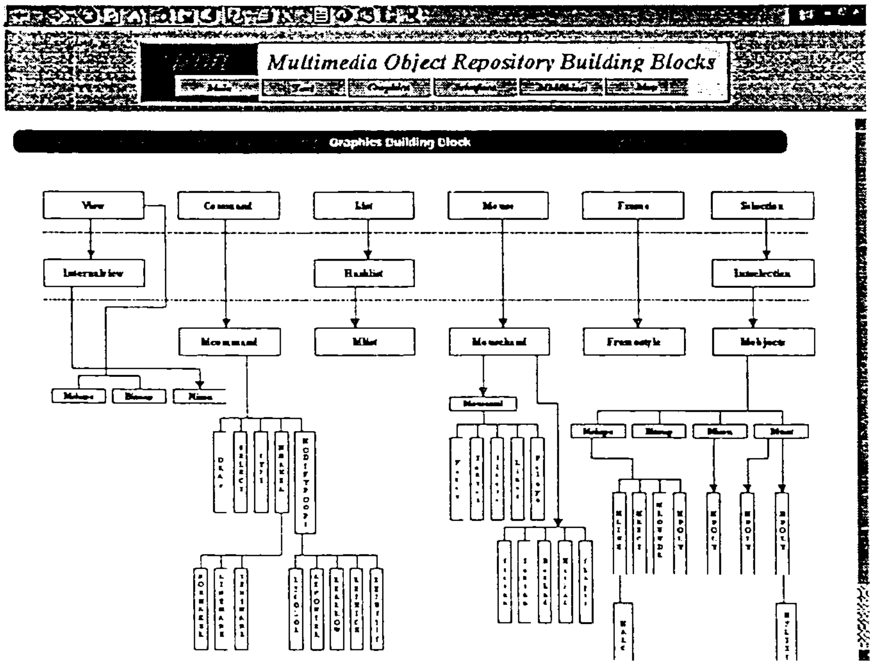

Figure 1 Graphic Building Block of the MOR

The building block includes the following functions [7]:

- Drawing – define functions for drawing (draw, paint, hilite, outline, offset, inset, map etc.), for intercepting (select, deselect), for storage and retrieval. Some of these functions implemented by put icons in front of block include a rectangle indication the primitive's bounding box, a unique id that is, assigned to each multimedia object for storage and retrieval, the order in which this object should be drawn in relation to others, and a flag indicating if the object is selected.

- Shape, Text, Icon, and Bitmap – describe a particular set of behaviours (mostly behaviour refined from Drawing). Shape adds additional field to keep track of fill style, pen style, and line width.

- View – maintains the additional information needed to render a view based on the graphics building block. View adds fields to keep track of the current drawing tool, to point to the current palette being used by the drawer, and most importantly, to point to the list of graphics objects that is maintained by a list. Since the graphics building block intends to provide capabilities to both draw and edit graphical primitives, it must provide facilities for adding and deleting things from the view. The list is designed to maintain a list of heterogeneous objects. The list implements protocol that allows a single operation to be performed on every member of the list. The View maintains one list that contains all the objects in the canvas and maintains another list that contains all the objects in the current selection. View has the ability to perform standard rendering operation and refines many of the behaviours to reflect the needs of the graphical view – Draw, for example, traverses the list to display objects. Most importantly, View adds methods to enable a client to add an object to the view, delete an object from the view, perform an action on all the objects in a view, and turn the entire contents of a view into a format for writing to disk.

- Motion-video – As full motion video is the most processing and storage intensive components. It needs large bandwidth for communications media, massive storage requirements, and high-density high performance compression technologies.

The Mobject is the main control block that is at the top of all editable graphics objects. Mobject define functions for drawing, interacting, storage and retrieval. Some of these functions are implemented at the Mobject level, while others are expected to be implemented by subblocks. Fields of Mobject include a rectangle indicating the primitive's bounding box, a unique id that is assigned to each graphic object for storage and retrieval, the order in which this object should be drawn in relation to others, and a flag indicating if the object is selected.

Subblocks of Mobject include Mshape, Mtext, Micon, and Mbitmap. The first two of these subblocks describe a particular set of functions (mostly function refined from Mobject) that will be refined in subblocks. Mshape adds additional spaces to keep track of fill style, pen style, and line width, since all subblocks of Mshape are expected to use those spaces. Mobject functions Paint, Select, Deselect are overridden, while a SetPen function is added. Mline, a subblock of Mshape, adds fields that keep track of the endpoints of the line and a field that indicates whether the line has arrows on its endpoints. The other Mshape subblocks, Mrect, MroundRect, Mpoly, and Marc refine Mobject and Mshape to a greater or lesser extent. Similarly, Mtext has two operative subblocks, Mstatictext and Meditabletext that define the specific functions indicated by their name. Micon and Mbitmap are typically used to create objects of the primitives they represent. Subblocks of Mousehand are declared to handle the interaction for each of the multimedia object.

Mview maintains the additional information needed render a view based on the graphics building block. Mview adds fields to keep track of the current drawing tool, to point to the current palette being used by the drawer, and most importantly, to point to the list of graphics objects that is maintained by an object of Mlist. Since the graphics building clock intends to provide capabilities to both draw and edit graphical deleting things from the view. The Mlist is designed to maintain a list of heterogeneous objects, each an object of some subblocks of Mobject. The Mlist implements protocol that allows a single operation to be performed on every member of the list. For example, the function of Mlist would invoke the Offset function of every element in its list to cause it to move appropriately. The Mview maintains one list that contains all he objects in the canvas, Mselection, a subblock of Intselection, maintains another Mlist that contains all the objects in the current selection. Lview inherits the ability to perform standard block/link operations and standard rendering operations. Lview refines many of the Intview functions to reflect the needs of graphical view, Draw, for example, traverses the Mview's Mlist to display objects. Most importantly, Mview adds functions to enable a client to add an object to the Mview, delete an object from the Mview, perform an action on all the objects in an Mview, and turn the entire contents of a view into format for writing to disk.

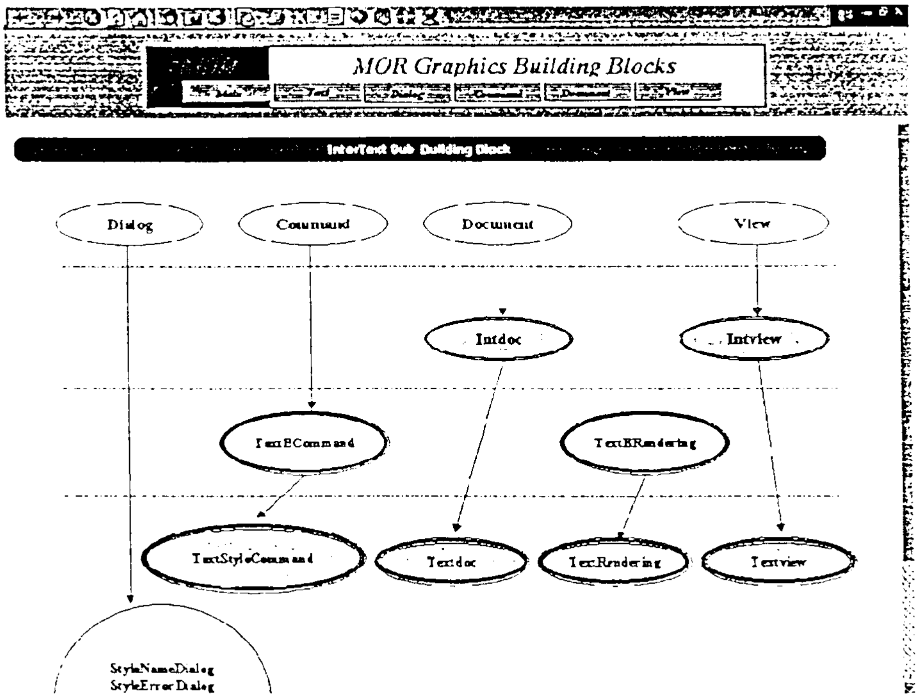

Figure 2 Text Building Block of the Graphic Unit

4. Text building block

The text unit (see figure 2), a major client of the text building block, contains the additional functions necessary to support the InterText processor. The major text unit objects are the text document and the text view subblocks of IntDoc. TextDoc adds an extra field, the most important of which is a handle to the object of the text building block that this document will be using. Similarly, TextDoc creates and initialise the appropriate text building block objects when a new InterText document is requested. The remaining functions defined in the text unit represent dialog boxes and a few additional command objects. The text unit serves as a user interface wrapper around the building block.

To avoid the conflicts between graphics and text objects, we need to test the text unit standard alone. A TextAppl subblock is created. The TextAppl subblock differs from Intblock in only one major way. In Intblock, the Makedocument function brings up a dialog box that allows the user to choose among several different document types to launch when the new menu item is chosen, in TextAppl the Makedocument function can only create the document within Textdoc and Textview subblocks.

To bind the units together into a coherent integrated multimedia application, MOR provides a FrameApp block. This block comprises a merge of all of the fields, functions for each subblock, plus a few additional functions particular to the integrated world itself. The FrameApp consists of merging the functionality of each Makedocument function of the standalone applications into one routine. Where each standalone function simply created a document of a specific type, the new Makedocument contains a choose function that chooses what document to crate based upon the parameter that was in. Creating an integrated application from a set of standalone units took only a few seconds.

5. Conclusion

We have addressed the graphic building block and text building subblock structure in MOR. Integration of all applications consists of two major themes, the appearance of the applications and the ability of the applications to exchange data. One of important contribution of the MOR is to increase the availability of all types of multimedia objects for the designing of web-based multimedia applications. As mentioned in [6], Using MS Access relational database system, the Multimedia Object Repository is vitally important to the application architecture design, the user interface design and graphics manipulation.

Acknowledgments I would like to thank Andy and Desley Zosars for thoughtful suggestions on drafts of this paper.

References

[1] Setrag Khoshafian and A. Brad Baker. Multimedia and Imaging Databases. Morgan Kaufmann Publishing, Inc. San Francisco, Califomia 1996.

[2] Mark Swank and Drew Kttel, "World Wide Web Database Developer's Guide", Sams.net Publishing, Indianapolis, IN USA, 1996.

[3] Prabhat K. Andleigh and Kiran Thakrar. Multimedia Systems Design. Prentice Hall PTR., Upper Saddle River, NJ 1996.

[4] Thomas A. Powell, with David L. Jones and Dominique C. Cutts, "Web Site Engineering – Beyond web Page Design", Prentice Hall PTR Inc. USA, 1998.

[5] Bingyang Zhou, Multimedia Object Repository for the Web Based Multimedia Application Design, Intemational Conference on Intemet Computing (IC'2000). June 26th - 29th, 2000 Monte Carlo Resort, Las Vegas, Nevada, USA.

[6] Microsoft Corporation. Building Applications with Microsoft Access 97. Microsoft Corp. January 1997.

[7] B. Zhou and G. Chen, Design Consideration Using the Multimedia Object Repository, 36th Intemational Conference on Technology of Object-Oriented Languages and Systems (TOOLS 2000), 30 October - 4 November 2000, Xi'an, IEEE Computer Society 2000