A Visual Query Facility for Multimedia Databases[*]

Ghada El-Medani, M. Tamer Özsu, Duane Szafron, Chiradeep Vittal
Laboratory for Database Systems Research
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1
{ghada, ozsu, duane, vittal}@cs.ualberta.ca

Abstract

This paper presents a visual query facility prototype developed for News-on-demand, a multimedia news application. The graphical user interface provides three major facilities for users of this application: presentation, navigation and querying of multimedia news documents. The main focus, however, is querying of multimedia objects stored in the database.

1. Introduction

One of the major reasons for the recent popularity of multimedia information systems is the opportunities they provide for easier and more effective human-computer interaction (HCI). The significance of multimedia systems lies in enhancing the communication between computers and human users. They incorporate and integrate information from diverse media sources, such as text, images, audio and video, presenting the users with various channels for communication and information delivery. This increases system usability as humans communicate more effectively through various channels [BD92].

The media sources, incorporated in multimedia documents, involve large data objects with complex spatial and temporal relationships. This results in a large body of information which comprise the multimedia database. For the user to easily and effectively access and make use of this large information space, there is a need for an interactive user

[*]This research is supported by a grant from the Canadian Institute for Telecommunications Research (CITR) which is one of the Networks of Centres of Excellence funded by the Government of Canada.

interface to act as an intermediary between the database and the user. Thus, a considerable portion of achieving the usability and effectiveness desired from multimedia lies in being able to provide users with easy-to-use effective interfaces.

Multimedia systems involve a significant number of issues that span several fields of computing science including networking, databases and human-computer interaction [Ada93]. Our work aims at studying two aspects of the development of distributed, interactive multimedia applications: the logical database modeling of multimedia objects and the relationships between them, and a visual querying facility to allow users to access the multimedia database through queries as well as a browsing facility. In this paper, we concentrate on the visual user interface; the database design is reported elsewhere [VÖSEM95].

Our target application is news-on-demand, which we describe further in the following section. The client application environment includes our visual querying facility, the synchronization routines [LG94], a quality-of-service negotiation component [VDB+93], and the ObjectStore [LLOW91] database client. The clients are connected to a set of servers over a broadband ATM network. The servers are responsible for the storage and management of multimedia documents.

Many multimedia information systems provide a browsing-based user interface, but have limited querying capabilities. Perhaps, the most popular interface of this type is Mosaic for accessing the World Wide Web. Along the same lines, a model for hypertext-based information retrieval is presented in [Luc90]. In [Wil91], a hypermedia system for on-line technical documentation is described. It provides a browsing facility and supports limited keyword search. There are many other hypermedia systems which provide elaborate browsing facilities but they have limited querying capabilities [GT94, HKR+92, Pla91] . The system described in [FS91] supports storage and retrieval, as well as browsing of multimedia information. These functions are achieved through hyperlinks between related information and a searching facility. Queries are handled using a functional database query language (FQL*) which is extended to include elementary similarity functions to allow for imprecise queries.

In contrast to these systems, our interest is mainly in querying capabilities. Eventually, we would like to enable the user to pose queries such as "*Show me all the news items on the Israeli-Egyptian peace treaty in which President Sadat shakes hands with Prime Minister Begin and in which President Carter comments on* ...". We are a long-way away from this goal which requires, among other things, content-based indexing of multimedia objects. We have, as a first step, developed a visual query facility to access a multimedia database for news-on-demand applications. The novel aspects of our approach are the following:

- A combination of querying and browsing.
- Tight integration with a multimedia database.
- Use of a visual query interface to relieve users from typing complex database queries.
- Separation of the logical document content from presentation formats.

The rest of the paper is organized as follows. Section 2 presents the news-on-demand application. Section 3 discusses the characteristics and design requirements of multimedia user interfaces, specifically addressing user requirements with respect to the news-on-demand application. In Section 4, we discuss the system prototype, highlighting its functionality, with an emphasis on querying capabilities. Section 5 talks about the link between the visual query facility and the database. The conclusion and future work follows in Section 6.

2. The News-on-demand Application

2.1 Overview

News-on-demand is an application that provides its subscribers/users with access to multimedia news documents that reside in a distributed database. Multimedia documents are inserted into the database by news providers who are responsible for gathering news and organizing it in multimedia documents. Examples of such organizations are television networks, newspapers, magazines, and wire services. Once they are in the database, the multimedia news documents are not updated. At different client sites, the subscribers access the news database via a broadband network. The readers pay a cost for this service. The cost includes the cost of the information content as well as the transmission and retrieval costs. The users can affect the cost they must pay by specifying quality-of-service parameters like the resolution of images and video.

In this context, a multimedia document is a structured collection of pieces of information related to a particular subject [VÖSEM95]. The information units can include any media type such as text, images, audio and video. They can also include combinations of different media, such as video synchronized with audio and text captions. Documents often have links to other information (whether complete documents or components of documents). Examples of such links are more news coverage, background information, and expert analysis.

The news-on-demand application suggests the following issues:

- There is a need for a common database document representation so that documents and their standard components can be retrieved by regular queries. All news providers must convert their documents to this common representation whenever they insert documents into the database. Since many authoring systems may be used by news providers, the document representation should follow some recognized standards. We have chosen to follow the SGML/Hytime standards [VÖSEM95].
- It may be necessary for subscribers to customize their system view due to personal preferences and/or hardware constraints (e.g., absence of a graphics display).
- Subscribers must be made aware of the costs incurred for retrieval and transmission of multimedia news documents. Cost information should be stored in the database and made available to users through the quality-of-service component.

2.2 User Requirements

In this subsection, we discuss the requirements imposed on multimedia user interface design. We start with some general requirements and then focus on the requirements emerging from the news-on-demand application.

Since multimedia systems have a lot of potential in delivering information to the users, special attention should be given to the actual usability of these systems. A system's usability is determined by how easily and effectively the users can use and communicate with the system. Usability parameters for most systems include ease of use, efficiency, ease of remembering and pleasantness [Nie90a]. Some general design requirements for usable multimedia user interfaces are [Hay93]:

- Simplicity: A multimedia user interface should be simple to use. Overwhelming the user with complex icons, too many choices and a multitude of color patterns (the trend in some current multimedia interfaces) may distract the user from the information being presented.

- Consistency: Providing a consistent design throughout the interface makes it easier and more predictable for the user to follow. It reduces the overhead that the user must make in trying to understand different parts of the interface. Again, this allows the user to concentrate on the actual information being presented.
- Engagement: Engagement is determined by the degree to which the user can participate, affect and control the actions of the system. "*Multimedia should invite the user to participate*" [Hay93]. Interactive interfaces provide added value by allowing the user to stimulate the system and get an immediate response.
- Depth: A multimedia interface should encourage users to explore the system to a greater depth by making it easy to do so. However, it should not force the user to understand the system to any greater depth than the user wishes to explore.
- Fun: Multimedia user interfaces should be fun. This will encourage more people to use the system and each person to use the system in more ways. Note that the goal is not to encourage people to waste time using the system. The goal is to encourage them to use the system in novel but productive ways.

Although these general design requirements apply to all multimedia applications, there are more specific requirements for the news-on-demand application. Users of news-on-demand are expected to use the system to achieve several tasks related to the news. These tasks include analysis of financial and political situations which can affect planning, regular follow-up of news, and access to background and reference news material. The general high-level functions required of the system can be summarized as follows:

- Browsing/Viewing information: Users should be able to view multimedia documents by reading text, looking at images, playing video, listening to audio and following links to related information.
- Searching for information: Users should be able to search the news using a variety of criteria such as date, author, subject, location, and, most importantly, its content. The system should provide a fast and easy way for searching and an efficient mechanism for displaying search results and accessing their components.

- Customizing the system: Users should be able to define and modify system settings. Settings should include: document layout, screen layout, window specifications, quality of service parameters and others.
- Other functions: These include: allowing users to add their own annotations to news documents, providing users with additional navigational aids such as maps and subject indexes, and providing users with a history of the visited documents.

These requirements point to a customizable and easily extensible interface which combines the browsing capability (found in most existing multimedia interfaces) with a querying capability (lacking in many of the same interfaces). Furthermore, the querying capability should be a visual one that merges seamlessly (is consistent) with the browsing facility and satisfies the other general usability requirements of a multimedia user interface.

3. Our System Prototype

3.1 Basic Design Principles

We have examined the various requirements and design goals for multimedia user interfaces in general and for the news-on-demand application in particular. According to these requirements, we can categorize our design choices by three major points:

- **Hypermedia**

  Hypertext/Hypermedia provides the user with a non-sequential means of freely browsing information according to individual need. This is accomplished by following links from one information unit to another [Nie91a]. Some of these systems also provide navigational aids such as subject indexes, maps, history of visited nodes, etc. to facilitate browsing.

  Not all multimedia systems use hypermedia links [BD92, Hay93]. However, in the news-on-demand application, documents often have links to other related data such as background information, more news coverage, and expert analysis. Therefore, a hypermedia interface is a good design choice as it provides the news readers with an easy and efficient way of accessing and browsing related information.

- **Query Mechanism**

  A hypertext/hypermedia interface to a multimedia system may not be sufficient to provide all of the accessing mechanisms the user needs to obtain information from the database. In many applications, such as news-on-demand, users need to search for specific information based on partial knowledge. This must be accomplished more simply and quickly than is possible through the browsing facilities of hypermedia. Moreover, as the information increases in quantity and complexity, the browsing facility of hypermedia becomes more and more inadequate. According to studies of the usability of hypertext systems, users have often reported that they become disoriented while navigating through hypertext systems and fail to reach points of interest. This phenomenon was reported even when using the most popular commercial hypertext systems [Nie90b, Nie91]. For the news-on-demand application, a querying facility that allow users to search and retrieve information directly from the database is a good design choice. The user interface should provide an easy way for performing queries and searches, and examining the results. A typical querying scenario is for the user to first filter the list of documents in the database to include only relevant articles. The user then performs a search for documents and document components that are of interest for specific topics and then browses this limited set. The design must support this incremental process.

- **Input Modalities**

  A "good" user interface should provide the user with appropriate interaction modes, depending on the application and the types of input needed from the user. We can categorize the modes of interaction into three major categories [BD92]:

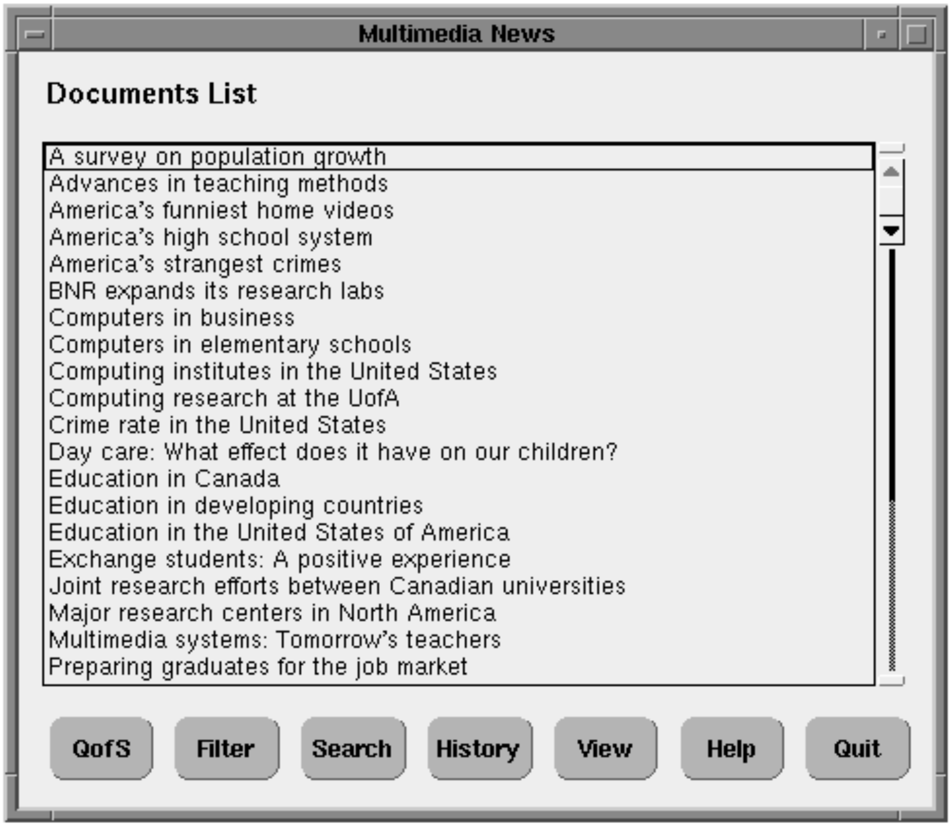
  - Direct manipulation of graphical objects on the screen.
  - The use of natural language.
  - The use of formal languages.

  In case of the news-on-demand application, a graphical user interface, with direct manipulation techniques, is sufficient to deal with user input. Typically, users need to click on icons to follow links, choose options from menus and lists, type text, etc. A natural language interface would be a great facility if provided with the graphical interface. However, this is beyond the scope of our research. We have eliminated the need for formal languages (in our case, a query language) by designing the visual query interface to make the system more usable, especially by novice users. Our design choice then is to use a visual query interface as a front-end to the ObjectStore database management system's query language.

3.2 Overview

The visual interface we have developed for the news-on-demand multimedia information system provides a number of functions (Figure 1):

- Initiate a quality-of-service negotiation (**QofS**).
- Start a filtering operation in the database (**Filter**).
- Perform a search (**Search**).
- Display a list of visited documents (**History**).
- Retrieve and display a document (**View**).



A survey on population growth
Advances in teaching methods
America's funniest home videos
America's high school system
America's strangest crimes
BNR expands its research labs
Computers in business
Computers in elementary schools
Computing institutes in the United States
Computing research at the UofA
Crime rate in the United States
Day care: What effect does it have on our children?
Education in Canada
Education in developing countries
Education in the United States of America
Exchange students: A positive experience
Joint research efforts between Canadian universities
Major research centers in North America
Multimedia systems: Tomorrow's teachers
Preparing graduates for the job market

Figure 1: Documents List

The search and filter operations result in a subset of the news articles to be displayed. A user may conduct further searches on this subject. Any of the news articles may be opened by clicking on the textual description of that document.

Once a document has been opened, the user can access related material by clicking on an anchor to follow a link. The representation of anchors can be specified by the user (underlines, boxes, icons, etc.). In many multimedia systems, a link can only have a single destination. However, we take the more general approach of allowing multiple destinations where each destination may be a whole multimedia document or a component of a document. The user can navigate between documents by following or returning from links. More importantly, the interface allows the user to perform a search from anywhere within a document and to specify the scope of the search. This scope may be the current document, the documents linked to the current document, or all documents in the database.

Another important feature of the interface is allowing the users to customize the presentation. Customization ranges over a variety of system parameters, such as defining the presentation of anchors within a document, attaching a font format (such as italics, bold, reversed video, etc.) to an emphasized text, and defining different layouts for the same document. Other examples of customization include immediate playing of a video when the icon is clicked versus opening up a control panel (similar to a VCR) to allow explicit activation of the video. The user can also define a default document filter, quality of service parameters, etc. for system startup. All user preferences, for the system and presentation issues, are stored in the database as *style sheets* and/or *user profiles*.

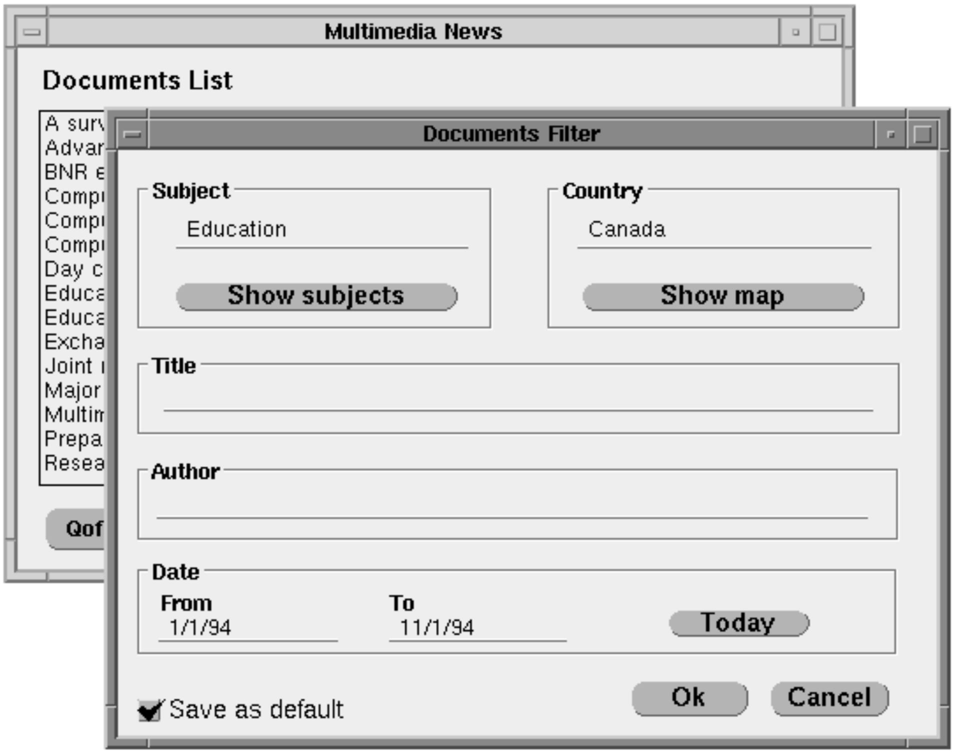
The visual query facility prototype is not intended to provide a complete user interface that adequately covers all needed functionality. Instead, its main purpose is to provide users with an easy and efficient way of accessing the multimedia news database. The prototype provides most of the functionality highlighted in the user requirements. Although the emphasis is on searching and filtering, an adequate browsing facility is also provided.

3.3 The Querying Interface

Since our main emphasis in this paper is querying capabilities and their tight integration with the database, we will discuss them in some detail. The full interface functionality is presented elsewhere [EM95]. Three classes of querying capabilities are provided: filtering documents, searching for documents, and searching from within a document

Filtering Documents

The user can specify the scope of the documents displayed in the document list by using the **Filter** option (Figure 2). Filtering is a search on document attributes. The result of the filter is a set of documents whose attributes match the ones specified by the user. Filtering of documents can be performed based on subject, country of publication, title, author, or date. Other attributes may be added by storing them in the database. The settings of any filter can be saved (in the user profile) as the default filter which is used during system startup, but filters can also be applied to the document list at any time.

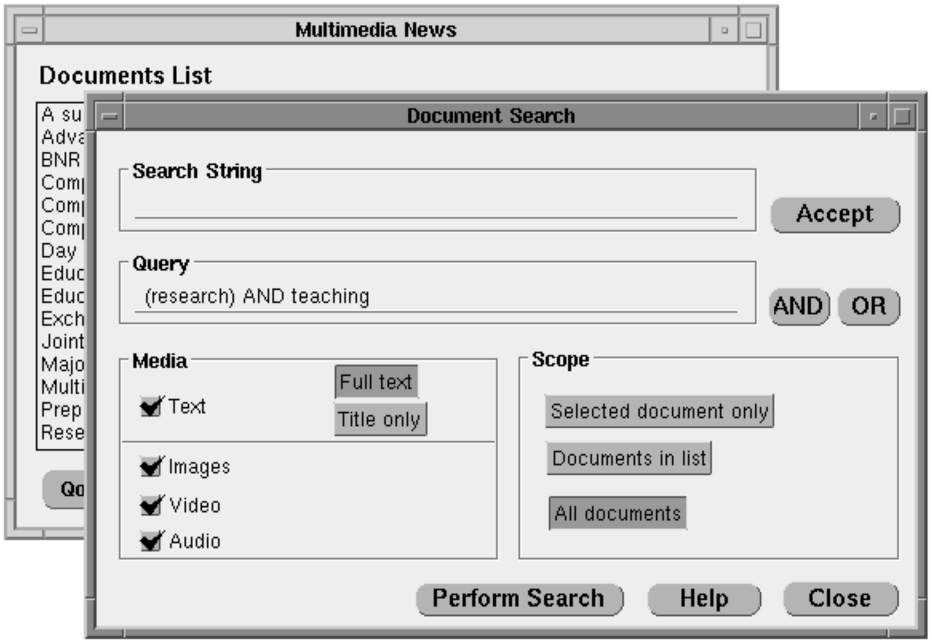


Subject: Education
Country: Canada
Title:
Author:
Date From: 1/1/94 To: 11/1/94
✓ Save as default

Figure 2: Filtering Documents

Searching Documents

The user can search documents for specific information (apart from attributes) by choosing the **Search** option in the document list dialog. The user then specifies the text to search for (optionally using Boolean combinations of words), the media types as well as the documents scope for the search (Figure 3). For images, audio and video, a keyword search is performed (as indicated before, we have not yet developed content-based indexing and access techniques for these media types). The document scope to be searched can be all the documents in the database, only the documents currently displayed in the list, or only the selected document. The search facility allows the user to query the database and locate specific information directly as needed. This provides an easier and more efficient way of accessing data than using the navigation facility where exploration can be very time-consuming.



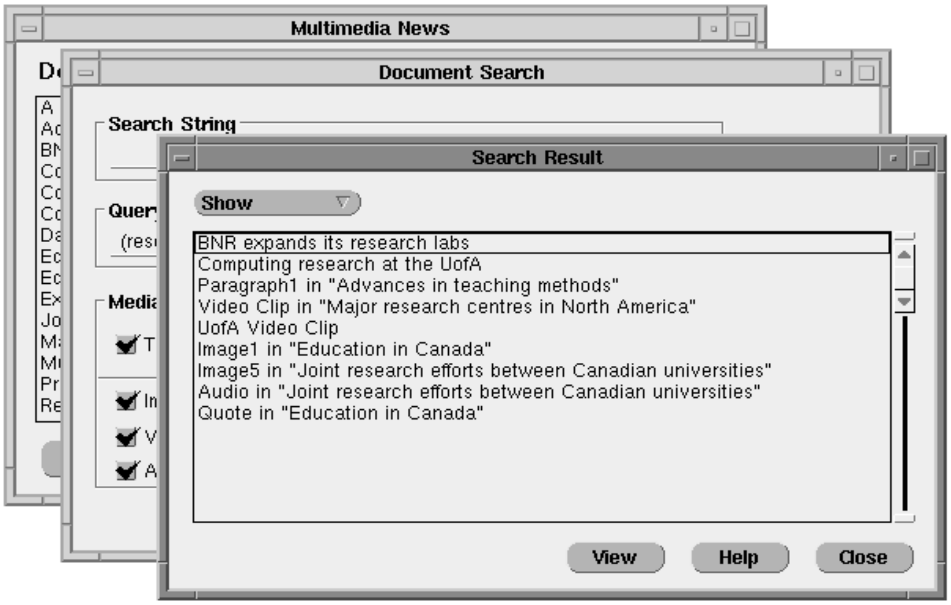
Query: (research) AND teaching

Media: [✓ Text] [✓ Images] [✓ Video] [✓ Audio]

Figure 3: Searching Documents

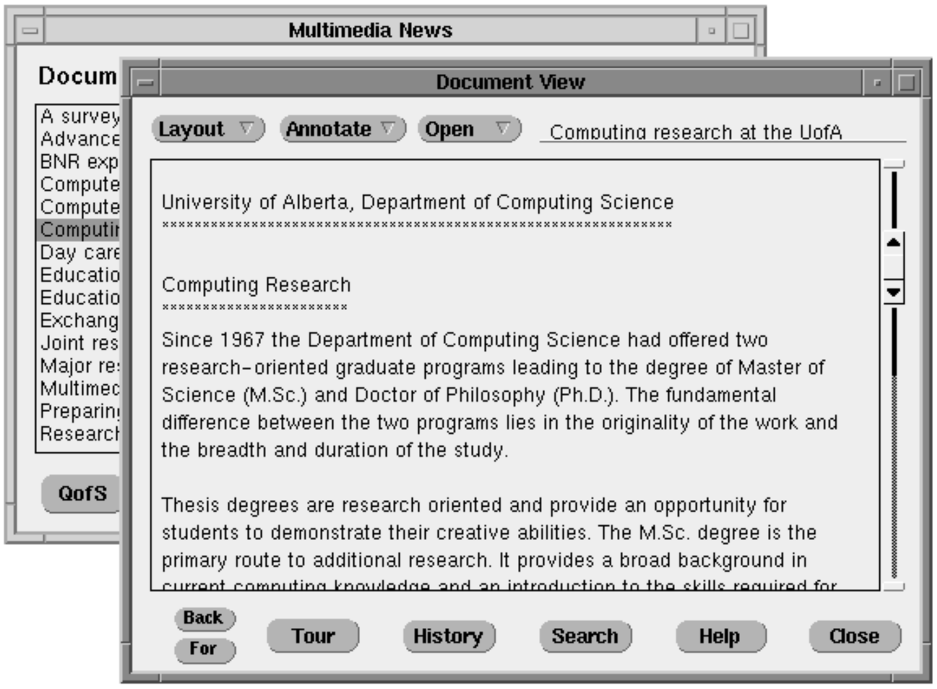
The search facility returns a list of objects which match the query (Figure 4). The objects can be text paragraphs, video clips, images, audio streams or even complete documents. The types of objects returned are specified by the user during the search (it corresponds to the media types checked). For example, if the user selects only video and images as the media types to be searched, the query will only return video and image objects which match the query, but not the associated text or audio. The user can go back and narrow or widen the types of objects returned by re-marking the media types.

Since the search results list consists only of text summaries of objects, there is no need to fetch these objects from the database server. This significantly reduces communication costs since multimedia objects can be very large and expensive to transmit. It is only when the user selects a search result object to view that the object is actually transferred to the client (Figure 5).



BNR expands its research labs
Computing research at the UofA
Paragraph1 in "Advances in teaching methods"
Video Clip in "Major research centres in North America"
UofA Video Clip
Image1 in "Education in Canada"
Image5 in "Joint research efforts between Canadian universities"
Audio in "Joint research efforts between Canadian universities"
Quote in "Education in Canada"

Figure 4: Search Results



University of Alberta, Department of Computing Science
****

Computing Research
****

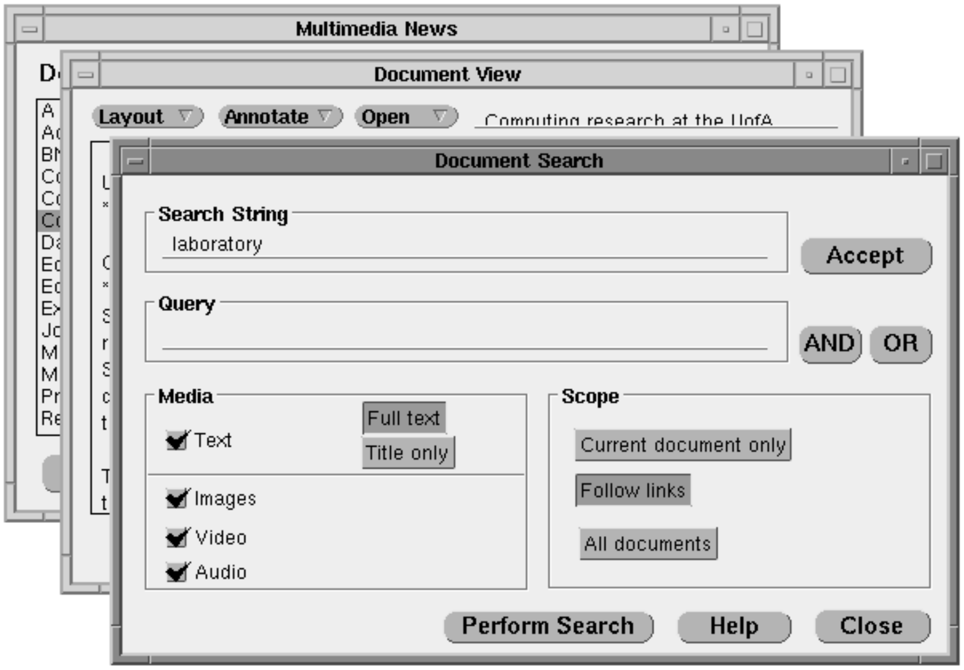
Since 1967 the Department of Computing Science had offered two research-oriented graduate programs leading to the degree of Master of Science (M.Sc.) and Doctor of Philosophy (Ph.D.). The fundamental difference between the two programs lies in the originality of the work and the breadth and duration of the study.

Thesis degrees are research oriented and provide an opportunity for students to demonstrate their creative abilities. The M.Sc. degree is the primary route to additional research. It provides a broad background in current computing knowledge and an introduction to the skills required for

Figure 5: Document view

Searching within a Document

The user can also perform a search from within a document by selecting the **Search** option in a Document View (Figure 5). The Document Search Dialog (Figure 6) is similar to the search dialog that is used to search a document list (Figure 3) except that the search scope options are different. The scope of the search can be the current document only, the documents directly linked to the current document or all the documents in the system. The user can also specify the media types to be searched. The search results are displayed in the same kind of dialog window as that is used to display the results of searching from a document list (Figure 4).



Search String: laboratory

Query:

Media: [✓ Text] [✓ Full text] Title only; [✓ Images] [✓ Video] [✓ Audio]

Scope: Current document only [✓ Follow links] All documents

Figure 6: Searching within a Document

4. Linking with the Database

A unique feature of the visual query facility is its tight coupling with the multimedia database. Each time a search is performed or a document component is to be viewed, a database query is issued to the underlying ObjectStore database [OHMS92]. The visual query facility, running on a client machine, issues ObjectStore queries which fetch the required objects from the server. The ObjectStore client then returns the matching objects to the interface. It is the responsibility of the visual query interface to manipulate these objects depending on the tasks to be performed.

4.1 Presentation Control

We define a multimedia document as a structured collection of objects. A document contains three types of information: the data *content*, the *logical structure* and the *presentation structure*. The data content consists of mono-media objects: text strings, audio streams, video clips and images. The logical structure determines the organization of the different components in relation to one another. The *presentation structure* is concerned with the layout of the document on the user's display. For example, the content of a book is its text string. Its logical structure is its chapters, sections, paragraphs and so on. Its presentation structure includes font types and sizes, page layouts, etc. The content, logical and presentation structures are stored separately in the database. Separating content from logical structure has many advantages related to the efficiency of transferring objects from the database server to the database client [VÖSEM95]. Separating presentation structure from logical structure allows users to customize their presentations based on hardware and personal preferences. As mentioned earlier, customization spans a variety of system parameters such as defining the presentation of anchors within a document, defining different layouts for the same document and attaching a font style (such as italics, bold, reversed video, etc.) to emphasized text. All these presentation specifications are stored as *style sheets* in the database. A style sheet maps logical element types to their specified display settings. To represent style sheets in the database, an SGML DTD (Document Type Definition) for a style sheet must be defined [VÖSEM95]. An example of a simple style sheet, which specifies that a headline element (in a news article) should be written using the Times font, would be:

```
<rule>
      <source>hdline</source>
      <spec><attr>font</attr><value>Times</value></spec>
</rule>
```

A more sophisticated example of a style sheet is:

```
<rule>
      <source>listitem</source>
      <spec>bullets<value>square</value></spec>
      <spec>indent<value> 7 </value></spec>
      <spec>font<value>courier bold</value></spec>
</rule>
```

This specifies that a list item has square bullets marking the item, is indented by 7 spaces, and that the font is courier bold.

As it currently stands, the implementation of style sheets provides sophisticated handling of textual data, but relatively simple handling of continuous media (video and audio). When displaying a document or document component, the visual interface uses the style sheets to determine display settings (font and layout information) associated with that document or component. Note that neither the content of a document nor its logical structure can be updated in the database. However, style sheets provide a user interface customization mechanism since they allow the presentation structure to be updated in the database.

Furthermore, we can define certain presentational specifications, that are user dependent, but document independent. Such information can be stored in *user profiles*. These include window size, control panels for various displays, default document filters, etc. User profiles are also stored in the database and can be updated.

4.2 Translating Visual Queries to Database Queries

Visual queries are translated to ObjectStore queries [Obj94]. We will now give a simple example of the translation process.

Assume that the user wishes to filter the initial document list that appears in the Document List window (Figure 1) to include only documents dealing with education in Canada. The user presses the **Filter** button and the Filter window (Figure 2) is displayed. After the user fills in the **Subject** and **Country** fields, the **Ok** button is pressed. The visual query interface translates this information into an ObjectStore query as follows.

Assume that the class of multimedia documents is called **article** and is defined as follows in C++ (note that we only include a sample of the class attributes here):

```
class article {
public:
   char * title;
   char * country;
   char * date;
   char * source;
   char * subject;
   .....
}
```

The following ObjectStore query will be produced by the visual query interface and will return a collection of all the articles dealing with education in Canada:

```
os_database *news_database;
os_Set<article *> *all_articles;

os_Set<article *>& matching_articles =
    all_articles->query ("article *",
       "this->subject == 'Education' &&
       this->country == 'Canada'", news_database);
```

ObjectStore queries are always over a single top-level collection, i.e., a collection not embedded in another object. A query returns a subset of the queried collection (`os_Collection::query()`), a single object (single element query; `os_Collection::query_pick()`), or a Boolean (existential query; `os_Collection::exists()`). Queries can be nested [OHMS92]. The query constraint (the condition satisfying the query) must use data members or function calls involving a comparison operator [Obj94]. These specifications required by ObjectStore must be taken into account in translating all user actions to equivalent queries; for example, when a document object is fetched for viewing.

5 Conclusions and Future Work

In this paper, we describe a visual query facility that allows users to access a distributed multimedia database. This facility provides two major classes of capabilities: *querying* and *browsing*. Querying capabilities are necessary to allow users to directly and efficiently retrieve information from the database. Browsing capabilities allow users to easily explore and obtain information that is directly related to query results. Our visual interface combines these capabilities by concentrating on two areas of technology: query mechanisms and hypertext/hypermedia systems.

The identifying and novel features of our work are:

- Close integration with a multimedia database.
- Provision of a strong querying facility without losing the flexibility of browsing information in a hypermedia fashion.
- A visual query interface so that users need not type complicated ObjectStore queries.
- Flexibility to define the scope of searches and the type(s) of media obtained.

- User customizable style sheets and user profiles that separate the presentation structure of a document from its content and logical structure.

The testbed environment for the system consists of client machines which are IBM RS6000/360's with 64 Mbytes of memory and equipped with video cards. The server machines are RS6000/360's with 128 Mbytes of memory. The servers and clients are connected via a Newbridge ATM switch. The logical database design is modeled using a commercial Object-Oriented database management system, ObjectStore [LLOW91]. The visual query facility (and graphical user interface) is currently implemented in Smalltalk-80 [GR85]. When complete, it will be re-implemented in xlC, which is IBM's C++ compiler for AIX environments, and integrated with the database on the RS6000's.

In the long run, we are interested in development of query languages, access primitives, and visual query facilities that allow for sophisticated querying of multimedia databases. Our interest extends to content-based querying and retrieval of all media types: text, video, audio, and images.

References

[Ada93] J.A. Adam. Interactive multimedia: Special report. *IEEE Spectrum*, pages 22-39, March 1993.

[BD92] M.M. Blattner and R.B. Dannenberg, editors. *Multimedia Interface Design*. New York, NY: ACM Press, 1992.

[EM95] G. El-Medani. Design and implementation of a hypertext user interface for a multimedia kernel. Master's thesis, University of Alberta, Department of Computing Science, 1995.

[FS91] H.P. Frei and P.Schauble. Designing a hypermedia information system. In *DEXA 91. Database and Expert Systems Applications*, pages 449-454, Berlin, Germany: Springer-Verlag, Wein, 1991.

[GR85] A. Goldberg and D. Robson. *Smalltalk-80: The Language and its Implementation. Addison Wesley*, 1985.

[GT94] K. Grønbæk and R.H. Trigg. Design issues for a dexter-based hypermedia system. *Communications of the ACM*, 37(2):40-49, February 1994.

[Hay93] R. Haykin. *Demystifying Multimedia*. Apple Computer, Inc., 1993.

[HKR+92] B.J. Haan, P. Kahn, V.A. Riley, J.H. Coombs, and N.K. Meyrowitz. IRIS hypermedia services. *Communications of the ACM*, 35(1):36-51, January 1992.

[LG94] L. Li and N. Georganas. MPEG-2 coded- and uncoded-stream synchronization control for real-time multimedia transmission and presentation over B-ISDN. In *ACM Multimedia 94. Proceedings of Second ACM International Conference on Multimedia*, pages 239--246, San Francisco, CA, USA, October 15-20 1994.

[LLOW91] C. Lamb, G. Landis, J. Orenstein, and D. Weinreb. The ObjectStore database system. *Communications of the ACM*, 34(10):50-63, October 1991.

[Luc90] D. Lucarella. A model for hypertext based information retrieval. In *Hypertext: Concepts, Systems and Applications. Proceedings of the European Conference on Hypertext*, pages 81-94, INRIA, France, Nov. 1990.

[Nie90a] J. Nielsen. *Hypertext & Hypermedia*. San Diego, CA: Academic Press, Inc., 1990.

[Nie90b] J. Nielsen. The art of navigating through hypertext. *Communications of the ACM*, 33(3):296-310, March 1990.

[Nie91] J. Nielsen. Usability considerations in introducing hypertext. In H. Brown, editor, *Hypermedia/Hypertext And Object-Oriented Databases*, pages 3--17. Chapman & Hall, 1991.

[Obj94] Object Design, Inc., Burlington, MA, USA. *ObjectStore User Guide for OS/2 and AIX/xlC Systems*, January 1994.

[OHMS92] J. Orenstein, S. Haradhvala, B. Margulies, and D. Sakahara. Query processing in the ObjectStore database system. In *Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data.*, pages 403-412, San Diego, CA, USA, June 2-5 1992.

[Pla91] C. Plaisant. An overview of Hyperties, its user interface and data model. In H. Brown, editor, *Hypermedia/Hypertext And Object-Oriented Databases*, pages 17-31. Chapman & Hall, 1991.

[VBD+93] A. Vogel, G.V. Bochmann, R. Dssouli, J. Gecsei, A. Hafid, and B.Kerheve. On QoS negotiation in distributed multimedia applications. Internal report, Université de Montrèal, Canada, 1993.

[VÖSEM95] C. Vittal, M.T. Özsu, D. Szafron, and G. El-Medani. Object-oriented modeling of multimedia documents for a news-on-demand application. submitted to *ACM Int. Conference on Management of Data*, May 1995.

[Wil91] I. Williams. Hypermedia for multi-user technical documentation. In H. Brown, editor, *Hypermedia/Hypertext And Object-Oriented Databases*, pages 17-31. Chapman & Hall, 1991.