

The Multimedia Object Presentation Manager of MINOS: A Symmetric Approach

S. Christodoulakis, F. Ho, and M. Theodoridou

Department of Computer Science
University of Waterloo,
Waterloo, Ontario N2L 3G1

Abstract

Large multimedia data bases become feasible due to recent advances in hardware technology. A very important component of multimedia data base management systems will be the presentation manager which will be responsible for effective multimedia presentation and browsing on the screen of workstations.

In this paper we present the functions provided for multimedia presentation and browsing in MINOS, a multimedia information system. The presentation and browsing capabilities provided make effective use of the capabilities of a modern workstation to increase the man-machine communication bandwidth. We regard voice as an important means of communication. Symmetric capabilities for text and voice browsing are provided.

1. Introduction

Data base management systems have been very successful in the commercial world for handling formatted data. New opportunities for data base management companies and researchers emerge in application environments which require unformatted data such as text, voice, and images [Christodoulakis 85a]. These application environments become important as result of recent advances in the hardware technology.

Affordable, powerful workstations appeared in the market. They have or will soon have enough processing power to meet the very demanding CPU processing requirements of unformatted data. They offer high resolution display devices appropriate for demanding image processing applications. Optical disks with huge storage capacities become reality. They will be appropriate for storing text, digitized voice and digitized images. Very high bandwidth communication links become available. These advances in technology make it possible to develop very large, computer based, data banks of unformatted information.

Traditional data base management research and systems did not pay much attention on data presentation and browsing. The major effort was concentrated on efficient management of data on secondary storage devices, English keyword specification and manipulation languages, concurrency control, correctness and security of data. Methods for formation and presentation of text mainly data have been developed in the context of editors formatters, but they are mainly oriented towards producing documents on a paper output device.

In the near future very large information banks will exist possibly utilizing optical disk devices. Two very important problems will be how to find the desired information within volumes of loosely structured and largely unrelated information, and how to view it effectively. In this environment it is not easy for the user to specify precisely what he wants to see or not see. The process of identifying the relevant significant information and absorbing this information is slow and difficult. Very powerful presentation and browsing facilities are required in order to *increase the communication bandwidth between*

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1986 ACM 0-89791-191-1/86/0500/0295 \$00.75

user and machine. It is our thesis that the *data presentation manager* will be a very important component which will contribute towards this objective in future multimedia data base systems.

Voice will be a very important way of communication with future computer systems. It allows people that are not familiar with typing or they are not fast typers such as doctors or managers to efficiently enter information in some data bank. It also allows users to access information using telephones. In the future *information will be mostly generated, live, and die, within the computer system.* The paper document will not be the principal way of communicating information. Voice information may acquire legal significance. In such an environment voice and text present just two alternative ways of representing information. The information system should offer *symmetric capabilities* for entering, presenting, and browsing through voice or text.

In this paper we describe the presentation manager of MINOS, a multimedia information system under development. Multimedia information in MINOS is composed of attributes, text, voice and image data. The presentation manager provides functions for effective multimedia information presentation and browsing. The presentation and browsing capabilities which are provided are much more powerful than these provided by paper documents. In addition the presentation manager presents a symmetric functionality for presentation of text and voice information. It is our objective to define a set of presentation and browsing primitives which will be powerful enough to handle a wide class of applications which require multimedia information.

In Section 2 we describe the primitives provided in MINOS for multimedia presentation and browsing. In Section 3 we describe how these facilities can be used in various environments which utilize multimedia information. We use the facilities implemented in MINOS to demonstrate how these primitives can be combined for an effective presentation and browsing in various application environments. In Section 4 we describe the way that multimedia information is created and combined to form multimedia objects. In Section 5 we discuss architectural and implementation issues. Finally in Section 6 we describe summary conclusions and related work.

2. Symmetric Multimedia Objects in MINOS

In this section we describe the multimedia objects in MINOS, their primitive components, and the presentation capabilities associated with them. We emphasize the need for symmetric presentation capabilities for text-driven and voice-driven objects, since text and voice may be used interchangeably to describe the same information. In section 3 we give several examples from potential application environments where these presentation primitives may be useful. We used the presentation facilities of MINOS to create these examples.

The unit of information in MINOS is a *multimedia object*. Multimedia objects may be composed of *attributes*, an *object text part* (collection of text segments) an *object voice part* (collection of voice segments), and an *object image part* (collection of images). A unique *object identifier* is associated with each multimedia object. Multimedia objects may be *related* to other multimedia objects. Information about the related objects is kept within the object itself. Multimedia objects may be in an *editing state* or in an *archived state*. Objects in an editing state are allowed to be modified. Objects in the archived state are not allowed to be modified. The presentation and browsing capabilities described in this paper are applicable to multimedia objects which are in the archived state. Each multimedia object has information stored within it which describes how its various parts are interrelated. This information is used for presentation and browsing. The presentation and browsing functions which are available for each multimedia object depend on the object itself and they are presented in the form of menu options.

Text and voice may be used interchangeably to describe the same information. Moreover, they are both one-dimensional in nature. We would like to provide the interactive user with the same capabilities for browsing through information described with text or with voice. Images on the other hand are two-dimensional in nature. A different set of browsing primitives is required.

Browsing facilities for text and voice in MINOS may utilize the *presentation form* of text or voice, the *logical components* of text or voice, or *pattern matching* primitives.

In speech different emphasis and meaning is expressed by a speaker by employing various methods such as increased loudness, different intonations, length of pause between words, etc. In text the same emphasis and meaning aspects are expressed by some special symbols (!, ?, ..) as well as by some conventions such as underlined words, tilted words, bold tones, etc. Text presentation will have to support such mechanisms. MINOS supports text presentation facilities similar to those that are provided by text formatters.

The presentation form of text is subdivided into *text pages*. A text page is all the text information which is presented at the same time at the screen of the workstation. Often text is intermixed with images in the same page. We call these generic pages *visual pages*. When a user browses within text (in a paper document or in the workstation) he typically has an abstract idea on how far within the text he wants to look next, and based on that he estimates how many pages he should advance.

The page concept is required in voice in order to achieve the same effect. *Audio pages* (or *voice pages*) in a speech are consecutive *partitions of the audio* object part which are of approximately constant time length. The user can advance several voice pages at a time in order to find some relevant information. A difference that we would like to accept is that speech is not interrupted at the end of each voice page. In contrast, visual pages are not turned automatically, but only with the explicit selection of a menu option. The reason is that the time required for reading a text page is different from user to user.

Visual page browsing capabilities allow the user to move to next page, previous page, advance a number of pages forth and back, or find a page with a given page number. Voice browsing capabilities allow the user to interrupt the voice output, resume the voice output from the current position, resume the voice output from the beginning of the current voice page, as well as to browse between pages in a similar fashion with text browsing (e.g. next page, previous page, etc.).

A difference between reading text and hearing voice is that text pages present a cache of information to the user. The user can read twice the same word, sentence, or paragraph if he did not understand its meaning. He can even

look at a previous paragraph in the same page in order to understand better what he is reading. A similar facility in voice would allow the user to interrupt the speech, and then use instructions such as previous word, previous sentence, previous paragraph, etc.

It is however very difficult to provide such a facility automatically. Detecting word boundaries, end of sentences, and end of paragraphs automatically from digitized voice (unlimited vocabulary) is very difficult (at least with the current and near future voice recognition and A.I. technology). Moreover, the user himself may not be absolutely sure about paragraph or sentence boundaries in uninterrupted speech. The *short pause* and *long pause* options are provided in MINOS in order to achieve a similar capability. Pause is a segment of digitized voice which does not contain any sound (in practice the intensity of the registered sound is very small). The user may specify that the audio is replayed starting from a number of short or long pauses back from the current position.

The length of the short pause roughly corresponds to the average length of a pause between word boundaries, while the length of the long pause roughly corresponds to the length of a pause between paragraphs. The exact timing for short, and long pauses depends on the speaker and the section of the speech. It is decided from the current context by sampling. Of course there is no guarantee that these mechanisms will match word boundaries and paragraph boundaries respectively. The user may however recall any long recent pauses from the cash of information in his memory, and the combination of those two facilities provide a form of browsing near the current context which is always available to the user, independently on the degree of manual editing that may or may not have been done on the voice input.

A text segment of a multimedia object in MINOS may be logically subdivided into *title*, *abstract*, *chapters*, and *references*. Each chapter is subdivided into *sections*, sections into *paragraphs*, paragraphs into *sentences* and sentences into *words*. For objects which have been generated interactively in a given environment, these subdivisions can be easily identified by the tags that the user inserts in order to format the text.

A voice segment of a multimedia object in MINOS may also be subdivided into logical components as in text. Browsing capabilities in text or in voice allow the user to see or hear the page with the next or previous start of a logical unit (such as chapter, section, etc.).

The logical components of voice may be manually identified at the time of the insertion by pressing the appropriate buttons (or at some later point in time). When the logical components are identified at insertion time the speed with which the user can insert information in the computer is reduced, and in addition the user's hands become occupied with the buttons. It may not be desirable to manually edit all incoming information.

Some information is clearly more important than other and it may deserve to be edited if this is going to facilitate other users when they browse on it. Some other information may not be as important. The degree of desired editing varies according to the importance of information. For example, in a certain object, only identification of chapters may be desirable. In another, identification of chapters and sections and paragraphs may be desirable. A similar situation arises in text information when this information has been inserted by means of an image capturing capability (as a collection of bit-maps of pages).

The logical browsing options that are available to the user in MINOS depend on the object (e.g. what logical units have been identified for the object). The menu options which are displayed define the set of available operations (e.g. next chapter, etc.).

The third type of browsing on text and voice information is based on pattern matching. A user types a text pattern or speaks a voice pattern which is recognized, and the system returns the next page with the occurrence of this pattern in the object's text or voice. Voice recognition is not taking place at the time of browsing. Instead, some voice segments have been recognized at the time of voice insertion, or at machine's idle time, from the digitized voice. The recognized voice segments are used to provide content addressability and browsing by using the same access methods as in text. Again, the extent to which this capability will be used will depend on the importance of the particular object. Two aspects are of interest: First, voice

recognition (even limited) is used to reduce (or eliminate) the need for manual indexing which would be necessary for both, retrieving objects based on content as well as for browsing within a particular object. Second, recognized utterances are associated with a particular point of the object voice part in order to facilitate browsing within an object.

Each multimedia object has a *driving mode* associated with it. The driving mode is the principal way of presenting the information in the object, and it can be either *visual* or *audio*. *Visual mode objects* have a principal presentation form which is based on visual pages. The "next page" command for visual mode objects always implies the next page of visual information (text or images). The same is true with all the remaining page browsing commands.

Audio mode objects have a principal presentation form which is based on audio pages. Next page in those objects implies the next audio page. The reason for enforcing a driving mode for each multimedia object is so that the users do not become confused trying to navigate in two different media at the same time.

Voice logical messages are unstructured audio segments (typically short). They can be *attached* to either visual mode objects or audio mode objects. When attached to visual mode objects they may be associated with text segments or images. (Text is linear. Two points identify the beginning and the end of a text segment. The two points may coincide.) When attached to audio mode objects they may be associated with voice segments or with particular points within the object voice part. The semantics are that the voice logical message will be played when the user first branches into the corresponding segments during browsing. In the case of audio mode objects the logical voice message is played *before* the voice of the related segment. Voice logical messages may be attached to overlapping text segments or images.

Visual logical messages are short (at most one visual page long) segments of visual information (text and/or images). They are unstructured in the sense that they are always displayed in the *same* page of the presentation form (top part). They can be attached to either audio mode objects or visual mode objects as in the case of audio logical messages. When attached to audio mode objects the semantics are that the

visual logical message will stay on display for the duration of the play of each voice segment to which it is attached. The semantics when they are attached to audio mode objects are that the logical message is displayed at the upper part of the screen while the lower part of the screen is devoted to the display of parts of the related visual segment. When the user turns pages another part of the related segment is displayed in the lower part of the page while the visual logical message keeps being displayed at the top of the page. The user has the option to specify that the visual logical message is displayed only once whenever the user branches during browsing from a non-related segment at any position within a related segment.

Relevant objects are objects which contain information related to the information which exists in a section of a given (parent) object. Relevant objects are independent multimedia objects (e.g. they have existence by themselves) in contrast to voice logical messages and visual logical messages which have only existence as a part of a multimedia object. An object may have several relevant objects (including itself), each one having some information related to a part of the parent object. The user does not automatically see the relevant objects (in contrast to logical messages).

A *relevant object indicator* which is displayed on the screen of the workstation indicates the existence of a relevant object. The user can browse through a relevant object by explicitly selecting the relevant object indicator using the mouse. The driving mode of the relevant object may be different than the driving mode of the parent object. The user can browse through the information of the relevant object by using the driving mode of the relevant object. He can return back to the parent object by explicitly selecting the *return from relevant object* indicator. At this point the mode of browsing of the parent object is reestablished.

Browsing within a relevant object can be done by using the menu options for browsing through a multimedia object as we described before, or by using *relevances*. Relevances are sections of text or voice or parts of images of the relevant object which are related to the content of the particular section of the parent object. Relevances to text sections are indicated graphically with beginning and end indicators.

Relevances to images are indicated by closed polygons displayed at the top of the image. Relevances to voice segments are indicated by the fact that the voice segment is played independently. (A menu option has to be selected in order to hear the next related voice segment).

Transparencies are visual pages which allow the user to see the previous visual page displayed on the screen of the workstation. A *transparency set* is an ordered set of consecutive transparencies. The multimedia object designer may specify one of two different ways for displaying the transparencies of a set. The first method is by displaying every transparency on the top of one another (and on the top of the last page before the transparency set). The second method is by displaying every transparency of the set separately, on the top of the last page before the transparency set. The user may alter the presentation order specified by the document designer and he may choose to see certain transparencies of the set only projected at the same time. He can do that by displaying the transparencies independently (using the second method above) and selecting the ones that he wants to see superimposed.

Transparencies are typically very useful for displaying information on the top of images. They are also useful for displaying information on the top of text information, such as an alternative set of values for some measurements of an experiment for example. We could not however find realistic examples to justify the use of a similar mechanism on the top of voice pages.

Images in MINOS may be *bitmaps* or *graphics*. Images with graphics contain *graphics objects* such as points, polygons, polylines, circles, etc. Graphics objects may have a *label* associated with them. A label is some short information about the object. The presentation form of a label may be *invisible*, *text label*, or *voice label*. Text label is a short piece of text which is associated with a graphics object, and voice label is a short piece of voice which is associated with a graphics object. Text labels are displayed near the graphics object, at a designer's specified position. A *voice label indication* is also displayed near a graphics object with a voice label (at a designer specified position). Invisible labels may be text labels or voice labels which do not display any information about their existence by default.

Voice labels are not played automatically. This is not incompatible with the text labels because the fact that the text is displayed on an image it does not imply that the user reads it (even if he browses through the image). The user reads some textual information only when he focuses on it. This act is simulated for voice labels by selecting the appropriate voice indicator using the mouse. The user may also request that all voice labels are played, in which case the system plays them in a system defined order.

Labels may be used to identify the corresponding objects in an image. The user can specify a pattern and request that the objects in which this pattern appears within their label are highlighted. This facility is useful for browsing through large images with many objects on them, such as a road map. The inverse facility is also provided: the user can select an object using the mouse and the system plays or displays the label associated with the object.

Images are two-dimensional in nature. In very large images the user may want to see a small portion of the image (window) at a time. He can use this window to browse through the image. The system will only retrieve the relevant data. A *view* is a rectangle overlaid on an image. The portion of the image which is enclosed by the rectangle is presented into the display of the workstation so that it utilizes a major part of the area of the screen which is used for display. The view can be moved at the top of the image using menu options and the mouse. If the voice option has been turned on the system plays the voice labels which are encountered as the view moves. Non-contiguous moves (jumps) of the view can also be specified by choosing menu options. The dimensions of the view can be shrunk or expanded by small quantities at a time by selecting the appropriate menu options and defining the size of the new rectangle with respect to the old size. When the size increases new labels may be played.

View definition can be done on the top of a *representation* of the image. A representation of the image is an image itself, where only a high level representation of the content of the image are presented in positions which correspond to the actual positions of the objects of the image (a miniature). The representation of the image is much smaller than the image itself, and thus it is easily transferable to main memory and

projected on the display. When a view is defined on the representation image the system has to transfer only the data of the view in main memory and not the whole image as in the case that a user retrieves all the data of the image and then he zooms to the desired data. The system explicitly indicates that an image is a representation, and the menu options displayed when a representation is shown allow for the definition of a view and the retrieval of the related data from the image.

A *tour* is a sequence of views defined on an image by the multimedia object designer. The sequence is played automatically (the user does not need to press the next page button). A tour is defined by a rectangle and a sequence of points indicating the position of the rectangle on the large image or on a representation of it. A logical message (visual or audio) may be associated with each position of the tour. The user may interrupt the tour and move the window all round in order to navigate through other positions of the image.

An *overwrite* is a visual page with an image which contains a number of bitmaps or graphics objects (possibly shaded). When the overwrite page is turned, the bitmaps, lines, and shades of the overwrite image replace whatever existed in the previous page but they leave anything else intact.

Process simulation is an ordered set of consecutive visual pages which is displayed one after the other automatically (without pressing the next page button). Logical messages may be attached to each page. When audio messages are attached the next visual page is only shown after the logical audio message has been played. The relative speed by which pages are placed one on the top of another is set at object creation time but it may be altered by the user. The automatic page change may be used to give the impression of a contiguous move or change, thus simulating a process.

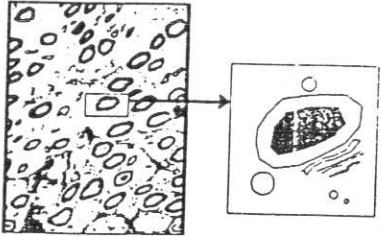
3. Application Environments

In this section we describe how some of the presentation capabilities that we defined in the previous section could be used in various multimedia data base environments. We also present some examples from these environments using the facilities of MINOS.

- 11 -

FATTY TISSUE MORPHOLOGY AND PHYSIOLOGY

Fatty tissue is observed in all fishes, amphibians, reptiles, birds and mammals in all of them the function of the tissue is the same. The storage of nutrient compounds with high chemical energy content such as fatty acids in order to maintain the organism during lack of food is that sense the tissue is a pool from where chemical compounds can be recovered under a certain control and is where other metabolites can be stored in the form of fatty acids when plenty of food is available.



The morphology of the fatty tissue is characteristic and easy to recognize in all the animals. The blood circulation system is usually not well developed in the tissue. The cells are rather big with a large cytoplasm where not many organelles can be found.

RETURN TO BROWSE	CREATE A FILTER	APPEND TO FILTER	PLAY VOICE
NEXT PAGE	PREVIOUS PAGE	SELECT ANNOTATION	RETURN FROM ANNOTATION

- 12 -

DIRECTIONS

By V-1: Take 401 west to Hwy. 10 Travel north through Brampton to Victoria. Turn left at Spadina. Travel west over the railway tracks to the 3rd Concession. Turn north to Cheltenham. The road continues along up to the Inglewood Side Rd. Cross the road and continue north. Follow the road past the Caldon Farming and Sheds Club. The farm is the 1st driveway on the right after the Club.

By L-1: Take the Gray Coach (978-8111) bus to Clonara (on route to Orangeville). Get off on Highway 10 at the Inglewood Side Road. Walk west to the 3rd Concession and north to the farm. It is about 2 miles.

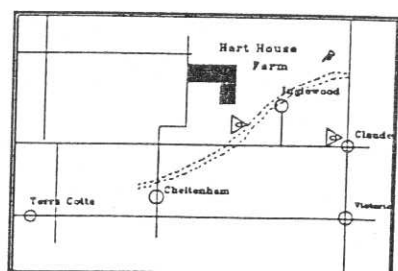


Fig 1: Hart House Farm Road Map

RETURN TO BROWSE	CREATE A FILTER	APPEND TO FILTER	PLAY VOICE
NEXT PAGE	PREVIOUS PAGE	SELECT ANNOTATION	RETURN FROM ANNOTATION

Figures 1 and 2: Visual pages with text, graphics and bitmaps in MINOS.

As we mentioned in the previous section, MINOS provides presentation capabilities for text similar to those found in traditional text formatters. Such capabilities for example allow for various character fonts, letter sizes, paragraphing, indenting, etc. Typical documents that circulate in offices are a special case of the multimedia objects of MINOS.

Complex images with bitmaps, graphics, and text information can be interactively generated and integrated in the presentation from of an object. Figures (1), and (2) show visual pages of multimedia objects with text, graphics and bitmaps on them. In the right hand side of the screen some menu options displayed are shown. These menu options allow the user to browse through the visual pages in the various ways that we described in the previous section.

Paper form documents of an office are only a limited special form of multimedia information. In the future, complex multimedia objects will be created, live and die within computer systems. The presentation form of these objects need not be restricted by the limitations of the paper form. Innovative presentation capabilities in MINOS include audio mode objects, logical messages, transparencies, relationships, tours, process simulation and complex image presentation capabilities.

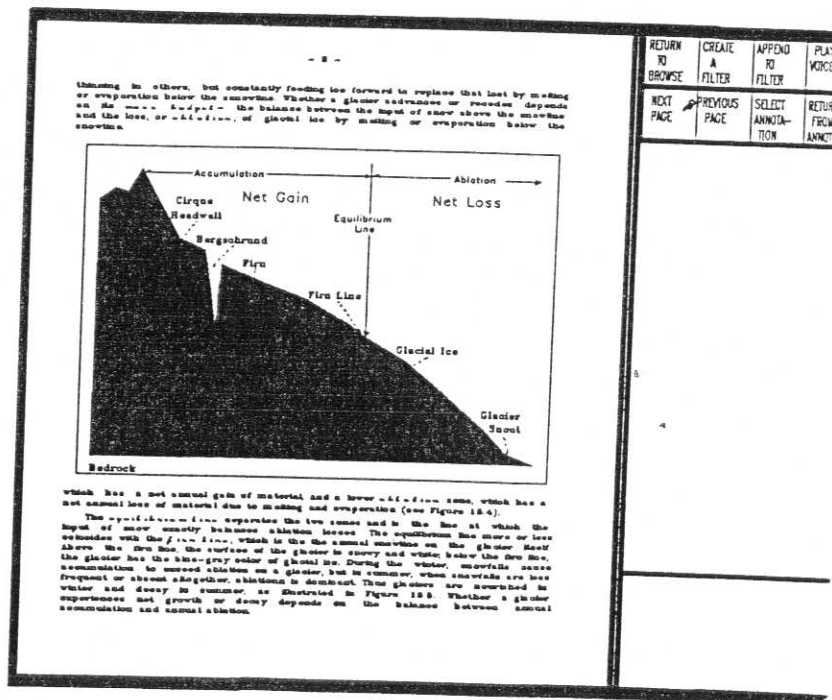
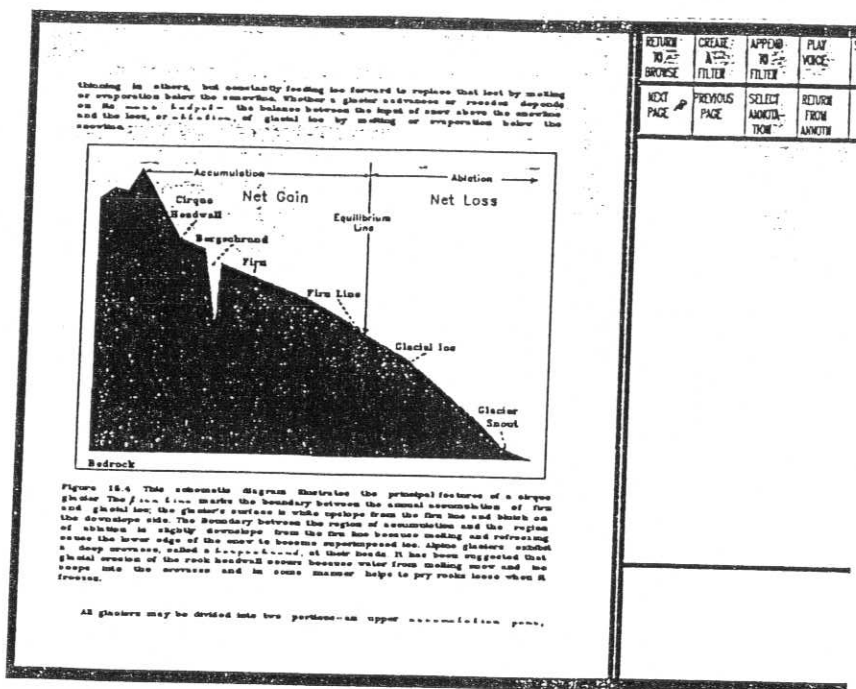
Audio mode objects can be used together with visual logical messages to describe the same information which can be described by objects composed of text and images. Consider for example a doctor who wants to file information about his observations on an x-ray of a patient. He may find it easier to insert this information as an audio mode object. (Doctors are notoriously bad typers!) The x-ray can be attached as a logical visual message to the section of the voice object part which relates to the x-ray. At presentation, the x-ray will only appear on the screen of the workstation during the related section of the speech. In addition, if the user during his browsing branches at some section of the speech which relates to the x-ray, the x-ray will automatically be displayed. In the previous example all the voice information which was related to a particular image was played while the image was displayed in the screen of the workstation. This way the user could see the image while he was hearing the observations about the contents of the images.

A symmetric functionality can be achieved for visual mode objects by attaching visual logical messages to sections of a visual mode object. For example the visual logical message could be the x-ray which is displayed by default on the upper part of the screen. The lower part of the screen can be used for displaying the related text information which contains the observations of the doctor. A user (presumably the doctor himself or other doctors at some other point in time) can browse through the related text by keeping continuously the x-ray in front of him. This way he will not have to turn forth and back the visual pages in order to associate the meaning expressed in the text with the contents of the image (as is usually the case with paper documents). The x-ray bitmap is only stored once within the multimedia object.

An example of visual logical messages on a visual mode object is shown in figures 3 and 4. In this example the text to which the image (visual logical message) relates does not fit in the lower part of a single visual page. When the user selects the next page button a new section of related text replaces the first. Three pages are needed in this particular example to fit all the related text. The final page is only partially filled. Selecting the next page button at this point will result in the display of a new visual page which does not contain the image.

Transparencies and transparency sets are useful in many application environments. In an office filing environment transparencies may be used to create multimedia documents which simulate the act of an active speaker who superimposes transparencies in order to interrelate information (for example compare values of an experiment or curves indicating some results projected on the same axes). Logical voice messages may be associated with each transparency to simulate this act. This is a much more effective way of presentation of information than just reading sequential text. The result may be increased man-machine communication bandwidth. This capability is also desirable for future, computer resident, textbooks.

In a medical information system environment a doctor can use this capability to file an observation made on an x-ray. In an audio mode object the x-ray (typically a large bitmap) will be a logical visual message associated to a segment of voice. Transparencies which are also logical

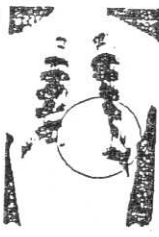


Figures 3 and 4: A visual logical message (image) on a visual mode object. By pressing a mouse button various parts of the text associated with the image are displayed in the same page with the image. The image is only stored once.

RETURN TO BROWSE	CREATE A FILTER	APPEND TO FILTER	PLAY VOICE	SC P
NEXT PAGE	PREVIOUS PAGE	SELECT ANNOTATION	RETURN FROM ANNOTATION	

TORONTO GENERAL HOSPITAL
DEPARTMENT OF PATHOLOGY

PATIENT'S NAME: A. Smith
 INSURANCE #: 123456
 EXAM. DATE: 4/8/88
 DATE OF BIRTH: 29/2/50

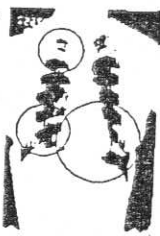


Dr. P. McDonald: Previous operation has been done in Oct 23, 1984. A tumor has been removed. The results from the biopsy for malignancy were negative. After the operation a minor contamination had been observed and cured.

RETURN TO BROWSE	CREATE A FILTER	APPEND TO FILTER	PLAY VOICE	SC P
NEXT PAGE	PREVIOUS PAGE	SELECT ANNOTATION	RETURN FROM ANNOTATION	

TORONTO GENERAL HOSPITAL
DEPARTMENT OF PATHOLOGY

PATIENT'S NAME: A. Smith
 INSURANCE #: 123456
 EXAM. DATE: 4/8/88
 DATE OF BIRTH: 29/2/50



Dr. P. McDonald: Previous operation has been done in Oct 23, 1984. A tumor has been removed. The results from the biopsy for malignancy were negative. After the operation a minor contamination had been observed and cured.

Dr. M. Lee: During the nineteenth year of the patient he has been treated in Wesley Hospital after a car accident for minor injuries in the chest.

Dr. B. Alexander: A kidney malfunction had been observed in March 1978. A bacterial infection had been determined and Erythromycin had been used for 15 days since no penicillin could be provided to the patient due to an allergy to the previous antibiotic.

Figures 5 and 6: Application of the transparency capability of MINOS in a medical information system equipment. Transparencies may be superimposed on the top of a bitmap as the user presses the next page button. Each transparency contains some graphics information (circle) to identify a section on the x-ray, and some text information related to it.

visual messages can be used to pinpoint particular areas within the x-ray that relates to some subsection of the speech.

A symmetric capability can be achieved in visual mode objects by superimposing transparencies which contain a polygon or a circle to identify an area of interest on the x-ray, as well as some related observations in a text form which are displayed under the image. This example is shown in figures 5 and 6.

Relationships between objects (in terms of relevant objects and relevancies) provide a powerful means of browsing within the same object or navigating between multimedia objects which contain some related information. One important use is to allow the user to browse through related information which has been inserted into the computer system using various modes (e.g. primarily visual or primarily audio). The presentation manager requires the explicit selection of a relevant object indicator in order to allow the user to browse through an object of possibly different mode. He can return back to the original object by explicitly selecting a return from relevant object indicator. These actions are enforced in order to keep the user confident on where he is and what he does at each point in time.

Relevancies are useful in many applications in order to display information which is relevant to a particular section of the objects. Consider for example a set of images describing an engineering design in various levels of description. One object in a level of description (image) may correspond to one or more objects in a different level of description. The user may want to identify the corresponding objects. This facility can be easily provided by associating a relevant object indicator with the object. When the indicator is selected the related image is displayed and a set of polygons projected on it identifying all the corresponding objects.

Relevant objects provide also an easy way to identify and correlate interesting information within a larger repository of information. A set of options each of which is associated with a relevant object can be displayed. By selecting one of them the user focuses only to what he is interested from a larger volume of data. Figures 7 and 8 show an example. A subway map for a city is projected on the screen together with some options and relevant object indicators. By

selecting one of these options the user can see for example the sites of a university (figure 7) or the locations of the hospitals of a city (figure 8). In this example the related objects are just transparencies which are superimposed on the subway map.

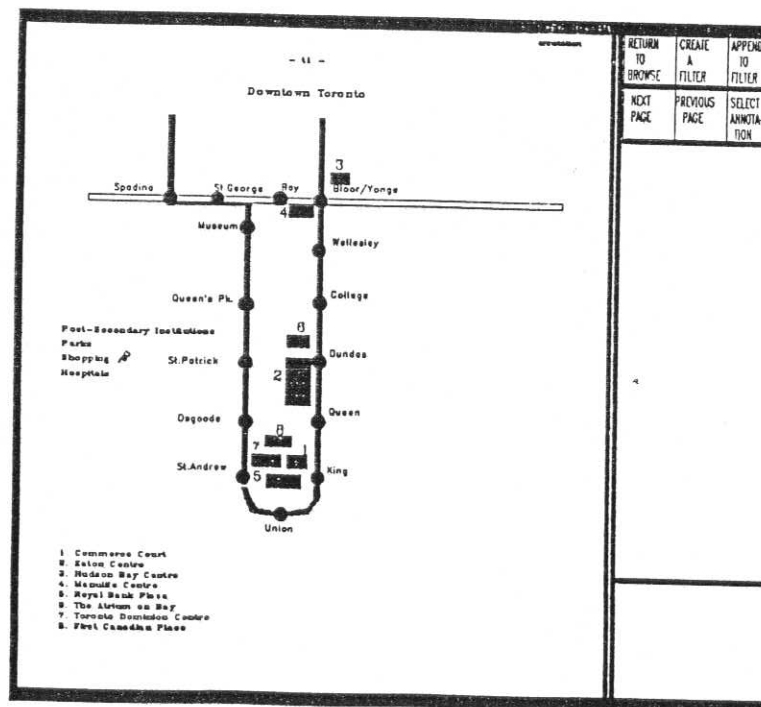
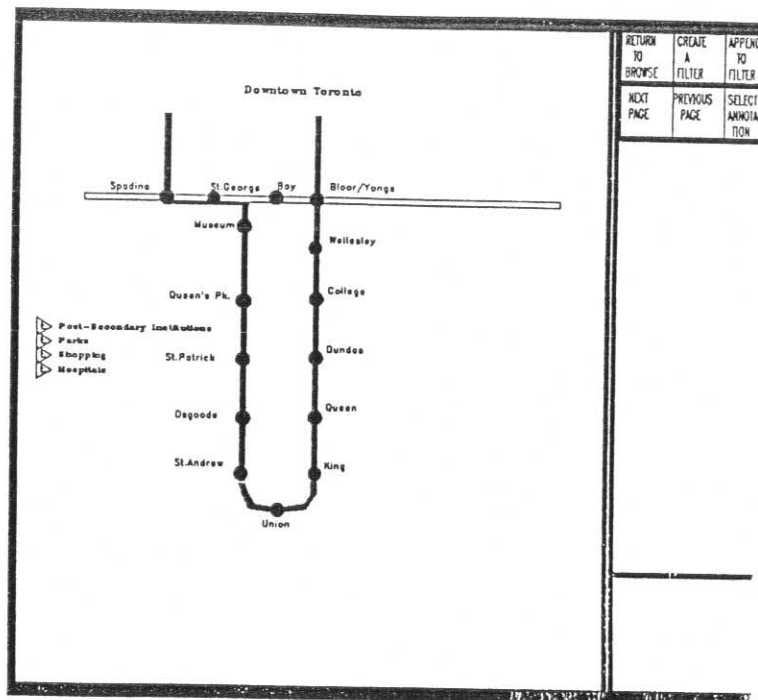
Views facilitate retrieval of pieces of information from very large images without retrieving the whole image. This facility is useful for retrieving information from very large images containing engineering designs (CAD/CAM) or for retrieval of information from large bitmaps. Tours facilitate the automatic movement of a window within a map. If logical voice is associated with each of the views the overall effect is to simulate a guided tour through various sections of the map. This facility is useful in tourist information systems for example.

Associating labels with objects is useful for locating objects within a larger image (as well as for enhancing content addressability and browsing capabilities). Such facilities for example may be found desirable in real estate data bases, tourist information data bases, city maps, and large engineering designs.

Process simulation provides means for non-programmer multimedia object designers to simulate a process. This is done by having the system automatically turning pages at a defined speed. Each new page could be a new image or a transparency or an overwrite. The new page may have a logical voice message associated with it.

Examples of applications are many. It provides an easy way to "program" some forms of animation which could be easily used by non-programmer multimedia object designers. It can be used to explain a complicated installation process in a manual. It could be used to explain a factory process to new employees or visitors. It could be useful to explain the evolution of a battle in tourist information systems or textbooks. Finally in a medical information system it could be used to simulate the propagation of some medicine or disease through the human body.

Figures 9 and 10 show an example where a process simulation is used in order to describe a walk through a part of a city and explain the various sites seen. It is done with a single image and overwrites on the top of it. The overwrites have logical voice messages associated with them.



Figures 7 and 8: Relevant objects which are transparencies are superimposed on a subway map when the relevant object indicator is selected. In this example the relevant object is a map of the hospitals of the city.

4. Multimedia Object Formation in MINOS

There is a number of editors in MINOS. These editors are responsible for the interactive generation and editing of text, image and voice data. We will not describe in detail their operation in here. Their functionality is similar to other editors described in the literature.

The data interrelationships that are useful for multimedia object presentation and browsing are encoded within the *multimedia object descriptor*. The presentation manager uses the descriptor in order to navigate through various parts of an object during browsing.

The multimedia object formatter is responsible for the creation of the multimedia object descriptor. The formatter is declarative and interactive. Declarative formatters emphasize more the logical structure of the object instead of how to do the formatting. Interactive formatters allow the user to see immediately the result of local changes in the formatting commands.

The archived objects are composed of the object descriptor concatenated with the *composition file*. The composition file is the concatenation of several *data files* each one of which contains a certain part of the multimedia object (text parts, images, etc.). The object descriptor indicates how these parts are presented in the physical object. In the case of archived or mailed within the organization objects the object descriptor may also have pointers to other locations within the object archiver so that data duplication is avoided.

Multimedia objects in the *editing state* are composed of a set of files within a *multimedia object file*. The multimedia object file is a set of files organized within a directory which has the name of the multimedia object. This set of files contains a *synthesis-file*, the *object descriptor*, a *composition-file*, a *data-directory* file, and a set of *data files* which contain various pieces of the multimedia object which is to be created (text, images, voice, etc.).

The data directory file contains information about the various data files as well as about data in the archiver that have been extracted but not copied. Such information is the name, type, location, length, and status of data. The status information describes if the data in a particular file is in its final form which is to be used

for archiving or mailing. For images with graphics for example the archival form may be different than the editing form. When the editing of an image is completed its archival form (which is device and software package independent) is produced. The presentation interface of the archiver expects always the data in its final form.

The object formation process starts when the user creates the synthesis file. The synthesis file contains information about the presentation form of the multimedia object, tags with the names of various data files, and possibly text (this will typically be the case for visual mode objects). In parallel the composition file is also created by concatenating the information in the synthesis file with the data of those data files which have been referred to by a tag in the synthesis file. The object descriptor is updated automatically to indicate the location in the physical object where the data of the composition file is displayed. In the case that a data tag in the synthesis file refers to data which exist in the archiver, the object descriptor is updated with a pointer to the location within the archiver where the data is located. This information is found from the data directory. Thus the object descriptor points either to offsets within the composition file or to offsets within the archiver.

When the user inserts information in the synthesis file for visual mode objects a miniature of the current page of the formatted object is displayed in the right hand side of the screen, below the menu options. This way the user can immediately see the results of his formatting actions and he can make the appropriate modifications. If the user makes certain changes in the synthesis or the data files part of the descriptor file and the composition file may have to be deleted and recreated. The user can navigate through the pages of the miniature.

The user can use the same browsing within object capabilities as in the object archiver in order to view objects which are in the editing stage. In this case the object browsing software uses the object descriptor and the composition file in order to derive the presentation form of the edited object. Duplication of software is not required. The user can interrupt the browsing process and go back to the editing formatting process.

When the user is satisfied with the presentation form of the multimedia object which he is editing, he may want to mail or archive it. Archived or mailed within the organization multimedia objects are composed of the concatenation of the descriptor file with the composition file. In the case that objects are archived the offsets of the descriptor have to be incremented by the offset where the composition file is placed within the archiver. Finally when the multimedia object is mailed outside the organization the object descriptor is searched for pointers to information which exists in the archiver. If such pointers exist, the relevant data is extracted from the archiver and appended to the composition file. The pointers of the descriptor which pointed to the archiver are changed to point within the composition file. Finally the object descriptor is concatenated with the composition file and mailed.

5. Architectural Issues

We envision the overall system architecture for MINOS as being composed of a multimedia object server subsystem and a number of workstations interconnected through high capacity links.

The workstations may have some disk devices associated with them. Some of the disks may be shared among workstations. Multimedia objects in an editing state are stored in those disks. Retrieval is done by name. The user edits only a number of these objects at any point in time and he can easily recall their names.

The multimedia object server subsystem is optical disk based and it may also contain one or more high performance magnetic disks. It is used to store objects in an archived state. The major concern in the server subsystem is performance. Performance may be crucial due to queueing delays that may be experienced when several users try to access data from the same device. The subsystem provides access methods, scheduling, caching, version control.

Users submit queries based on object content from their workstation. The queries are evaluated by the server subsystem against the multimedia data base. Users in this environment may not be able to express precisely what they want. Miniatures of qualifying objects may be returned to the user using a sequential browsing interface in order to facilitate browsing through

a large number of objects that may qualify. Miniatures are representations of the information in an object which may facilitate the user to pinpoint relevant objects. (They can for example contain a small bitmap of the first visual page or an indication that an object is an audio mode object and some voice segments which are played as the miniature passes through the screen, etc.).

When the user selects the miniature of an object the multimedia object presentation manager undertakes the responsibility to present the information of the selected object. The multimedia object presentation manager will also facilitate the user in navigating from the current object to other related objects. The user may interrupt this process and return back to the sequential browsing interface or to the query specification interface to refine his filter.

The multimedia object presentation manager resides in the user's workstation and requests the appropriate pieces of information from the multimedia object server subsystems. Several issues related to performance arise in this architecture and they merit further investigation ([Christodoulakis 85], [Christodoulakis and Faloutsos 84]).

Our implementation is done on a SUN-3 workstation supplemented with voice input, output, digitization, and recognition devices. The workstation is connected to several other machines through Ethernet. Image capturing and preprocessing is done using high resolution devices and transferred to the workstation through Ethernet. At the point of writing a large portion of the primitives defined in our design which was described in section 2 of the paper has been implemented. These primitives are mainly for video driven objects. The primitives defined for audio driven objects are under experimentation and implementation.

6. Summary

In this paper we have described presentation and browsing primitives used by the multimedia object presentation manager of MINOS. The presentation manager provides means for effective multimedia object presentation on the screen of a workstation. It aims to multimedia object presentation which are mostly created live and die within a computer system. The manager treats symmetrically objects which are mainly composed of text and objects which are mainly

composed of voice. We have used examples from various multimedia application environments to demonstrate the use of the primitives provided.

We have also described how multimedia objects are formatted using high level descriptive tags which interrelates information. The formation is interactive and the user can look on the screen of the workstation at the same time the formatting part and the results of the formation process.

Previous relevant work on data base environments tried to extend the functionality of DBMS's to handle unstructured data ([Haskin and Lorie 82], [Stonebraker et al. 83]). Some recent approaches provide means for report generation or graph generation on the top of a data base management system [Row 85], or specialized applications implemented on the top of a data base management system [Herot 80].

Work in the document formation area has produced several high quality document formatters, most of them text formatters (e.g. [Chamberlin et al 81], [Reid 80], [Knuth 79], [Thauker et al 79], [Futura 78]). These formatters are mainly oriented towards producing a high quality paper output. In contrast, the multimedia object formatter of MINOS aims towards exploiting the presentation advantages that a workstation provides. In the same spirit with traditional formatters it tries to relieve the user from the painful specification of the details of presentation.

References

- [Chamberlin et al. 81]
 - D. Chamberlin, J. King, D. Slutz, S. Todd, B. Wade: "Janus: An Interactive System for Document Composition", ACM, 1981.
- [Christodoulakis 85]
 - S. Christodoulakis: "Issues in the Architecture of a Document Archiver using Optical Disk Technology", Proceedings ACM SIGMOD 1985.
- [Christodoulakis 85a]
 - S. Christodoulakis: "Multimedia Data Base Management: Applications and Problems. A Position Paper", Proc. ACM SIGMOD 1985.
- [Christodoulakis and Faloutsos 84]
 - S. Christodoulakis and C. Faloutsos: "Design Considerations for a Message File Server", IEEE Transactions on Software Engineering, March 1984.
- [Futura et al. 82]
 - R. Futura, J. Scofield, A. Shaw: "Document Formatting Systems: Survey, Concepts and Issues", ACM Computing Surveys 14,3, 1982.
- [Haskin and Lorie 82]
 - R. Haskin and R. Lorie: "On Extending the Functions of a Relational Database System", Proc. ACM SIGMOD 1982.
- [Herot 80]
 - C. Herot: "SDMS: A Spatial Data Base System", TODS, Dec. 1980.
- [Reid 80]
 - B. Reid: "A High-Level Approach to Computer Document Formatting", Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, NV, 1980, pp.24-31.
- [Row 85]
 - L. Row: "Fill-In-The-Form Programming", Proc. VLDB 1985.
- [Knuth 79]
 - D. Knuth: "TEX: A System for Technical Text", American Mathematical Society, Providence, RI, 1979.
- [Stonebraker et al. 83]
 - M. Stonebraker, A. Stettner, N. Lynn, J. Kalash and A. Guttman: "Document Processing" in a Relational Database System", ACM TOOLS 1,2, April, 1983.
- [Thacker et al. 79]
 - C. Thacker, E. McGreight, B. Lampson, R. Sproull, D. Boggs: "Alto: A Personal Computer", Technical Report CSL-79-11, Xerox Palo Alto Research Center, 1979.