NPS52-88-025

NAVAL POSTGRADUATE SCHOOL

Monterey, California

A CONCEPTUAL DESIGN OF A MULTIMEDIA DBMS
FOR ADVANCED APPLICATIONS

Vincent Lum

Klaus Meyer-Wegener

August 1988

Approved for public release; distribution is unlimited

Prepared for:

Naval Postgraduate School
Monterey, CA 93943

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY 2b DECLASSIFICATION/DOWNGRADING SCHEDULE	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS52-88-025	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 52	7a NAME OF MONITORING ORGANIZATION Naval Ocean Systems Center
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code) San Diego, CA 92152
8a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER O&MN, Direct Funding
8c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
A CONCEPTUAL DESIGN OF A MULTIMEDIA DBMS FOR ADVANCED APPLICATIONS (U)

12 PERSONAL AUTHOR(S)
LUM, Vincent Y., MEYER-WEGENER, Klaus

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Summary	FROM 87/07 TO 88/06	1988, August	2?

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	multimedia databases, image management, SQL extension
			Theses .

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Managing Multimedia Data is becoming more and more important. There are already various operational systems for this task, but they are usually built as special-purpose systems and lack the general capability as exhibited in a database management system (DBMS), suitable for a wide variety of applications. This paper argues that DBMS should be extended to manage multimedia data as well as the standard structured data, exploiting all the established techniques and providing the new data types to all applications. The paper examines the characteristics of multimedia data and outlines some of the current research projects in this area. It recognizes the significant and successful applications of the database technology and information retrieval techniques developed in the last two decades and proposes to capitalize on these advances to develop a DBMS for handling multimedia data. The paper also sketches some directions where future research may be headed to solve the complex issues in multimedia data processing.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Vincent Y. Lum	22b TELEPHONE (Include Area Code) (408)646-3091 22c OFFICE SYMBOL 52Lu

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

* U.S. Government Printing Office: 1986—606-243

UNCLASSIFIED

A Conceptual Design of a Multimedia DBMS for Advanced Applications

June 1988

Vincent Lum, Klaus Meyer-Wegener

Naval Postgraduate School
Department of Computer Science
Code 52
Monterey, CA 93943

Contents:

The work is sponsored by the Naval Oceans System Center (NOSC) as project no. RC32510.

Abstract

Managing Multimedia Data is becoming more and more important. There are already various operational systems for this task, but they are usually built as special-purpose systems and lack the general capability as exhibited in a database management system (DBMS), suitable for a wide variety of applications. This paper argues that DBMS should be extended to manage multimedia data as well as the standard structured data, exploiting all the established techniques and providing the new data types to all the applications. The paper examines the characteristics of multimedia data and outlines some of the current research projects in this area. It recognizes the significant and successful applications of the database technology and information retrieval techniques developed in the last two decades and proposes to capitalize on these advances to develop a DBMS for handling multimedia data. The paper also sketches some directions where future research may be headed to solve the complex issues in multimedia data processing.

1. Introduction

Growth of computing power and the decrease in storage cost make it practical for applications to process text, graphics, voice, sound, and signal data as well as the traditional numerical and alphanumerical data. Storing this new kind of data, generally referred to as *multimedia data*, is one thing, organizing a large amount of them for efficient search and retrieval is quite another (Lum, Wu and Hsiao 1987). The development of *database management systems* (DBMS) has provided a rich selection of methods to organize and process the traditional, formatted data. The question now is how these methods can be extended to handle multimedia data as well. The purpose of this paper is to examine the issues in handling multimedia data and to suggest a solution. Before we go into more detail about the various aspects, we shall present our motivation in searching for a system to handle multimedia data.

Let us take a brief look at the various applications where multimedia data play a strong role.

Information retrieval in the broadest sense: Books contain not only plain text, but photos, graphics, and other types of images as well. For easier searching and management, we can provide multidimensional links between related chapters, paragraphs, and keywords within a book, but also between different books (Nelson 1980).

Publishing: A current application using computers is publishing. Again text segments, pictures, graphics, sketches, and notes or dictation on audio tape are involved. Sophisticated data organization and handling is a necessity (Yankelovich, Meyrowitz and van Dam 1985).

Advertising: Advertising may not be generally considered as handling multimedia data, but in fact it is. An architect can give his clients a better impression of the house he is going to build if he shows them a three-dimensional model from arbitrary viewpoints. He can even guide them on a "video tour" through the house (Phillips 1988).

Artificial reality: The most ambitious approach is mentioned in (Woelk and Luther 1985): The system simulates some kind of virtual office with mailbox, telephone, file drawer etc. that can be used in the same way as the real devices. Similar approaches can be imagined for process control and robotics in factories and power plants.

These are but some of the multimedia data applications we encounter in our everyday life. It should be clear now that there are many others.

Hardware to record and store images, voice, sound, and signals is available and well established. This kind of data can be stored in various digitized formats, ready for processing. However, systems to handle these data are usually based on *highly specialized solutions* for data storage and organization. Often there are several such systems, one for image processing, another for voice recording, yet another for signal processing. Even if the same object is represented in each of the systems, there is no system-maintained link between the data describing it. For instance, there can be a picture database of ships, a standard database of structured data (length, manufacturer, year built, capacity, ...) and an audio database that holds the sound pattern of the ship's engine. This necessarily implies a significant amount of *redundancy*, because they are totally separate systems leading to a data redundancy, i.e. some of the structured data required to identify the object or to process the images and sounds are repeated in each of the systems (e.g. name of the ship, length, etc.). In addition to that the same information, e.g. an image, may be stored several times in different formats needed to display it on different output devices. Such kind of approach makes maintenance difficult and securing data consistency practically impossible.

In contrast to this, it is a general philosophy of *database management systems* to manage all the data shared by a set of applications and to provide each single application with the specific view of the data it needs. This avoids redundancy and makes it much easier to maintain consistency among all the data concerning one object. In addition. new applications can be built that make use of the cross-referencing provided by a system that holds all the information. DBMS further provide mechanisms to handle multiuser operation, preserve consistency, and recover after various kinds of failures.

The configuration we have in mind puts a database system in the center, surrounded by a number of applications and users (fig. 1). This may just be a software solution with all the applications and the DBMS running on the same (mainframe) processor. It may as well be a distributed hardware solution where all the applications are running on dedicated systems (e.g.

workstations) with the DBMS processor acting as the central server.

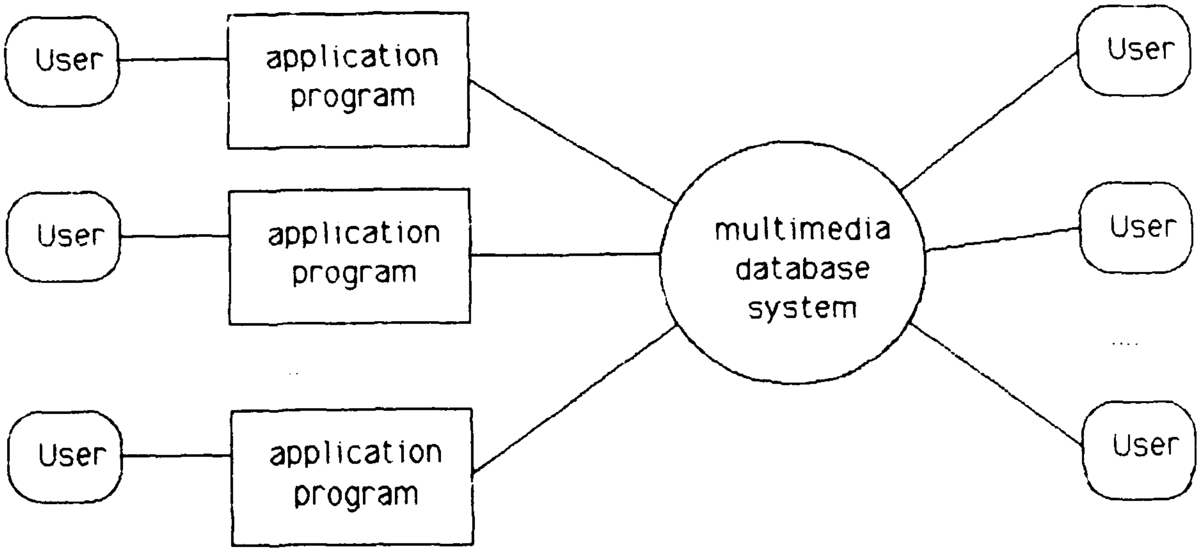

Figure 1: Configuration of a (multimedia) database system with its users and applications

It makes a difference whether a user works with the DBMS directly or through the mediation of an application, i.e. a set of programs. The task of the DBMS is storage and retrieval, but not processing (Masunaga 1987). So when a DBMS is used directly it merely displays the data stored. It may do so in many different ways, using tables, screen formats, graphics and others, but it hardly does any type of algorithmic evaluation. Finding a basic and wisely restricted set of DBMS functions that support a variety of application programs as well as users seems to be the most important design issue for the multimedia database system.

For the rest of the paper, we shall first establish what actually makes the multimedia data different. This means to look at the characteristics of each type of multimedia data in detail. Second, a brief overview on existing work will be given. Although current prototype systems in general use highly specialized solutions for their data management, nevertheless they show how multimedia data are processed and presented to the users, and thus define the database requirements. Next the paper goes on to propose a solution and a brief description of such a system. It then goes on to show why a complete solution to handle multimedia data well requires advanced techniques from other computer science disciplines. Finally a summary and conclusion will be presented.

2. Characteristics of Multimedia Data

Multimedia data have been introduced as text, graphics, images, voice, sound, and signal. They all have in common that a single "value" or object of that type tends to be rather *long*, i.e. in the range of 100 K to 10 M bytes. They are often referred to as being *unformatted*, which means that they consist of a *large and varying number of small items*, e.g. characters, pixels, lines, or frequency indicators. They all carry a more complex structure which varies strongly from value to value and is often not known when the object is stored. Detecting it requires some level of understanding and recognition.

There is sometimes the opinion that multimedia data are just different representations of the same information. It is indeed possible to describe a drawing or a picture with words, or to transfer a voice recording into written text. However, this is accompanied by a loss of information: A picture imagined when listening to a description is always different from the real picture, and the written text can only roughly indicate the sound of the voice that may also carry some information - especially if we know the speaker. To have a better understanding, let us take a closer look at each of the multimedia data types and its specific properties.

2.1. Text

There is a long tradition of storing and retrieving text in computer systems, covered by the scientific discipline of *information retrieval* (Salton and McGill 1983; Lancaster and Fayen 1973; Sharp 1964), which is also called information science. The abstract or full text of a document is augmented by keywords, also known as *descriptors*. This can be done manually or semi-automatically, using a given set of descriptors (the *thesaurus*). The so-called *automatic indexing* that assigns keywords to a text uses special forms of text understanding that originate from artificial intelligence. Many problems remain to be solved before the systems can better a human reader.

Text can be stored just as a variable-length sequence of characters, leaving all the interpretation to the application. There are few operations for a data structure like that: get substring, search for pattern. A DBMS can do much more if it knows about the internal structure. It looks for the formatting commands or special characters (full stop followed by blank: end of sentence, <return>: end of paragraph). It should also distinguish types of text such as book, article, memo, report, thesis, note, etc. There are different levels of complexity in the structures for different applications. There are different degrees of difficulty in handling texts, with the most difficult one being *language understanding* which requires among other complex tasks the construction of a parse tree to exhibit the syntactical structure (Winograd 1983).

Dealing with data structures and language understanding in text has a history longer than the development of techniques in database systems. Due to the complexity and the richness in semantics in languages, no complete solution has been found. However, much progress has been made and many possible solutions though complex do exist.

2.2. Graphics

The term "graphics" (or drawing) is used for pictures that are defined as a *collection of geometrical objects*, i.e. lines, circles, curves, and areas, whereas "image" denotes camera or video pictures (bitmap, raster). The basic items of graphics are rather complex compared to the characters in text and the pixels in an image. Just imagine what is needed to define an area. A drawing can be transformed into an image. This can already be done by hardware that accepts commands like "draw line" or "color area". Storing the geometrical elements instead of the pixels usually saves a lot of space.

A line drawing can be the result of an image analysis. It can as well be derived from a three-dimensional object model using projection and hidden-line techniques, for instance. The latter is very common in computer-aided design (CAD) and provides much more structural information to group the geometrical information (all the lines and areas belonging to one object). The three-dimensional models are more abstract and neutral in that they allow the derivation of several different graphics, and different perspective graphics and images could be generated on demand. The derived graphics or image contains less information than the high-level description. This is actually just the opposite to image analysis where only the original image holds all the information, and the extracted line drawing necessarily neglects some detail.

2.3. Images

As we mentioned earlier, images originate from camera or video recordings and can be stored in the video signal format or in the bitmap (raster) format (Woelk and Luther 1985). The *video signal format* is used on a video tape, which is relatively slow in access, or on an optical video disk. The latter can hold up to 54,000 image frames with an access time of 1-2 seconds for a single frame. Images in the *bitmap or raster format* can be compressed at least an order of magnitude. But even then an 8.5" by 11" page will require more than a million bits, and a color display up to 48" by 80" needs between 2 million and 40 million bytes (Woelk and Luther 1985).

In the raster format the image is represented by a matrix of *pixels* (picture elements). Each pixel may occupy just one bit to indicate black or white, but it might need several bits to code color and greyness. The RGB encoding uses real numbers between 0 and 1 to quantify the intensity of the three colors red, green, and blue. Alternatively, the IHS system and the YIQ system can be used (Ballard and Brown 1982). Formulae are available to calculate one encoding from the other. Different image processing devices (i.e. monitors) need different encodings for the pixels, so the DBMS should use a kind of generic representation and generate the different encodings on demand.

Graphics can be used to generate images with generally a loss of information. The capabilities of recognizing content in a picture are at best primitive.

2.4. Voice/Speech

Voice recording seems to be a much more convenient way of data input, especially for people like doctors or managers who are known to be unwilling to type. Thus, the MINOS system regards voice as equivalent to text and tries to handle it in the same way ("symmetric approach", (Christodoulakis, Ho and Theodoridou 1986)). Of course, browsing through a set of voice recordings is different from browsing through a pile of paper. The current approach to handle the audio part in the systems is only to record the speech, not to process it. The best thing one can do at this time is to simulate a tape recorder with buttons for play, wind, rewind and a position indicator (track, minutes played). As with some dictation machines, the user can be given the chance to divide a tape into sections by pushing a special button and generating acoustic marks. Such kind of approach to handle audio data has very restricted use and is not in tune with the way we handle the formatted data in a normal DBMS.

There are different types of speech coding (Woelk and Luther 1985): source modeling (VOCODER), parametric methods, and waveform coding methods. Data encoding rates from 2400 bits per second of speech for Linear Predictive Coding to 64,000 bits per second of speech for pulse code modulation. This sums up to 18,000 - 480,000 bytes for a minute of voice note.

To structure a speech is difficult. Even to detect the words and sentences in continuous speech requires a high degree of understanding. Voice recognition is still a very time-consuming process. Only if it is speaker-dependent and restricted to single command words, it is simple enough to have practical applications (Rosch 1987).

2.5. Sound and Signal

Voice is a special case of sound, with the important difference that voice and speech can usually be transcribed into text. Sound can be music, noise of an engine, birds singing, and much more. Signal is an even broader category, as sketched in fig. 2. Almost any kind of sensor data or measurement can be regarded as a signal, e.g. radar, radio signals, sonar, EKG, and laser.

Sonar is based on acoustics and could be regarded as sound, but it cannot be heard by humans and is therefore treated as a signal. They all have different recording and encoding techniques. It is expected that they have some general properties apart from being long and unstructured. For example, a certain kind of sound may serve as a warning signal and is so designated.

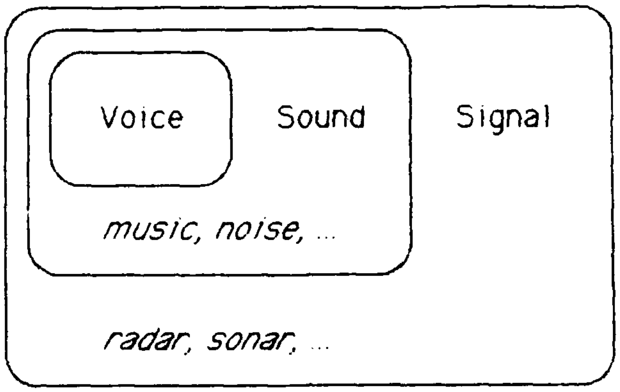

Figure 2: The relation of voice, sound, and signal

3. State of the Art and Other Projects

A variety of hardware is available to record image, voice, sound, and signal data in digitized form. The "write-once-read-many-times" (WORM) disk and the videodisk provide sufficient storage at reasonable cost. But to organize the huge amount of data and to retrieve parts of it according to the specific needs of a user or an application is not yet properly solved.

3.1. Information Retrieval Systems

As mentioned before, information retrieval systems have been around for a long time (Lancaster and Fayen 1973). An example is IBM's STAIRS (IBM 1971). They run on mainframe computers and use their own customized file organizations. Hence, they are completely separated from the DBMS that manages the structured data. Further, as systems like these generally create inverted lists of the entire document, whose size can be voluminous, they do not manage data on-line. Only recently some projects and prototypes have been started that try to integrate the functions of an information retrieval system into a DBMS, e.g. the AIM-P (Lum et al. 1985; Schek and Pistor 1982; Macleod 1981).

Because the needs for information retrieval and database technology are different, researches between these two go in different directions. While database technology is well developed for handling the formatted data that generally exist in the commercial environment, information retrieval technology concentrated in attempts to find pertinent documents based on the content of the unformatted, textual data. To do so, researchers have tried to develop methods that would handle not only the syntactic form of a query, like *operating* adjacent *system* to mean that we wish to find documents with "operating system" appearing in such a format, but also to look into the meaning as written in the document. For example, synonyms are defined and documents that are only likely to meet the specified queries are analyzed for possible retrieval. Those coming "close" as indicated by the various measures are returned along with those that match the query specification completely. This is quite different than DBMS processing where all returned result must satisfy the queries completely. In processing unformatted data, the DBMS technique is deemed not suitable and the approach in information retrieval simulates much better how people actually behave.

3.2. Picture Databases

Picture databases or pictorial databases are being developed since the 70s. In (Chang and Kunii 1981) they are defined as a "collection of sharable pictorial data in various formats". The article gives an overview of several projects in this area. In contrast to multimedia systems the image is regarded as an object of its own right, not as a description of another object. Pictures can be described by some attributes, and they can be linked with each other. They are in the center of data organization and retrieval, not the objects they show. Typical examples are collections of x-ray photographies or tomographic scans in a hospital.

Searching in picture databases is generally done over additional structured attributes. In general, contents of the pictures are not specified nor analyzed for searches. Certain information associated with the pictures such as picture source and color coding must be included so that they can be displayed on a monitor. Some systems allow the user to manipulate them, using some interactive graphics software to draw circles or rectangles around important things for instance. As stated in the article by Raskin and Stone (1987), the lack of standardization in the forms of storing pictures makes it impossible to combine parts of one system with parts of another.

In the processing of the picture databases, invariably the pictures are stored as files specialized for the particular system. No DBMS is used for managing the data.

Technology for handling other kind of multimedia data like voices and signals is hardly developed and shall not be discussed further.

3.3. Multimedia Projects

MINOS has been developed at the Universities of Toronto and Waterloo (Christodoulakis et al. 1986; Christodoulakis, Ho and Theodoridou 1986; Christodoulakis and Faloutsos 1986). It manages highly structured "multimedia objects" that consist of attributes, a text part, an image part, and a voice part. Objects are either in an editing state where they can be modified, or in an archived state where they are available for presentation and browsing. Sophisticated browsing features follow the object's structure, stepping through visual pages and audio pages as well as sections and paragraphs. Logical messages (visual or audio) can be attached to text, image, or audio segments, so that they are shown or played along with them. Multimedia objects can be linked to each other.

MINOS is designed for office automation; its emphasis lies on the user interface. The system is implemented on a file basis. The schema is fixed, it provides a fixed set of elements with associated operations. As updates are only possible after the complete "checkout" of the whole multimedia object into the editing mode, synchronization of updates is fairly simple, but adding a small annotation to an object requires significant overhead.

It is not known what has happened to the prototype of MINOS since 1986. There are no more publications on this subject.

The MCC in Austin, Texas, is running several projects to support multimedia applications. One Program includes a multi-mode integration project that sets out for the development of *MUSE*, the "MUlti SEnsory Information Management System", described as a multimedia logical storage management system, which means it is to cope with the capture, storage, retrieval, presentation, manipulation, and editing of multimedia data as well as the traditional data (Woelk and Luther 1985; Woelk, Luther and Kim 1987). Multimedia applications including end-user interfaces are built on top of MUSE.

MUSE itself will employ a database system to store the multimedia data. MCC programs studying the database requirements of multimedia applications (Woelk and Luther 1985; Woelk, Kim and Luther 1986) determined that an object-oriented DBMS is needed and proceeded to develop *ORION* (Woelk and Kim 1987; Banerjee et al. 1987; Banerjee, Kim and Kim 1988). It will support aggregation hierarchies as well as generalization hierarchies, shared components, historical and alternative versions, long transactions, query and browsing modes, non-persistent presentation of multimedia objects (direct display or replay on output device without intermediate storage in main memory), pattern matching, and media translation. Apart from aggregation and generalization the data model allows for flexible definition and modification of the schema, the attachment of procedural data (e.g. rules), and arbitrary user-defined relationships between objects (Woelk, Luther and Kim 1987; Banerjee et al. 1987).

The approach looks very ambitious. There are no solutions published yet how this all can be accomplished. As several other authors point out, e.g. (Larson 1988; Orenstein 1988), object-orientation may provide the appropriate framework, but it does not itself solve the problems of multimedia management. For example, it does not address how content analysis of multimedia is to be handled.

Another current topic involving multimedia data is *Hypertext*. The concepts of Hypertext are relatively old; they have been transferred to computer systems since the 60's (Nelson 1980). Originally intended to manage arbitrarily linked text segments, Hypertext has been extended to manage images and sound as well ("Hypermedia"). An overview on the numerous projects is given in (Conklin 1987). While there are claims of various kind about hypertext, hypertext is not a general multimedia DBMS. It merely provides a data structure for linking some items together.

Masunaga has developed a framework that helps to classify and compare the projects (Masunaga 1987). He assumes different databases for the different media. They are integrated using an additional object-oriented database that refers to them. There is either a single (extensible) DBMS managing all these different databases ("single DBMS architecture"), a "primary" multimedia DBMS that calls the "secondary" media-specific DBMS as subroutines ("primary-secondary DBMS architecture"), or a collection of cooperating DBMS accessing each other via Remote Data Access ("federated DBMS architecture").

4. Proposed Approach for a Multimedia DBMS

4.1. Concepts

The complexity of the problem and the shortness in the history of research in handling multimedia data result in the current situation that there are not generally accepted solutions at this time. To allow us to make progresses, most projects attempted to develop specialized system for a special application to reduce complexity (e.g. a system for office environment or engineering environment). While this is one possible approach, we wish to propose a different direction which we think may be more fruitful in developing a general system for diversified applications.

The approach in this paper illustrates our alternative to develop a basic functional DBMS that can handle multimedia data for any application. It is analogous to the way how one constructs a normal DBMS for handling formatted data by concentrating on developing a DBMS with the basic functions for retrieving, searching, and managing multimedia data. As complexity in multimedia data handling is a major issue, we shall discuss a little about this aspect first.

The fundamental difficulty in handling multimedia lies in the problem of handling the rich semantics that is contained in the multimedia data. The semantics that can be associated with the traditional, formatted data is very restrictive. For example, the value of 5 in the data item for the attribute of weight in pounds can mean only 5 pounds in weight, and nothing more. If further semantics in the interpretation of the data is to be done, it would be at a different level. Handling semantics is difficult and complex, and it gives rise to the research of semantic data modeling the solution of which is not expected to come soon.

Multimedia data is unavoidably and intrinsically tied to a very rich semantics. A simple extension from formatted data into textual data, as we do in information retrieval, for example, already brings us much difficulty. Information retrieval scientists have spent a long number of years trying to solve this problem with some good success. Extending into other kind of media such as image is much more difficult. To illustrate such a difficulty, let us take a simple picture containing a dog and a cat in action. Given such a picture, how are we to know if the dog is chasing the cat or vice versa? Or are they simply playing with each other? To answer queries like these, a person must draw from a very rich experience one has encountered in life and perform integration, analysis, synthesis, and even extrapolation of his or her knowledge to derive a good answer. This kind of process requires high intelligence. As a result, persons with limited experience and knowledge, such as a child or someone who has not been exposed to many things in the world, will not be able to give good answers to queries on multimedia data. In fact, given the same picture, persons with different backgrounds will likely give different answers with respect to the content of it. For reasons of this kind, none of the cited projects intended to address the contents of multimedia data.

Systems with this kind of capability to answer multimedia queries are definitely beyond today's technology. We can, however, do the next best thing. As the proverb says, "a picture is worth ten thousand words", meaning people can describe the picture in a different medium, e.g. text, although one would never have exactly the same thing, feeling- or meaning-wise. Nevertheless, people can abstract the contents of the image data, sound data, or other forms into words or text. After we have the text description, we shall assume to have the "equivalent" of the original multimedia data, for searching and analysis purposes. We shall then apply the techniques of information retrieval and the formatted data processing to these multimedia data.

4.2. Proposed Design

In this section, we shall describe the approach just mentioned in some detail. For the sake of clarity and simplicity, we shall consider only the multimedia data type image, although the approach can be extended directly into other forms as well. Because of the flexibility in the relational model, it will be the data model used for designing our multimedia DBMS. Further, as we are interested in exploring the development of a basic DBMS for handling multimedia data at the present time, we are discussing only the programmer level interface and are not concerned for the end-user kind at this time.

To extend the relational system's capability we propose to add a new data type image to the system. IMAGE is therefore a new attribute domain and every image shall use it for data definition. An object can have the attribute image, but a picture (image) can also be a stand-alone object as defined in one of the three ways indicated below:

(1) OBJECT (<u>O-ID</u> , ..., O-IMAGE)

(2) OBJECT (<u>O-ID</u> , ...)
OBJECT-IMAGE (<u>O-ID, O-IMAGE</u>)

(3) OBJECT (<u>O-ID</u> , ...)
IMAGE-OBJECT (<u>I-ID</u> , I-IMAGE)
IS-SHOWN-ON (<u>O-ID, I-ID</u> , COORDINATES, ...)

where O-ID is object identifier and is the primary key or part of the primary key. The three alternatives allow users to represent images in different ways. For example, one can represent an image as a simple attribute of some object using option 1. This is appropriate if there is a one-to-one relationship between images and objects, e.g. a database with an employee photo for each employee record. One can use option 2 to represent an object that has a number of images, or use option 3 to represent images that show more than one objects (i.e. that is composed of smaller images) and images with unknown contents. Each of the options has advantages and disadvantages. The first one is the simplest but is most restrictive and the third one the most flexible but the most complicated in manipulation. Which one to use depends entirely on the application. Graphically the three options are presented in figure 3. The dotted line illustrates a one-to-many relationship between the tuples of the respective relations.

Not all operations of the relational algebra can be performed directly on the data type image. For example, direct joins on the image will not be allowed. Projection means either the image is kept or dropped completely. Selection on the value of the image will be treated differently than the normal formatted data and we will go into that further later. First, let us discuss a bit more how images are actually kept.

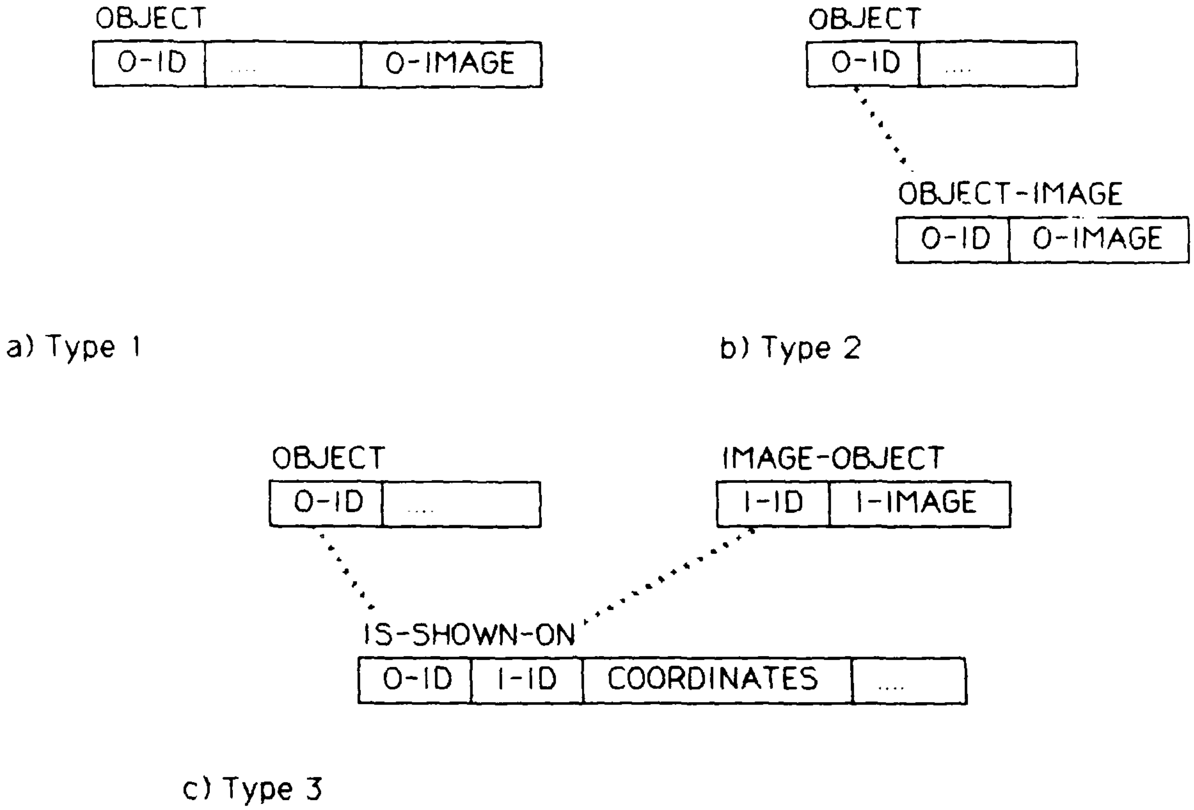

Figure 3: The Three Relation Schema Options for Storing Images

As mentioned earlier, each image will have three parts, although logically as one unit. The parts are *registration data, description data,* and *raw data* of an image (figure 4). Ideally the registration data are recorded automatically from camera settings when the image is captured, but in some cases the user might have to key them in. Description data can only be provided by the user. Registration data is mandatory while description data is optional. The raw data, which may be a bitmap of an image, cannot be queried nor used for any query operation. It can be invoked to be presented for editing and modification by a special module outside the DBMS. Edited images will be entered into the DBMS as new image values with the registration and description data adjusted accordingly.

As indicated in the diagram (fig. 4), registration data include the kind of information that is needed by the image handling device to display the image properly. Such information as color depth, image resolution, image source, etc. will be included in the registration data. Generally this part of the information deals with the physical aspect of displaying the image. The description data, on the other hand, deals with the content of an image. It is composed of *phrases and*

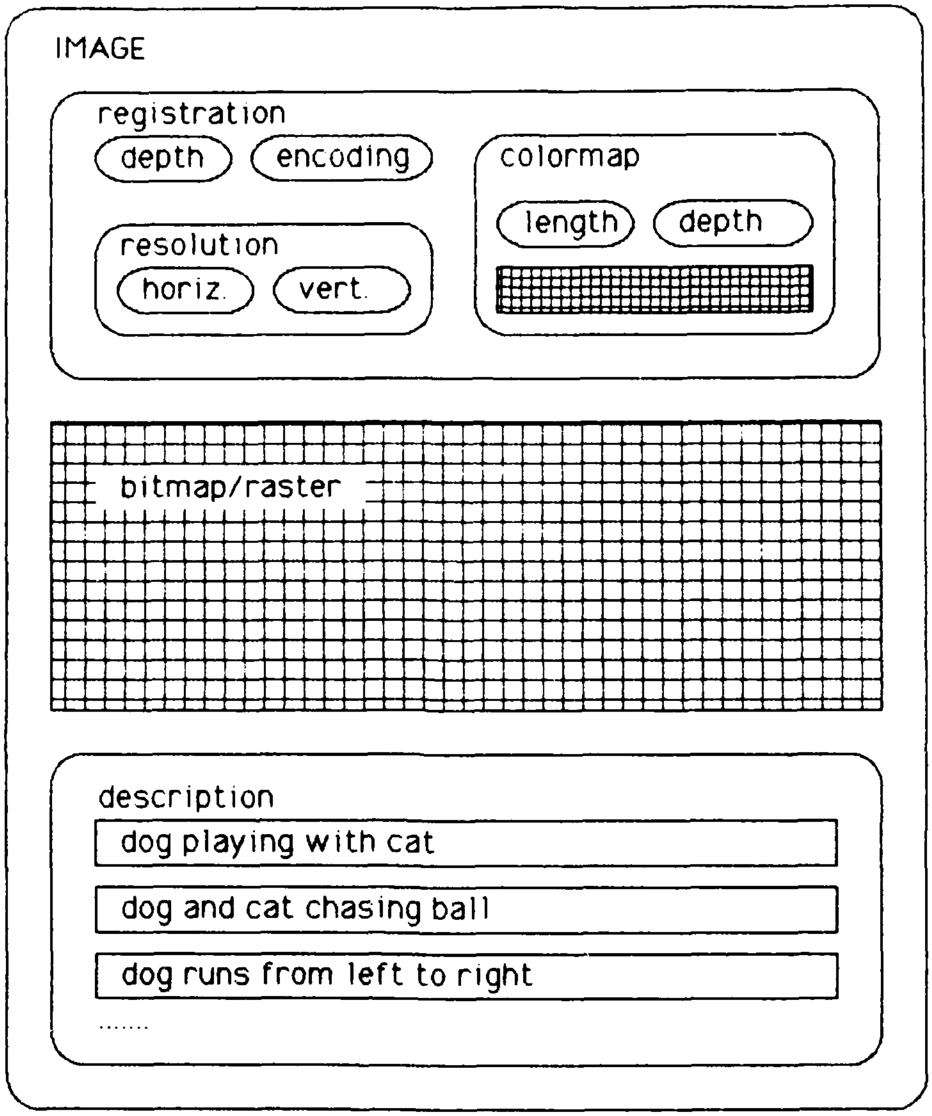

Figure 4: Conceptual View of an Instance or Value of the Data Type Image

sentences, with keywords to be a degenerate case. The selection of keywords, if desired, is simple and is done the same way a person does in selecting keywords for an article or a document. Entry of phrases and sentences is slightly more complex. Here, to avoid unnecessary complication, we shall restrict each sentence or phrase to be independent of the others, as in the notion of elements in a set. This naturally leads to redundancy in the definition of nouns in an image. But this additional work is well worthwhile as we can avoid other more difficult complications by so doing. As an example, consider an image of a dog and a cat in action. One may have in the description data the following:

dog playing with cat.
dog and cat chasing ball.
dog runs from left to right.
cat runs from right to left.
ball is between dog and cat.
ball bounces up in the air.
dog and cat are in the backyard of a house.

In this manner, a person can provide the system as much information as desirable and useful.

Special operations must be defined for access to the registration and description data. Similar to the delivery of the raw data of an image to the special image handler, we propose an inverse function to handle the construction of an image value (including the registration data) with the operation

```
CONSTRUCT_IMAGE (resolution, pixel-depth, encoding, colormap-size,
        colormap-depth, colormap, pixel-matrix).
```

This operation will be restricted to the use with the database operations of updates and inserts as indicated in the following:

```
UPDATE IMAGE-OBJECT
SET I-IMAGE = CONSTRUCT_IMAGE ($resolution, $depth, RGB_REAL_32, 256, ... )
WHERE I-ID = 1122;

INSERT (2001, CONSTRUCT_IMAGE ($resolution, 24, IHS_INT_8, 0, ... ))
INTO IMAGE-OBJECT;
```

The image created by the function cannot be assigned to program variables. The $ sign represents a program variable in the function and the parameters with capital letters indicate named constants.

Similar to the definition of the raw data and registration data, description data is also created by a special operation. Their definition can be indicated as in the following example:

```
UPDATE IMAGE-OBJECT
SET I-IMAGE = ADD_DESCRIPTION (I-IMAGE,

                    { dog playing with cat,
                      dog and cat chasing ball,
                      dog runs from left to right,
                      cat runs from right to left,
                      ball is between dog and cat,
                      ball bounces up in the air,
                      dog and cat are in the backyard of a house } )

WHERE I-ID = 1122;
```

The insert operation is very much similar and will not be detailed.

To allow the select operation to make use of the information in the registration and the description data, additional operations will be needed. They can also be used to retrieve the values in the IMAGE attribute into program variables. We propose the GET functions like the following:

```
GET_RESOLUTION(IMAGE attribute): resolution_type;
GET_DEPTH (IMAGE attribute): integer;
GET_ENCODING (IMAGE attribute): encoding_type;
etc.
```

These operations can then be used with the select operation as follows:

```
SELECT GET_8BIT_RASTER (I-IMAGE), GET_8BIT_COLORMAP (I-IMAGE)
INTO $rgb_screen, $rgb_colormap
FROM IMAGE-OBJECT
WHERE I-ID > 35
AND GET_ENCODING (I-IMAGE) = IHS_INT_8;
```

To use the description data for selection, we propose the additional operations to be embedded into the select operation. As an example, consider the image of the dog and cat playing as given above. If we want to retrieve all pictures that show a dog playing with a cat, and both are chasing after a ball, we may want to define a query as follows:

```
SELECT GET_RESOLUTION (I-IMAGE)
INTO $resolution
FROM IMAGE-OBJECT
WHERE CONTAINS (I-IMAGE,
                dog & play* & cat,
                dog & chas* & ball,
                cat & chas* & ball);
```

The & symbol means that there may be other words between the two words, but they are not used for the selection process, and the * symbol means a match of strings of arbitrary length is needed with the subset as given before the * sign. Thus, if one describes the picture as "a brown dog running in the backyard is playing with a black cat", this phrase will satisfy the first pattern of the select operation, and so does the phrase "dog playing with cat" given in the previous example.

Naturally much more syntax would have to be defined to make the above operations complete. For example, the search pattern definition must allow Boolean operations. The example just given is intended to mean logical intersection operation for the keywords and the phrases, i.e. an image is retrieved only if *each* search pattern is satisfied by one if its description phrases. A different syntactic structure is needed for the union operation and a combination of union and intersection operations. At this time, we have not designed all the different operations that are needed to handle multimedia data in the form as proposed in the paper. It is our intention to keep these operations as small and as simple as useful, but at the same time have the necessary basic functions for the various operations. The above is meant to serve as illustrations of how we intend to make use of the information retrieval and database technologies to handle multimedia data.

5. Integration with other Advanced Techniques

To be able to process the contents of the multimedia data, one needs to know what is represented in the data. Our proposal is to ask the users to describe them in their natural language. After all the multimedia data is described in this manner, we have a collection of data content in a natural language form. To be able to make good use of such information, it should be represented in a more structural way. This becomes the problem of *knowledge representation* intensively studied in the artificial intelligence discipline, though no broadly accepted solution is available.

Moreover, as information is described in language form, translation into some form of semantic network is necessary, as this gives us a much more precise representation of the information. This means dealing with natural language understanding, one of the important areas of artificial intelligence. One must also integrate the different pieces of information into a coherent form for searching as the inputs are given in discrete sentences each of which is assumed to be *independent* of the others.

Given the above, one must find a good data structure to store the information contained in the system about the multimedia data. This is expected to be a very large database as much information is contained in one single piece of multimedia data like an image, and a collection of the multimedia definitely is voluminous for any practical application.

Searching for answers in a system such as the one just discussed is expected to require much reasoning or rationalization. Moreover, it is also true that often the system would have to present a result that is not exactly matching the query posed but may be close. This happens in information retrieval and definitely will occur here. Algorithms to evaluate the closeness of the potential result to the query must be developed for the system. Techniques to find good potential results are needed. For example, synonym definition is believed to be necessary in production systems.

A more difficult problem is *extracting knowledge* directly from the multimedia data. Many researchers are concerned with this problem (Sheth 1988, Larson 1988, Phillips 1988, Orenstein 1988). This problem is unfortunately extremely complex and concrete, broadly capable solutions are not expected to be available for many years to come. In fact, even the smaller problem of translating or converting data from one medium to another will not come easy (Masunaga 1987). This task, however, is generally deemed to be the proper domain of AI research rather than database.

As a matter of fact, *multiple representations* of the same object in different media may be a goal one would want. In this way users can ask for objects to be presented in a medium of his choice. To achieve this goal requires us to solve the object representation and conversion problem which at this time is hardly addressed even in the research stage. In many circumstances, this goal is not achievable. For example, if an image is stored in the system in pictorial form, it is not possible for the system to present it in an audio manner.

Architectural design of a multimedia system is also an open issue. As we today do not understand the various, possible applications of a multimedia system, it is imperative for us to have a system architecture that is modular and expandable or extensible. Studies for system of this kind are being conducted (Batory et al. 1986; Batory 1987; Lindsay, McPherson and Pirahesh 1986; Stonebraker and Rowe 1987). But a broader goal is needed if we are to develop a system that would allow us to integrate new techniques, such as finding an answer to a query with uncertain information or presenting a result that is only approximate, into the system without major modification to the system.

Last but not least is the issue of the *interface* for a multimedia DBMS. Although our approach for now is to give a programmer's interface, such an interface is not expected to be used by the end-users. As applications for a multimedia DBMS are expected to be very diversified, it is probable that different kinds of interfaces may be needed for different categories of applications. For example, using such a system for office automation may require a different interface than using it for engineering design. It is also possible that multiple interfaces may coexist within a single system.

6. Conclusion

In this paper we have analyzed the characteristics and the application of multimedia data and presented an approach which capitalizes on the advances already made in database technology and information retrieval techniques. Both of these disciplines have a long history of research and application in the production environment and much has been learned.

Because of the complexity of the information content in each single piece of multimedia data like an image, it is practically impossible for us to expect a system to extract information directly from the raw multimedia data. Moreover, for a given piece of multimedia data, its content can be interpreted in many, many completely different ways, depending on the experience of its interpreter. This is well recognized by the psychologists and psychiatrists for making picture recognition one of the aspects in their analysis. Our approach is to have the system users represent the multimedia data content information in keywords and natural language sentences, a method of description that is familiar to and practiced by all of us. This eliminates the major hurdle which can be the block for advances in multimedia application development.

Once this step is done, there is a rich reservoir of knowledge on dealing with the problems of handling multimedia data. Database technology allows us to develop data structures to store them, although most likely new methods may have to be developed to provide the kind of system performance we wish to have. Information retrieval techniques can be applied to some of the problems such as text search and evaluation of the "closeness" of the potential result to the posed query. Researches in semantic networks are also expected to be useful.

This does not mean that all problems are solved. As stated in the last section, many problems remain. However, it is believed that the approach presented in this paper has simplified many of the complex and difficult problems into a more manageable form.

Literature

Ballard, D. H., and Brown, C. M. *Computer Vision*. Prentice-Hall, Englewood Cliffs, 1982.

Banerjee, J.; et al. "Data Model Issues for Object-Oriented Applications." *ACM TOOIS*, Vol. 5, No. 1, Jan. 1987, pp. 3-26.

Banerjee, J.; Kim, W.; and Kim, K.-C. "Queries in Object-Oriented Databases." *Proc. 4th Int. Conf. on Data Engineering* (Los Angeles, Febr. 1988), IEEE CS Press, order no. 827, pp. 31-38.

Batory D. S. "A Molecular Database Systems Technology." University of Texas at Austin, Technical Report TR-87-23, June 1987.

Batory, D. S.; Barnett, J. R.; Garza, J. F.; Smith, K. P.; Tsukuda, K.; Twichell, B. C.; and Wise, T. E. "Genesis: A Reconfigurable Database Management System." University of Austin at Texas, Technical Report TR-86-07, March 1986 (also to appear in IEEE Trans. on Softw. Eng.).

Chang, S.-K., and Kunii, T. K. "Pictorial Data-Base Systems." *IEEE Computer*, Vol. 14, No. 11, Nov. 1981, pp. 13-21.

Christodoulakis, S., and Faloutsos, C. "Design and Performance Considerations for an Optical-Disk Based, Multimedia Object Server." *IEEE Computer*, Vol. 19, No. 12, Dec. 1986, pp. 45-56.

Christodoulakis, S.; Ho, F.; and Theodoridou, M. "The Multimedia Object Presentation Manager of MINOS: A Symmetric Approach." *Proc. ACM SIGMOD '86 Int. Conf. on Management of Data* (Washington, D.C., May 1986), ACM SIGMOD Record, Vol. 15, No. 2, June 1986, pp. 295-310.

Christodoulakis, S.; Theodoridou, M.; Ho, F.; Papa, M.; and Pathria, A. "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System." *ACM TOOIS*, Vol. 4, No. 4, Oct. 1986, pp. 345-383.

Conklin, J. "Hypertext: An Introduction and Survey." *IEEE Computer*, Vol. 20, No. 9, Sept. 1987, pp. 17-41.

IBM System 360 and System 370 (OS) Storage and Information Retrieval System. General Information, Program Product 5734-XR3, 1st ed., Stuttgart, Germany, 1971.

Lancaster, F. W., and Fayen, E. G. *Information Retrieval On-Line*. Melville Publ. Comp., Los Angeles, CA, 1973.

Larson, J. A. "Browsing Unstructured Data." *Proc. 4th Int. Conf. on Data Engineering* (Los Angeles, Febr. 1988), IEEE CS Press, order no. 827, p. 600.

Lindsay, B.; McPherson, J.; and Pirahesh, H. "A Data Management Extension Architecture." IBM Research Report RJ5436 (55565), Dec. 1986.

Lum, V. Y.; Wu, C. T.; and Hsiao, D. K. "Integrating Advanced Techniques into Multimedia DBMS." report no. NPS52-87-050, Naval Postgraduate School, Monterey, CA, Nov. 1987.

Lum, V.; Dadam, P.; Erbe, R.; Guenauer, J.; Pistor, P.; Walch, G.; Werner, H.; and Woodfil, J. "Design of an Integrated DBMS to Support Advanced Applications." *Proc. Int. Conf. on the Foundations of Data Organization* (Kyoto, Japan, May, 1985), pp. 21-31.

Macleod, I. A. "A Database Management System for Document Retrieval Applications." *Information Systems*, Vol. 6, 1981, pp. 131-137.

Masunaga, Y. "Multimedia Databases: A Formal Framework." *Proc. IEEE CS Office Automation Symp.* (Gaithersburg, MD, Apr. 1987), IEEE CS Press, order no. 770, Washington, 1987, pp. 36-45.

Nelson, T. H. "Replacing the Printed Word: A Complete Literary System." In S. Lavington (ed.), *Information Processing 80*, Proc. IFIP Congress, North-Holland, Amsterdam 1980, pp. 1013-1023.

Orenstein, J. A. "Can We Meaningfully Integrate Drawings, Text, Images, and Voice with Structured Data?" *Proc. 4th Int. Conf. on Data Engineering* (Los Angeles, Febr. 1988), IEEE CS Press, order no. 827, p. 603.

Phillips, B. "Multimedia Systems and Text." *Proc. 4th Int. Conf. on Data Engineering* (Los Angeles, Febr. 1988), IEEE CS Press, order no. 827, p. 601.

Raskin, R., and Stone, M. D. "Picture Databases: Coming into Focus." *PC Magazine*, Vol. 6, No. 14, Aug. 1987, pp. 341-376.

Rosch, W. L. "Voice Recognition: Understanding the Master's Voice." *PC Magazine*, Vol. 6, No. 18, Oct. 1987, pp. 261-308.

Salton, G., and McGill, M. J. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York 1983.

Schek, H.-J., and Pistor, P. "Data Structures for an Integrated Data Base Management and Information Retrieval System." *Proc. 8th Int. Conf. on VLDB* (Mexico City, Sept. 1982), pp. 197-207.

Sharp, H. S. (ed.) *Readings in Information Retrieval*. The Scarecrow Press, New York & London 1964.

Sheth, A. "Managing and Integrating Structured and Unstructured Data: Problems of Representation, Features, and Abstraction (A Position Paper)." *Proc. 4th Int. Conf. on Data Engineering* (Los Angeles, Febr. 1988), IEEE CS Press, order no. 827, pp. 598-599.

Stonebraker, M., and Rowe, L. A. "The POSTGRES Papers." University of California at Berkeley, Technical Memorandum No. UCB/ERL M86/85, June 1987.

Winograd, T. *Language as a Cognitive Process*. Volume I: Syntax, Addison-Wesley, Reading, MA, 1983.

Woelk, D., and Kim, W. "Multimedia Information Management in an Object-Oriented Database System." *Proc. 13th Int. Conf. on VLDB* (Brighton, England, Sept. 1987).

Woelk, D.; Kim, W.; and Luther, W. "An Object-Oriented Approach to Multimedia Databases." *Proc. ACM SIGMOD '86 Int. Conf. on Management of Data* (Washington, D.C., May 1986), ACM SIGMOD Record, Vol. 15, No. 2, June 1986, pp. 311-325.

Woelk, D., and Luther, W. "Multimedia Database Requirements - Rev.0." MCC Technical Report No. DB-042-85, Austin, Texas, 1985.

Woelk, D.; Luther, W.; and Kim, W. "Multimedia Applications and Database Requirements." *Proc. IEEE CS Office Automation Symposium* (Gaithersburg, MD, Apr. 1987), pp. 180-189.

Yankelovich, N.; Meyrowitz, N. K.; and van Dam, A. "Reading and Writing the Electronic Book." *IEEE Computer*, Vol. 18, No. 10, Oct. 1985, pp. 15-30.

Distribution List

SPAWAR-3242 Attn: Phil Andrews Washington, DC 20363-5100	1
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
Center for Naval Analyses 4401 Ford Ave. Alexandria, VA 22302-0268	1
Director of Research Administration Code 012 Naval Postgraduate School Monterey, CA 93943	1
John Maynard Code 402 Command and Control Departments Naval Ocean Systems Center San Diego, CA 92152	1 1
Dr. Sherman Gee ONT-221 Chief of Naval Research 800 N. Quincy Street Arlington, VA 22217-5000	1
Leah Wong Code 443 Command and Control Department Naval Ocean Systems Center San Diego, CA 92152	1
Klaus Meyer-Wegener Code 52Mw Naval Postgraduate School Monterey, CA 93943	1

Vincent Y. Lum 50
Code 52Lu
Naval Postgraduate School
Monterey, CA 93943

C. Thomas Wu 1
Code 52Wq
Naval Postgraduate School
Monterey, CA 93943