SEPTEMBER 1984 VOL. 7 NO. 3

a quarterly bulletin of the IEEE computer society technical committee on

Database Engineering

Contents

Database Engineering Bulletin is a quarterly publication of the IEEE Computer Society Technical Committee on Database Engineering. Its scope of interest includes: data structures and models, access strategies, access control techniques, database architecture, database machines, intelligent front ends, mass storage for very large databases, distributed database systems and techniques, database software design and implementation, database utilities, database security and related areas.

Contribution to the Bulletin is hereby solicited. News items, letters, technical papers, book reviews, meeting previews, summaries, case studies, etc., should be sent to the Editor. All letters to the Editor will be considered for publication unless accompanied by a request to the contrary. Technical papers are unrefereed.

Opinions expressed in contributions are those of the individual author rather than the official position of the TC on Database Engineering, the IEEE Computer Society, or organizations with which the author may be affiliated.

Membership in the Database Engineering Technical Committee is open to individuals who demonstrate willingness to actively participate in the various activities of the TC. A member of the IEEE Computer Society may join the TC as a full member. A non-member of the Computer Society may join as a participating member, with approval from at least one officer of the TC. Both full members and participating members of the TC are entitled to receive the quarterly bulletin of the TC free of charge, until further notice.

Changes to the Editorial Staff

Since its revival in 1981, Database Engineering has gained a reputation as a timely and carefully written publication, covering current research and development work in the database area. Won Kim has been an effective, hard-working, and knowledgable editor, deserving much of the credit for the success of DBE. As the new Editor-in-Chief of DBE, I hope to continue these traditions.

One of the major strengths of DBE is its staff of Associate Editors, active researchers all, who are responsible for editing individual issues. Don Batory, Randy Katz, and Dan Ries, having made substantial contributions as members of this staff, are now moving on. As a fellow editor, I have appreciated their enthusiasm and breadth of expertise.

Fred Lochovsky, who has already edited one issue on office systems, will remain as an Associate Editor, and will be joined by three newcomers: Haran Boral, C. Mohan, and Yannis Vassiliou. I am looking forward to working with all of them.

Gio Wiederhold will be the new Chairperson of the Technical Committee on Database Engineering, replacing Bruce Berra as coordinator of TC activities. One of Gio's first projects is to increase the circulation of DBE.

Our tentative schedule of upcoming issues is:

12/84 (Reiner) Database Design Aids, Methods, Environments
3/85 (Boral) DBMS Performance
6/85 (Mohan) Concurrency Control and Recovery in DBMS's
9/85 (Vassiliou) Natural Languages and Databases
12/85 (Lochovsky) Object Oriented Systems and DBMS's

Our orientation will continue to be towards engineering aspects of databases, rather than abstract theory. Although submissions to DBE are not subject to a formal review process, the editors generally read articles very carefully, and work with the authors to achieve both clarity and brevity.

David Reiner

David Reiner
Cambridge, Massachusetts
September, 1984

Multimedia Data Management

Multimedia Data Management has the potential of providing office workers with integrated access to text, voice, graphics, picture and conventional data processing data. The realization of this potential requires the integration of multimedia technologies from the data storage levels such as optical disks, through the indexing and accessing of that stored data through convential database access methods, and to the presentation of the multiple types of data through advanced user interfaces. This issue presents five papers which describe the integration and use of the various multimedia technologies.

The first three papers describe the use of optical disks in support of multimedia systems. The paper by Koji Izawa of Toshiba in Japan presents an overview and some of the technical parameters of a commercially available document image filing system that uses an optical disk. Izawa points out that the extensive use of hand-written documents in Japan motivated the early development of image systems. The next paper by David Kramlich of CCA describes a spatial data management system that is being used on-board an aircraft carrier. The system integrates positional database data with various resolution map images stored on optical disks. The third paper by Malcom Easton of IBM then describes some of the research and implementation issues that must be addressed in using optical disks for archival storage of conventional text and database records. Specifically, Easton discusses the problems of write error space management.

The last two papers describe more general multimedia systems. Forsdick, Thomas, Robertson and Travers of BBN present experiences with the design and implementation of multimedia document model. They comment extensively on some of the user interface issues. The fifth paper by Stavros Christodoulakis describes a multimedia system being implemented at the University of Toronto. Christodoulakis provides some insights on querying and content addressability of a multimedia storage and presentation system.

Daniel R. Ries
Associat Editor

Document Image Filing System Utilizing Optical Disk Memories

Koji Izawa

Toshiba Research and Development Center
1 Komukai Toshiba-cho, Saiwai-ku, Kawasaki 210, Japan
Phone: 044-511-2111

Introduction

In recent years, the number of documents used in business offices has increased greatly. Up to this time, microfilm systems have been used to save space and to manage the storage of these documents. Although microfilm systems have an advantage in regard to stability, they have some defects in the real time operation. They require a developing process for storage and a complicated mechanism for automatic retrieval. In addition, due to their non-electrical magnetic representation, editing or transmitting document images is not normally possible. Document image files utilizing optical disk memories have been developed to address these difficulties.

This paper first mentions application areas and their characteristics for document image files. Next, an overview of a document image filing system utilizing an optical disk memory is presented.

Application Areas and Their Characteristics for Document Image Files

Application areas for document image files can be divided into two categories. One is where the number of documents is very large, but where the types of documents are limited. Literature, patent journals or drawings fall into this category. In these cases, the collection and classification of the documents are accomplished by a specific organization, working under a consistent concept. Their classification hierarchy is usually defined beforehand, and once it is defined, it seldom needs to be altered. These files can be called "static files". For mechanization of such files, a mass storage means becomes of prime importance. Though storing these types of documents can be carried out in batch operation, time for retrieval should be short. Specialized languages for document retrieval are required to handle the various retrieval requirements for many and unspecified persons.

The other category includes business office files. In this case, the number of documents is less than in the former, but the kinds of documents and the ways to classify them are varied according to the work accomplished by the office. Futhermore, with an increase in business or office organization, the kinds of necessary documents and their classfication will both change gradually. The more active the office is, the faster the change is. In this sense, these files can be called "dynamic files". For mechanization of such files, the following items are necessary.

1. Document images can be stored and referred to in real time.

2. Office workers themselves can make up their own classification hierarchy of documents.

3. The classification hierarchy can be grasped at a single glance, and documents can be filed very easily matching this classification.

4. The classification hierarchy can be altered very easily, as business requirements change.

Thus, different functions are required for these different types of document image files. Up to this time, microfilm systems have been used for the static files. For office automation, mechanization is needed for the dynamic files.

An Overview of a Document Image Filing System

Basic Technology: A document image filing system usually consists of an image scanner, an image printer, an image display, an optical disk memory and a system controller. As a disk medium, a DRAW (direct read after write) type disk is used in many systems. This medium is not erasable. However, this characteristic is rather a merit from the evidence preservation viewpoint. Additionally, though it is nonerasable, the large amount of storage capacity makes it possible to appear erasable, by writing the renewal data into new areas.

We next describe an optical disk memory, image processing techniques and a document managing software system.

An Optical Disk Memory System: In an optical disk memory system, high recording density has been realized by utilizing a laser beam as a means of recording and retrieving. A laser beam, focussed into a lum diameter high energy density point on the disk through optics, makes a physical or an optical change whose size is in micron order on a storage medium. In reading, a lower power laser beam is irradiated on the disk. By sensing the reflection, a change in the status on the recording medium can be detected optically. A recorded pit size is usually 0.8 in width and lum in length. A track pitch is 1.6 - 2um. This recording density is 10-100 times that for magnetic disks. Total storage capacity is 10 -10 bits in a 12 inch-diameter disk. Figure 1 shows an optical disk memory blockdiagram.

Image Processing Techniques When a letter size document is scanned in 200 lines per inch resolution, the image data amounts to 4M bits in all. Therefore, after being compressed by a factor of eight, image data is recorded into an optical disk. Because this process must be carried out at high speed, a special decoder/encoder circuit is needed.

Additionally, in the document image files, a document image must be displayed fully and with high quality. However, displaying a document with no manipulation needs a 2,400 scan lines display. It is very expensive. Therefore, a document image reducing technique with little quality degradation is important. One is high quality displaying technique, involving resampling a birary document image into gray levels.

In addition, techniques regarding image magnification, rotation, edition or multi-window displaying are also needed.

Software for Managing Documents: In ordinary information retrieval systems, keyword based information management is usual. However, in business office files, as already described, it must be easy to make up the documents classification hierarchy and to alter it as desired. Additionally, it is desirable to preserve the usability of paper files, such as the possibility of leafing through the pages or inserting a bookmarker somewhere. That is, it is important to install man-machine interfaces appealing to human intuition.

Document Image Filing System Specifications: A document image filing system utilizing an optical disk memory was placed on the market in January of 1982 by Toshiba Corporation in Japan, followed by Matsushita and Hitachi. The reason development is vigorous in Japan is that Hand-written documents are dominent there. Next, an outline of a system will be described, taking Toshiba's recently produced/DF3200 as an example.

Figure 2 is a photograph of the system. Figure 3 shows the system blockdiagram.

Characteristics:

(1) Optical Disk Memory System
- Storage capacity is sixty thousand 8 1/2 inch size document sheets per disk.
- Up to 8 disk drives can be connected.
- An optical disk autochanger can handle 25 disks. Up to 4 autochangers can be connected. In this case, 6 million document images can be managed.

(2) Scanner and Printer
- Maximal imput or output document size is 11.7 x 16.5 inch size.
- The resolution is 400 lines or 200 lines per inch. Therefore, large size documents, such as drawings, can be filed with high quality assured.
- An automatic document feeder is installed

(3) Document Managing
Documents are managed under a hierarchy which consists of cabinets, binders, documents and pages. Figure 4 shows the hierarchy. One side of an optical disk corresponds to one cabinet. In one cabinet, up to eight binders can be defined. One binder can manage up to 30,000 documents. One document consists of several pages, and a page number is put on automatically In each binder, individual keyword structure can be defined. Each document has a unique comment included as an identifier. Additionally, bookmarkers can be inserted into frequently referenced documents. The basic retrieval procedure is described below.

1. Documents selection utilizing the classification hierarchy.
2. Retrieval using a keyword formula for documents in a binder.
3. Turning over pages, one after another, in selected documents.

Figure 5 shows documents retrieval methods used in this system. Figure 6 shows DF3200 elemental characteristics.

Document Image Filing Systems Applications: Document image filing systems are utilized in various manners. In manufacturing industries, they are utilized for managing drawings, patent journals, technical materials or operation manuals. Besides, they are also used in information service systems by real estate dealers, a map retrieving system at a fire station, clinical chart management systems at hospitals and so on.

Conclusion

An outline of a document image filing system utilizing an optical disk memory has been presented. Future developments will require expanded capacities of optical disk memories and erasable medium. As these requirements are met, integrated document file systems to manage image text and numerical data will be developed. As the capacities of optical disks continue to grow, they will be used for more conventional secondary computer memories.

1) K. Izawa et al., "Visually Assisted Document File System", Proc. of 3rd International Display Research Conference, Kobe, Japan, October 1983.

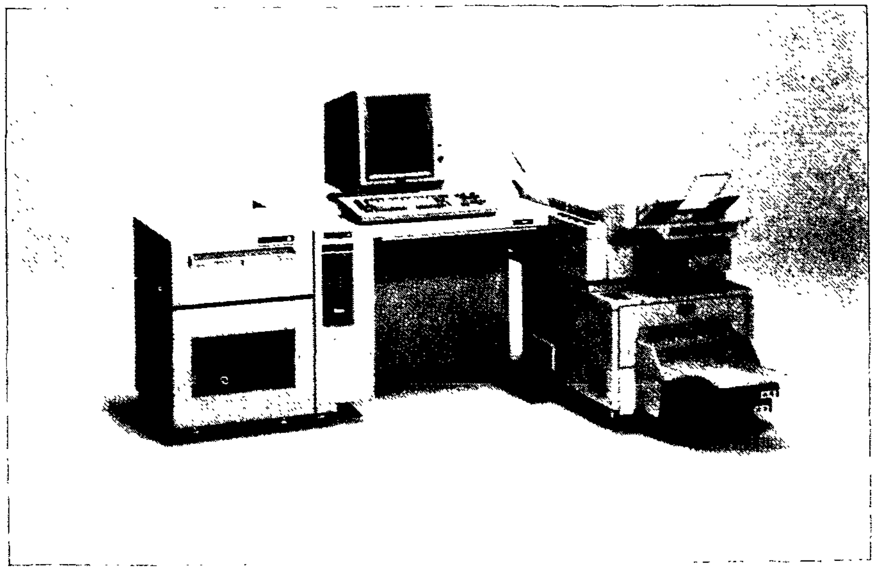


Fig. 2 DF3200

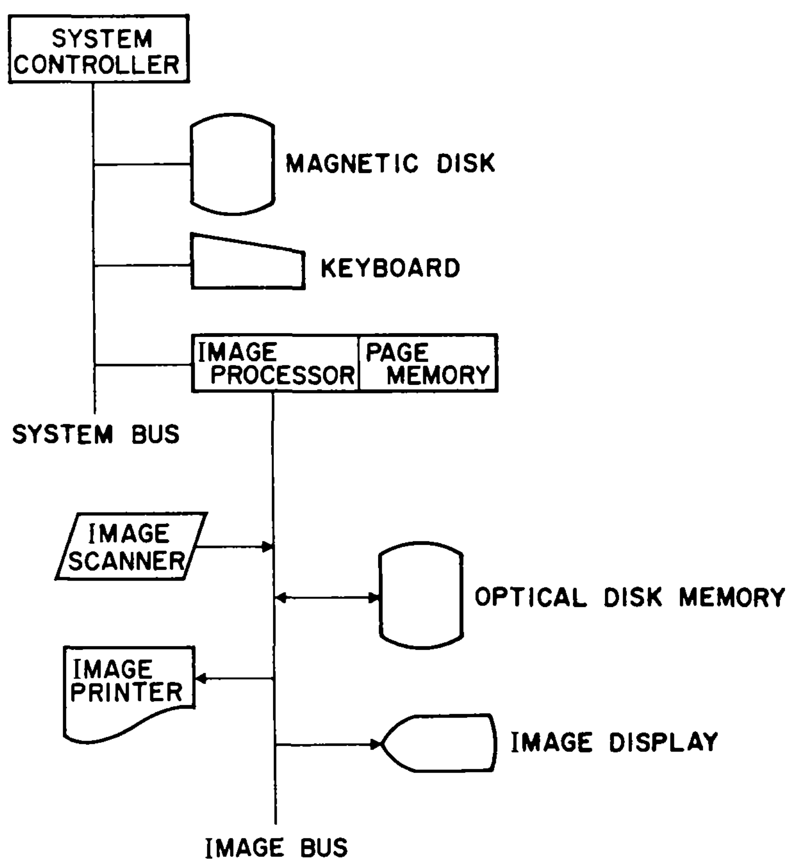


Fig. 3 Systerm blockdiagram

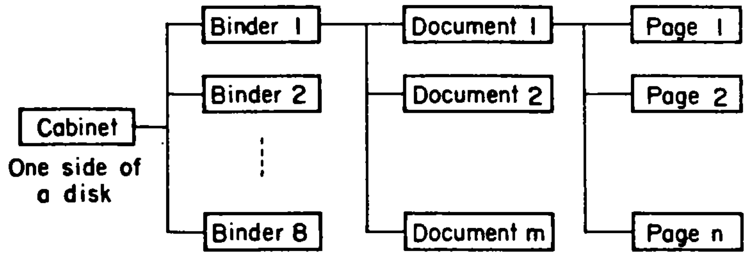


Fig. 4 File hierarchy

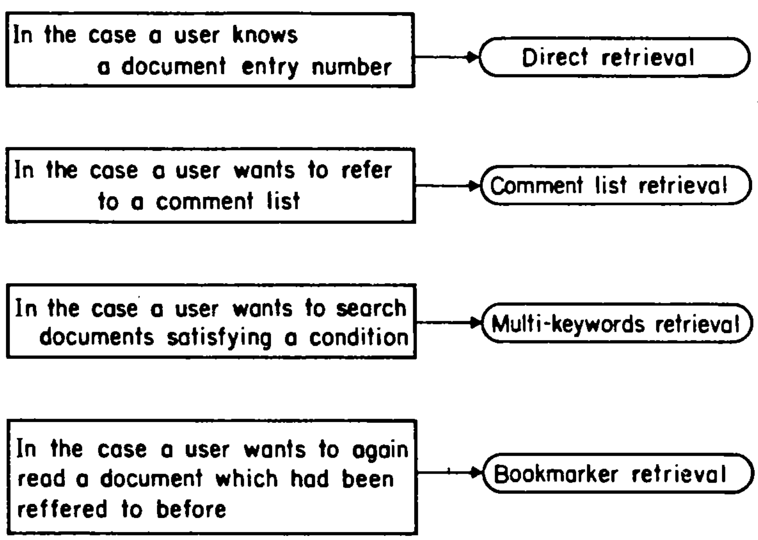


Fig. 5 DF3200 document retrieval methods

Scanner	Document Size Scanning speed Scanning resolution	Max. 11.7×16.5 inch size 3 sec./letter size 400 lines/inch or 200 lines/inch
Printer	Printing method Printing size Printing speed Printing resolution Printing paper	Electro-photographic method Max. 11.7×16.5 inch size 12 sheets/min. (letter size) 400 lines/inch Plain Paper
Display	CRT size Display capacity Display mode	15 inches 1.228×964 pixels Reverse/Rotation/Partial enlargment/Scroll
Optical disk memory	Capacity Head access speed	60,000 pages/disk (letter size) 0.5 sec.

Fig. 6 System specifications

Spatial Data Management on the USS Carl Vinson

David Kramlich
Computer Corporation of America
Four Cambridge Center
Cambridge, MA 02142

1. Introduction

The Spatial Data Management System (SDMS) presents information from a database to users by means of graphical representations. SDMS is also capable of displaying information which does not originate from conventional databases. It is uniquely tailored to users and situations where the use of a conventional query language is inappropriate and clumsy. SDMS has been installed on board the USS Carl Vinson, the most recently commissioned nuclear-powered aircraft carrier. This paper will describe the principles underlying SDMS, the system installed aboard the Vinson, problems which arose in the installation and operation of the system, and an evaluation of the system. We will close with a description of future work to be conducted on the Vinson SDMS.

2. SDMS Overview

Spatial data management is a technique for organizing and retrieving data information by representing and positioning it graphically. Data is viewed through a set of three color displays (Figure 1). The displays show flat "data surfaces" on which pictorial representations of the data (icons) are arranged. The left screen presents a scaled view of the current data surface and acts as a navigational aid. The center screen presents a detailed view of a portion of the data surface. A highlighted rectangle on the navigational aid indicates the current position on the data surface. The right screen displays menus for the SDMS subsystems. The SDMS "graphical data space" is the collection of all of the data surfaces, or all the pictures that the user can access. SDMS automatically creates these pictures from data stored by the DBMS. (More detailed descriptions of SDMS can be found in [1] and [2]).

This research was supported by the Office of Naval Research (ONR) of the Department of Defense under Contract No. N00038-81-C-0592, ARPA order 3958. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

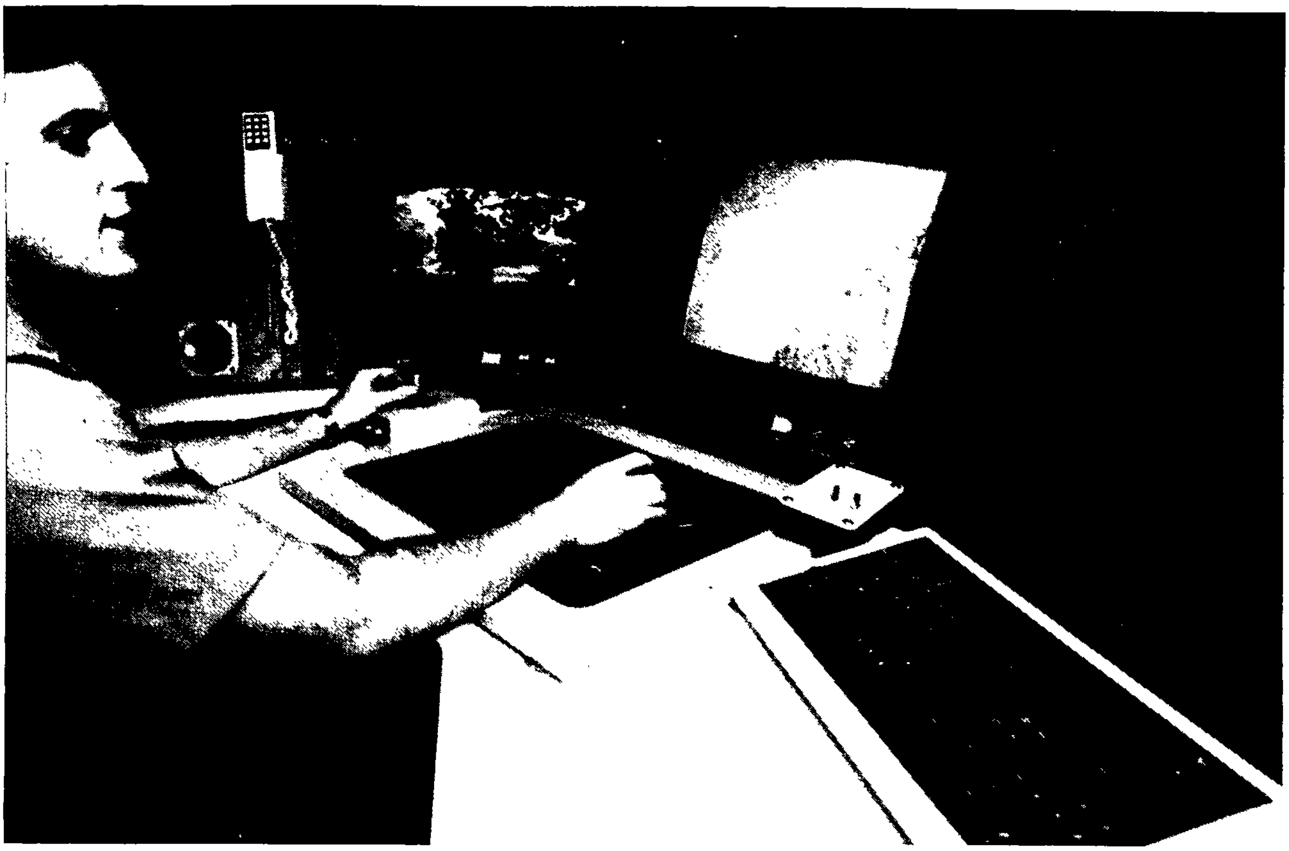


Figure 1. SDMS Workstation
Showing map data surface

The user can traverse the data surfaces or "zoom" into an image to obtain greater detail. The user controls motion about the data surface by means of a 3-axis joystick. This approach permits many types of questions to be answered without requiring the use of a keyboard. A conventional query language is also provided.

Spatial data management is motivated by the needs of a growing community of people who need to access information through a DBMS but are not trained in the use of such systems. A database viewed through SDMS is more accessible and its structure is more apparent than when viewed through a conventional DBMS. Users of conventional DBMSs can access data only by asking questions in a formal query language. In contrast, users of SDMS benefit from the ability to access computer-resident information while retaining a familiar, visual orientation.

By presenting information in a natural, spatial framework, SDMS encourages browsing and requires less prior knowledge of the contents and organization of the database. Thus, a user can find the information he needs without having to specify it precisely or know exactly where in the database it is stored. Users can easily organize, locate, and handle a great deal of information of different types.

SDMS is not restricted to displaying data that originates from a conventional database. SDMS can also present information that originates as text documents, video images, or computer program output.

3. The USS Carl Vinson

The USS Carl Vinson is unique among ships in the US Navy. It is a testbed for advanced information management technology in an operational setting. SDMS is one of several prototype systems installed on the Vinson to explore ways of improving efficiency.

SDMS was installed in the Intelligence Center on the Carl Vinson. The principal task of the Intelligence Center is to keep aware of the deployments of air, surface, and subsurface craft of both hostile and friendly powers. To this end, they rely on intelligence broadcasts of sightings of platforms and their own shipboard sensors - radar, sonar, and reports from AWACS-like aircraft based on the carrier. These reports must be correlated, displayed, and cross-indexed with information about the platforms observed to determine potential threats. SDMS serves as a central repository for the information used in assessing threats. On other ships, the task of analyzing and correlating the incoming information is done almost completely manually and is very labor-intensive.

The next two subsections will describe the particular requirements of the Vinson and how the system was implemented.

3.1 Requirements

The information handled in the Intelligence Center is of many different types. It can be roughly categorized as follows:

1. Real-time data. Position reports of platforms are continuously flowing into the Intelligence Center. Plots must be kept up-to-date.

2. Photographic data. Photos returned by reconnaissance aircraft are matched against file photos to identify platforms.

3. Graphical data. These consist mostly of performance charts and graphs.

4. Static textual and numeric data. These deal with platform characteristics - capabilities, history, armaments, etc. - and are infrequently updated.

The task of SDMS is to present this information in a consistent framework which can be easily understood and used by computer-naive personnel. Figure 2 illustrates the sources of

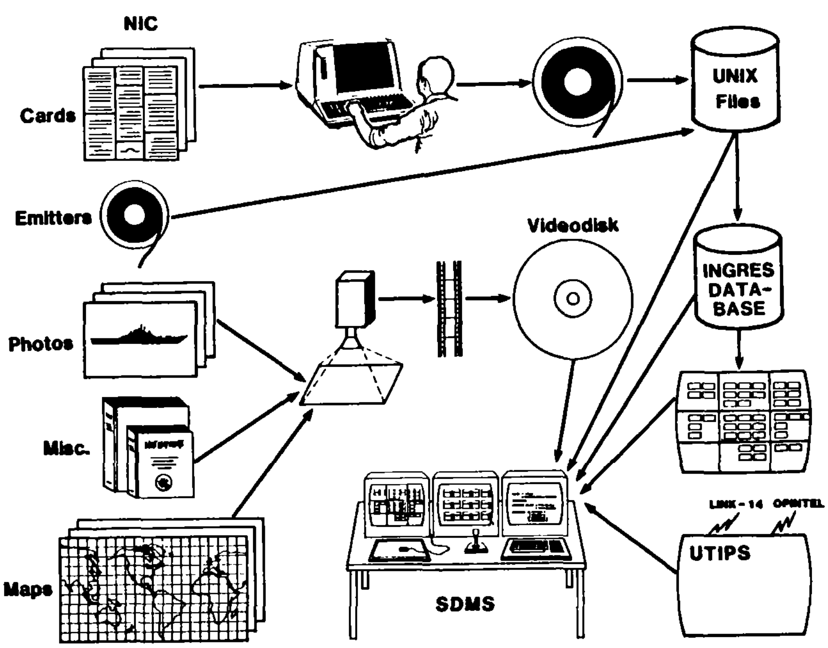


Figure 2. Data Sources

information in SDMS.

3.2 Implementation

The system installed on the Vinson is a modified version of the prototype system developed for the Defense Advanced Research Projects Agency. The basic SDMS system was enhanced in several areas to support the diverse data requirements of the Navy. This section will describe the key subsystems which were added to SDMS. First, however, we should take a quick look at the hardware environment.

The Vinson SDMS runs on a PDP-11/70 under a modified Version 6 Unix. The 11/70 is an older machine architecture with limited address space - using the split Instruction/Data space feature, total logical address space per process is 128Kb. The limited logical address space imposes severe restrictions on the complexity of a program and frequently results in a large task being decomposed into several communicating processes. The modified Unix allowed efficient communication among processes by means of a common block of memory which can be mapped into each process' address space. The database system used for the static data is INGRES from UC Berkeley.

Three major subsystems were added to the basic SDMS system. Each is described in detail below.

3.2.1 Map Display

To present the real-time positional data in the most effective manner, a subsystem was developed which allows ship icons to be overlayed on maps of the world (Figure 3). The maps are stored as photographs on a computer-controlled, random-access videodisk. As position updates arrive in the system, the overlay is updated. Thus the user always sees the most recent view of the data. The user navigates on the maps just as on a conventional data surface, by means of the joystick. Thus the user can scroll around the maps horizontally and vertically and zoom in to see a more detailed view. Because of the discrete nature of the images stored on the videodisk, the scrolling is in (small) discrete steps.

World maps were photographed at a variety of levels of detail and transferred to a videodisk. The maps were photographed in small sections, each section overlapping all of its neighbors. By decreasing the size of the sections, the effect of zooming is achieved (each section covers a progressively smaller portion of the world). Several different world maps at different scales were photographed, resulting in six levels of detail. In addition, navigational charts of the area around Puerto Rico were photographed, allowing the user to zoom in even further in that area.

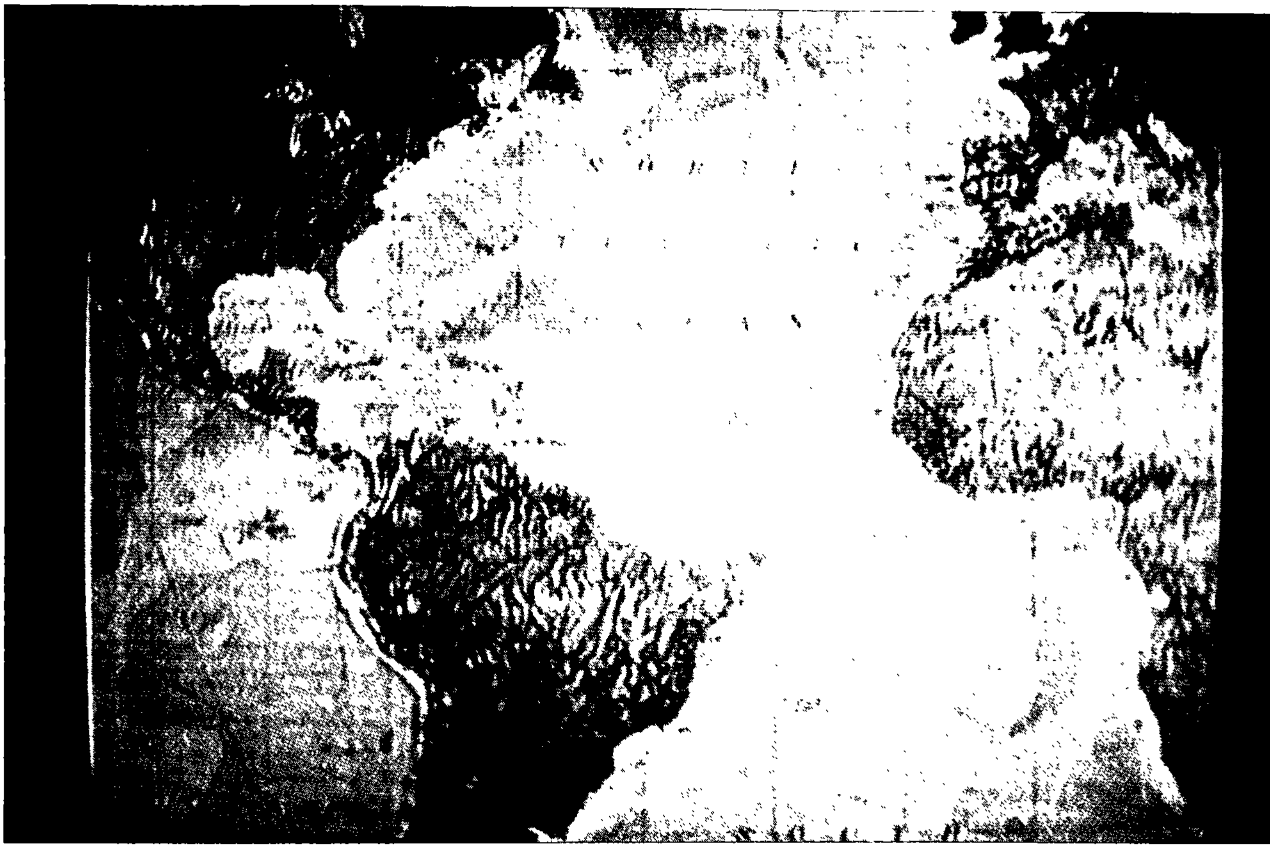


Figure 3. Map Display

The map display system is implemented as a separate program which runs as a subprocess of SDMS through a port facility (described below). In addition to the basic zooming and scrolling functions described above, the map display system provides the user with a menu of display options and a graphical editor for annotating the maps. The user can select from a variety of options controlling the presentation of the position overlays, including display of last deterministic or probabilistic position, toggled display of platform name and velocity vector, and subsetting of platforms displayed. The user can select a platform with a cursor and get a display of its identity and complete position history or call up a detailed display of its characteristics (described below).

3.2.2 UTIPS

SDMS is connected via a medium-speed data link to the Upgraded Tactical Information Processing System (UTIPS). UTIPS is a shipboard computer system that provides track identification and position reports to SDMS. Its function is to interpret and correlate intelligence broadcasts, producing a consistent view of platform positions in spite of frequently conflicting or incomplete reports. UTIPS also includes a predictive capability, providing a probabilistic view of track positions based on past deterministic position reports.

Position updates arrive from UTIPS as frequently as one every three seconds. In addition, as the user scrolls over a map, the position database is repeatedly queried for platforms within the current field of view. Because of the high transaction rate, INGRES on the 11/70 could not be used to store the positional data. Instead a specialized real-time data manager was built which could support the high transaction rate.

The real-time data manager uses a two-dimensional binary tree as an index structure. The latitude-longitude coordinates of a platform are used as search keys. The 2-d binary tree allows efficient updates and retrievals based on the geographic coordinates of the platforms. Basic transactions supported are: insertion of a new record, deletion of an existing record, and an area query which returns all records within the rectangular area specified. Area queries for map projections which are skewed (such as Lambert Conformal) are decomposed into a series of rectangular areas.

SDMS maintains an identical copy of the UTIPS database which can be queried by the UTIPS system for crash recovery.

3.2.3 Display Programs

SDMS supports a facility for embedding computer programs as data types in its data surfaces. These programs are referenced by special icons called ports. Zooming in on a port activates the program associated with that port; SDMS then waits for termination of that program. Two special information display programs were added to the Vinson SDMS to support the display of large amounts of text and graphic data which cannot be easily incorporated into a single icon. One of these programs is the map subsystem described above.

A second program displays detailed information about platforms. Information presented includes photographs, performance characteristics, electronics and weapons carried, history, and order of battle. The user selects a category of information for display from a menu of available categories. Most of the information is textual, but some is graphic. Photographs are stored on the same videodisk as contains the maps and can be displayed on the center screen. Selected weapons have digitally-stored charts of their flight characteristics, called mission profiles, which can also be displayed.

4. Evaluation

The USS Carl Vinson is a testbed for advanced systems technology. SDMS is one of several technology transfer projects on the Vinson. Although the system is an experiment in the use of advanced man-machine interfaces, it is being used operationally by the Intelligence Center.

User acceptance of the system has been quite good. New users quickly learn to use the system. The familiar spatial paradigm is maintained throughout the system, enhancing its ease of use. SDMS centralizes the information that intelligence specialists routinely use in their jobs.

The system has been remarkably trouble-free considering the (computer) hostile environment in which it has been installed. Hardware problems have been chiefly due to the age of the machine, rather than stress (the machine room is immediately below the flight deck and is subjected to strong vibrations while flight operations are being conducted). Personnel on board the Vinson have been trained in both hardware and system maintenance and have successfully applied that training on occasion when problems have developed in the middle of the ocean.

Users have suggested many improvements and new features for the system. As the support contracts have allowed, some of these features have been installed. Many of the suggestions have been for new display modes and options in the map display system. They have included a subsetting mechanism for controlling the display of platforms, display of velocity vectors on the tracks, and position history display.

The system has some limitations which affect its long-term utility. New maps and photographs cannot easily be added to the system because this requires creating a new videodisk. The hardware environment imposes limitations on the complexity of the software and often causes an artificial partitioning of the architecture. Finally, because the extensions to the system are largely independent of SDMS and each other (they are embedded in the SDMS framework), they are not well-integrated into the system as a whole. This makes it very difficult to implement some of the requested features.

5. Future Work

Future work on the Vinson SDMS is divided into two phases: porting the existing system onto new hardware and integrating the special features of the system into a general-purpose graphics frontend. The system will be ported onto a VAX-11/780 installed on the Vinson. It will be functionally identical to the existing system.

The second phase of the enhancements is the integration of the special features of the Vinson SDMS into a general-purpose system. The work involves re-implementing the system to eliminate the ad hoc nature of many of the extensions. The system as currently implemented is a browsing facility for the specialized databases on the Vinson. The new effort will result in a system which will allow the user to formulate an ad hoc query using both the static and dynamic data. The system will include a query menu facility which allows the user to formulate a query by traversing a map of the database schema. A prototype of this system has already been built at CCA [3]. The system will employ knowledge about user context and the query to formulate a data surface which is the most appropriate in the situation. This component of the system will be extensible to allow inclusion of new data types such as still photos and animated sequences stored on videodisk, map backgrounds for track plots, and real-time data. A general interface will be built which will allow communication between SDMS and external analysis tools. The tools will be able to query SDMS' databases and report their results to SDMS for display.

6. Acknowledgements

Many people at CCA contributed to the development of the SDMS system on the Vinson. They are: Jane Barnett, Richard Carling, David Dowd, Mark Friedell, Chris Herot, Martin Moeller, and Ronni Rosenberg. Their hard work and efforts beyond the call of duty made the system a success.

7. References

[1] Herot, Christopher F., Richard Carling, Mark Friedell, and David Kramlich, "A Prototype Spatial Data Management System," _ACM Computer Graphics_, 14, 3, July 1980.

[2] Herot, Christopher F., "Spatial Management of Data," _ACM Transactions on Database Systems_, 5, 4, December 1980.

[3] Friedell, Mark, Jane Barnett, and David Kramlich, "Context-Sensitive Graphic Presentation of Information," Computer Graphics, 16, 3, July 1982.

WRITE-ERROR MANAGEMENT ON WRITE-ONCE DIGITAL OPTICAL STORAGE

Malcolm C. Easton

IBM Research Laboratory
San Jose, California 95139

INTRODUCTION

Several authors ([1],[4]) have suggested the use of write-once optical disk in data base systems. Advantages cited are low cost per bit, large on-line capacity, and availability of a complete and indelible record of all transactions. Capacities of recently announced optical disks are in the range 700 - 2500 Mbytes per surface, with disk diameters 8 to 12 inches; OEM costs per storage unit are currently quoted in the $7000 to $35000 range and are expected to drop [5]. Besides the benefits of compactness and cost savings, the added security of a nonerasable record of all updates makes the new medium appealing and could encourage wider use of data base systems [1].

However, it appears that the obvious first use of write-once disks is for sequential storage of data. Applications requiring writing of records into non-sequential positions on a write-once disk may not be easy to implement on currently available devices. A major source of difficulty is associated with the management of write errors by the control unit of the write-once device or by I/O system software. Because of possible media defects, a data sector (the fixed-length unit of writing) is typically verified immediately after being written. If a criterion for accuracy of recording is not met, then typically the sector is marked as a rejected sector and is rewritten into the next space on the disk [3]. If the last sector of a track is rejected, then the writing continues onto the next track. This scheme will suffice if data are written sequentially on the disk, but is not satisfactory if sectors are to be written in arbitrary order. (The next space may be already occupied!) Moreover, with the cited scheme, the storage capacity of a track is variable and unpredictable. Thus, random retrieval of sequentially-written data will require use of a table to map the relationship between sector number and physical address.

To facilitate data base applications on write-once disk, it appears that provision should be made for spare space within each track and also for overflow space if the spare space proves inadequate. Similar provisions have long been available, at least for off-line allocation, on high-end magnetic disks. Since write-once disks cannot be pre-written for testing, the need for spare space allocation appears even more important in this new context. Ideally, the user program would not be aware of alternate space assignments. The control unit or system I/O software should present an interface through which the disk appears to have a continuous address space. If, for example, write errors exhaust the spare space in a track, then there should be an automatic action taken so that the program can continue unaffected.

With today's magnetic disks, allocation of spare space typically requires software intervention. Before being put into service, every track is initially checked; spare space is assigned where necessary. If permanent write errors occur in use, however, then the program that encounters these errors will abort unless it has its own capability for intercepting error conditions and assigning spare tracks. If the run-time software does not provide the alternate space assignment, then the disk is detached from users and a utility program is run to effect the new allocation. After an aborted run using erasable storage, it is generally possible to restore the disk to its former state (in all but the defective areas) and rerun the job. This inconvenience is tolerable because hard write errors are uncommon with today's magnetic storage devices.

Write-once disks, however, have higher raw bit error rates and untested capability for writing. Also, the former state cannot be restored if a job aborts; space written cannot be recovered. Therefore, it is essential that assignment of spare sectors be done while a user program is running. We briefly consider an application, then devote the remainder of this paper to some issues involved in providing this capability.

AN APPLICATION - A DIRECTORY STRUCTURE BASED ON HASHING

As a sample application, consider management of a directory to the contents of a disk. Insert, delete, update and retrieval operations, using the file name as a key, are required. The solution suggested here is suitable if the majority of directory changes are new entries rather than updates of existing entries.

Consider a simple variation on hash tables. The file name is hashed into a track address. The insert/update operation writes the new directory entry in the first free sector of the track. The delete operation writes another new entry containing a "delete flag." The current entry is defined to be the last entry with the selected file name. Thus an entry with the delete flag supersedes all previous entries for that file.

Assume the track holds N sectors. Then after N-1 entries have been written, a pointer to a continuation track is written. Thus each "hash bucket" becomes a chain of tracks. In a multi-track bucket, a new entry is always written into the first free sector of the bucket.

On retrieval, a name is hashed into a track address. The entire bucket is read. The last entry holding the specified name is the current entry. To minimize the number of seek operations, the initial buckets can be set up as multi-track buckets.

The simplicity of this approach depends on the assumption that there are a known number of sectors per (logical) track. Thus there is no doubt that the continuation pointer can be written at the end of the track. Without this assurance, the use of chaining appears far less attractive. One might, for example, allocate the continuation track in advance, writing its pointer into the first sector of the initial track. But what if a track is unable to hold even one good sector? Solutions can be devised for this and other special cases, but a general approach to management of write errors is preferable.

DISK DESIGN FEATURES

There is a significant range in the cost and complexity of the devices currently available or announced. Any listing provided here would rapidly become obsolete. We note, however, at least one product description in which spare sectors and tracks are explicitly mentioned. Hitachi's OD301 optical disk drive unit and OF301 optical formatter controller are to provide 62 sectors plus 2 spare sectors per track, as well as 41300 data tracks with 128 spare tracks [2]. The way these spare sectors and tracks will be used is not described.

For brevity, we consider here only the case of a disk of fixed-size sectors, where sectors may be written by the user in any order. (This is sometimes called a hard-sector disk.) The error correcting codes along with the criteria for rejecting a sector and calling for a rewrite vary significantly among products. One extreme is to reject a sector having any write error; the opposite is to accept any sector whose data can be corrected by the ECC. In the first case, the rate of rejecting sectors could be as high as 5 - 10 %, while in the second case the rate would be less than 1 %.

ALLOCATION OF SPARE SECTORS AND TRACKS

Our goal is for the control unit to present to the application program the appearance of a continuous address space. An alternative is the use of I/O software, such as an access method, that would provide this function. In either case, we propose a straightforward mapping of the logical address space into the physical address space, with provision for exceptions caused by errors. Thus each physical track, through possible use of spare space on the track, is expected to hold N logical sectors.

A physical track that cannot hold N good sectors is deficient. For such tracks, an overflow area is provided. Methods of overflow handling are discussed in a later section. First we consider the rate of occurrence of deficient tracks.

As an example, consider a case where the sector reject rate is 10%, and a track holds 50 sectors. Let K of the sectors be set aside as spares. If we assume that the rejected sectors occur at random and independently, then we can readily compute for each value of K the number of tracks that will not be able to hold 50-K good sectors. With K = 5, 38% of the tracks will be deficient. With K = 8, this is reduced to 6%. With K = 10, this is reduced to 1 %. Deficient tracks require an additional indirection to an overflow area that can hold the additional sectors. Extensive use of such indirection will hurt performance, and so it would seem reasonable to choose K in the range 8- 10. Therefore, it appears that about 20% of each track must be set aside for spare space. One way to reduce this amount would be to group a certain number of tracks together into a "region" and to provide all the spares for a region in one place. The capability of some devices to do a fast seek to neighboring tracks would be useful with this approach.

Suppose instead that the expected number of rejected sectors per track is much less than one. Then it is sufficient to omit spare sectors on tracks and to provide an overflow area for the few cases of defective tracks.

MANAGEMENT OF ALTERNATE SPACE

An important issue is how to find, on readback, the alternate location of a data sector that failed verification and was rewritten. A related question is how to distinguish between a sector that originally failed verification and one that was previously valid but currently causes a read error. On magnetic disks, information stored in the initial record of a track and/or in the record headers is used to find the alternate location of the data in case of defect skipping.

On write-once disk, one approach is to write a pointer following each bad sector to redirect retrieval to a new location. This pointer would consume space, however, and thus further reduce track capacity. Moreover, another error might occur while writing the pointer. Alternatively, one can mark or write over a sector that fails verification in a way that is unmistakable on readback. (This technique is used by Philips, but apparently only to redirect to the next sector in physical sequence [3].) It is easy to extend the marking method to permit the sector to be rewritten in the spare sector area of the track. The original sector address is included in each rewritten sector; this increases the storage space per sector by a few bytes. The control unit or I/O software, on reading a reject mark, searches the spare area and identifies the rewritten sector by its sector address. The case of track overflow is discussed below.

OVERFLOW AREA

Suppose that a track is unable to hold its nominal number of good sectors. The track's spare sectors, if any, have been consumed. There are two obvious ways to use the overflow area.

Method 1 copies all the valid sectors from the deficient track to an empty spare track. We have already argued that the system design should keep the number of deficient tracks to a small percentage of the total number of tracks. Therefore, space consumption should not be a major issue here. However, the time to carry out the recopying, at least two disk revolutions, is an obstacle. An advantage of the approach is that all sectors that logically belong to a certain track are stored on one physical track. This is beneficial because data management algorithms for write-once disk typically are concerned with the entire contents of a track ([4],[1]).

Method 2 writes into the overflow area only the sector or sectors that could not fit in the deficient track.

Random access to the contents of a track having a single overflow sector will generally require two more seeks with Method 2 than with Method 1, provided that the alternate track redirection is controlled in either case by a table stored in RAM. Thus Method 1 appears preferable for data base applications.

With either method, some thought must be given to keeping a list of pointers to the overflow area on the write-once disk itself, and to what happens if further write errors occur in the overflow area or in keeping the list. A complete treatment of these problems is beyond the scope of this paper, but the solutions appear straightforward.

CONCLUSIONS

The write-once optical disk shows considerable promise for use in data base applications, but not all currently available devices appear suited to such uses. The limitation is more likely to be found in the control unit than in the drive. Methods of spare space management, such as those discussed here, can be added to the control unit logic. With suitable interfaces, the system's I/O software can provide the same capabilities. The goal of these management methods is to provide the appearance of a continuous address space, with an underlying physical storage space as close to continuous as is practical. Moreover, the capability for non-sequential writing is preserved. These features should significantly improve the value of write-once devices in data base applications.

REFERENCES

[1] Copeland, G. What if mass storage were free. Computer (July 1982), pp. 27-35.

[2] Hitachi Ltd. OD301 Optical disk drive unit. OF301 optical formatter controller. Product specification. (Preliminary) (Nov. 1983).

[3] Kenney, G. C.,et. al. An optical disk replaces 25 mag tapes. IEEE Spectrum (Feb. 1979), pp. 33-38.

[4] Maier, D. Using write-once memory for database storage Proc. ACM Symposium on Principles of Data Base Systems, 29-31 March 1982. pp. 239-246.

[5] Ohr, S. Optical disks launching gigabyte data storage. Electronic Design, August 18,1983. pp. 137 - 146.

Initial Experience with Multimedia Documents[1] in Diamond[2]

Harry C. Forsdick, Robert H. Thomas,
George G. Robertson and Virginia M. Travers

Bolt Beranek and Newman, Inc.

Multimedia documents are collections of text, graphics, images, voice and other computer originated data presented on a single display surface such as a piece of paper or a computer display. This paper describes experience gained in the design and implementation of the multimedia document model used in the initial implementation of the Diamond multimedia system. Three different document models are described and compared.

1. Introduction

Diamond is computer-based system for creating, editing, transmitting, and printing multimedia documents[3]. A Diamond document may contain text, graphics, images and speech as well as other types of objects such as electonic spread-sheets[4]. For example, a map in the form of a drawing or image can be combined with directions described in text or by voice or both into a single Diamond document. Diamond documents can be used for a variety of purposes including messages, memos, notes, and forms.

This paper describes three models for multimedia documents that were explored during the development of the initial implementation of Diamond. Each of the models is based on the premise that a multimedia document is a structured composition of objects of possibly different media types to be presented in a coordinated way. The

---

[1]This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) under Contract No. F30602-81-C-0256 which is monitored by the Rome Air Development Center (RADC). Views and conclusions contained in this report are the authors' and should not be interpreted as representing the official opinion or policy of DARPA, the U.S. Government, or any agency connected with them. This paper has been approved for public release and distribution is unlimited.

[2]This paper is an updated version of a paper of the same name which appears in the Proceedings of the IFIP 6.5 Working Conference, May 1984 [5]

[3]Throughout this paper we speak of Diamond as a system which handles "documents" rather than as a system which only handles "messages". In our view, a message is a document that has been sent from one user to another. The more general term, "document", has been chosen because only part of what a modern message system does is concerned with message transmission; much is concerned with the preparation, storage, management and processing of documents which may or may not be sent as messages.

[4]Rows and columns of interrelated numeric data of the sort manipulated by programs such as VisiCalc [1].

three models are:

1. An Experimental model. This model was developed to experiment with ideas about how different types of media might be combined into a single document.

2. The evolving DARPA Internet model [6]. This model is being developed by several groups in the DARPA research community investigating the problem of transmitting multimedia documents between dissimilar computer systems

3. The model supported by the initial implementation of Diamond. The Diamond model is based on experience with both the Experimental model and the DARPA Internet model, and in that sense represents improvements to both. We expect the document model used in Diamond to evolve as experience is gained with its use.

In order to provide a context for discussing these models, we first briefly describe Diamond. A more detailed, though somewhat dated, description of the Diamond system is presented in a design document [4], and a paper describing Diamond is in preparation.

Several considerations beyond the goal of supporting multimedia documents have influenced the Diamond design including:

o Diamond should provide a responsive, easy to use and helpful user interface.

  Because a substantial amount of computing power must be dedicated to each user in order to provide an interface with adequate interactive responsiveness, the primary user access to Diamond should be through very powerful single user workstation computers[5].

o Diamond should be built upon a distributed architecture.

  This is, in part, a consequence of using single user computers for user access points. A Diamond configuration includes single user access point workstations and a collection of shared computers which support the workstations by providing services. such as message delivery and long term document storage. Two major benefits of this type of architecture for Diamond are that it can be expanded incrementally to support a growing user community and it can be structured to provide services in a highly reliable fashion by replicating key hardware. software and database elements.

o Diamond should be able to accommodate a wide variety of types of user access points and user interfaces.

  Not every Diamond user will have a workstation and those that do may need

[5]A workstation in this class includes a powerful processor, high resolution graphics, a substantial amount of main memory (1-2 MByte), an interface to a high performance local area network, and possibly secondary storage, and it would be configured with a graphical pointing device and voice I/O equipment.

to access Diamond when they are away from their workstations. Users who access Diamond from points that do not have the full complement of devices required to support all of the media will not be able to exercise Diamond's full multimedia capabilties. However, they should be able to deal with the parts of documents their access point is equipped to handle. In addition, different users may prefer very different styles of interacting with Diamond. To satisfy these users, it is important that the system be able to deal with different user interfaces.

o Diamond should operate to ensure the security and privacy of user messages and documents.

Users will create documents that contain sensitive information and will rely upon Diamond to protect them from unauthorized disclosure.

o Diamond should operate in an internetwork environment.

Like earlier generation text-only systems, Diamond will operate is an environment that includes many interconnected computer networks and a variety of other message systems. The Department of Defense Internetwork is an example of such an environment. A Diamond system should be able to interoperate with other Diamond systems as well as with other multimedia and text-only message systems.

o The Diamond implementation should be portable.

If it is successful, Diamond will outlive the hardware base used initially to support it, and as newer more powerful hardware becomes available there will be a desire to run Diamond on it. Consequently, it is important that the implementation be relatively easy to transport from one hardware base to another.

For the initial version of Diamond we have chosen to focus on the following four areas:

o The ability to handle multimedia documents.

o The use of a distributed architecture to support Diamond.

o The use of powerful single user workstation computers as Diamond user access points.

o Portability of the Diamond implementation

While important, the other considerations have not been the primary initial focus of the Diamond effort. In particular, advancing the state-of-the-art in message processing systems, except as required to handle multimedia messages, has not been a primary goal.

Diamond is implemented as a distributed system. Documents and folders, which hold collections of documents and other folders, are stored in a distributed database. Information about users, such as authentication information, the identity of their "inbox" folder and usage preferences, is maintained in a registry database managed by

Diamond. Users access Diamond through user interface components. The user interface components, typically run on powerful single user workstations and interact with other distributed components of Diamond to make the services they provide accessible to users.

The development of Diamond was undertaken as part of a research project in the areas of multimedia and distributed systems. From the outset, a primary project objective was to produce a "real" system targetted for a user community other than the system developers. There were several reasons for this. With others, we hold the view that the best way to establish the validity of new ideas and approaches for computer systems is through working systems that embody them. One of the project objectives is to understand how a capability for multimedia changes the way computer-based person-to-person communication systems are used, and the extent to which the multimedia capability improves the quality and effectiveness of such systems. A widely used system is required to begin to answer these questions. In addition, we felt that feedback from users would serve to focus the research, particularly in the multimedia area, on "real" problems.

The objective of developing an operational system has, at times, limited the extent to which promising ideas could be explored. For example, in order to ensure Diamond was a complete and useable system, the development of new mechanisms for handling various media types and for improving Diamond's performance and survivability characteristics as a distributed system had to be postponed until capabilities considered essential in modern message systems, such as "reply" and "forward" operations, were provided.

An initial implementation of Diamond is operational, and work is progressing to enhance it in a variety of areas.

2. Experimental Document Model

The Experimental document model and the software that supports it was developed as a vehicle for exploring techniques for combining different types of objects into a single integrated document. The software evolved to a level that permitted the construction and transmission of multimedia documents as messages. This facility proved to be useful both as the experimental vehicle it was intended to be and as a means for demonstrating the concept of multimedia mail. However, it was never intended to be an operational system for managing multimedia documents. Experience with the experimental facility suggested a number of extensions, both to the model and the supporting software, that should be incorporated into an operational multimedia system.

Figure 1 is an example of the type of document that can be composed using this model. For this model, the position of an object is specified by its (pixel) coordinates in a quarter plane. Every object has a width and a height, which are expressed in pixels. The underlying structure of the document in Figure 1 is presented in Figure 2. The types of objects that may be included in such a document are.

- **Text**: A line of text.

  Each line is a separate object which may be independently positioned. During text entry, the initial position of the pointing device (e.g., a mouse or trackball) determines the left margin. The right margin is determined by the width of the display window in which the document is being composed. New text objects (i.e., single lines of text) are created when the right margin is reached. The notion of collections of text objects grouped together, for example as in paragraphs, is not directly supported by the model, although the visual effect of such grouping can be achieved by carefully positioning the text objects.

- **Scanned image**: A picture or drawing which has been digitized on a facsimile scanner.

- **Voice**: A spoken passage encoded by a vocoder

  Since voice objects cannot be displayed directly, an icon and a caption are used to indicate the presence of voice. The voice can be played back by pointing to the icon that represents it and invoking the "MoreDetail" operation.

- **General Object**: An object that is a collection of data, whose presentation is performed by a companion program corresponding to the type of data.

  General objects represent a mechanism for extending the types of objects that may be included in multimedia documents. A general object is represented by a caption that indicates the presence of the object. For example, in Figure 1 the caption "Select to show. Space Shuttle Analysis" indicates a general object.

  Figure 3 illustrates the manner in which general objects are displayed. When the user requests "MoreDetail" while pointing at a general object, in this case, the object labeled "Select to show: Space Shuttle Analysis", the program for presenting the object is run to display the object within a newly created window. In general, the new window overlaps the window used to display the document itself. Examples of data types handled as general objects within the Experimental model and that can be included in documents include electronic spread-sheets, graphical line drawings, and graphs generated from tabular data.

One of the benefits of the general object notion is that the document system does not need to know the details of how a general object object is represented. It simply must know the type of the object and the name of a program used to present objects of that type. A difficulty with this approach as implemented by the software

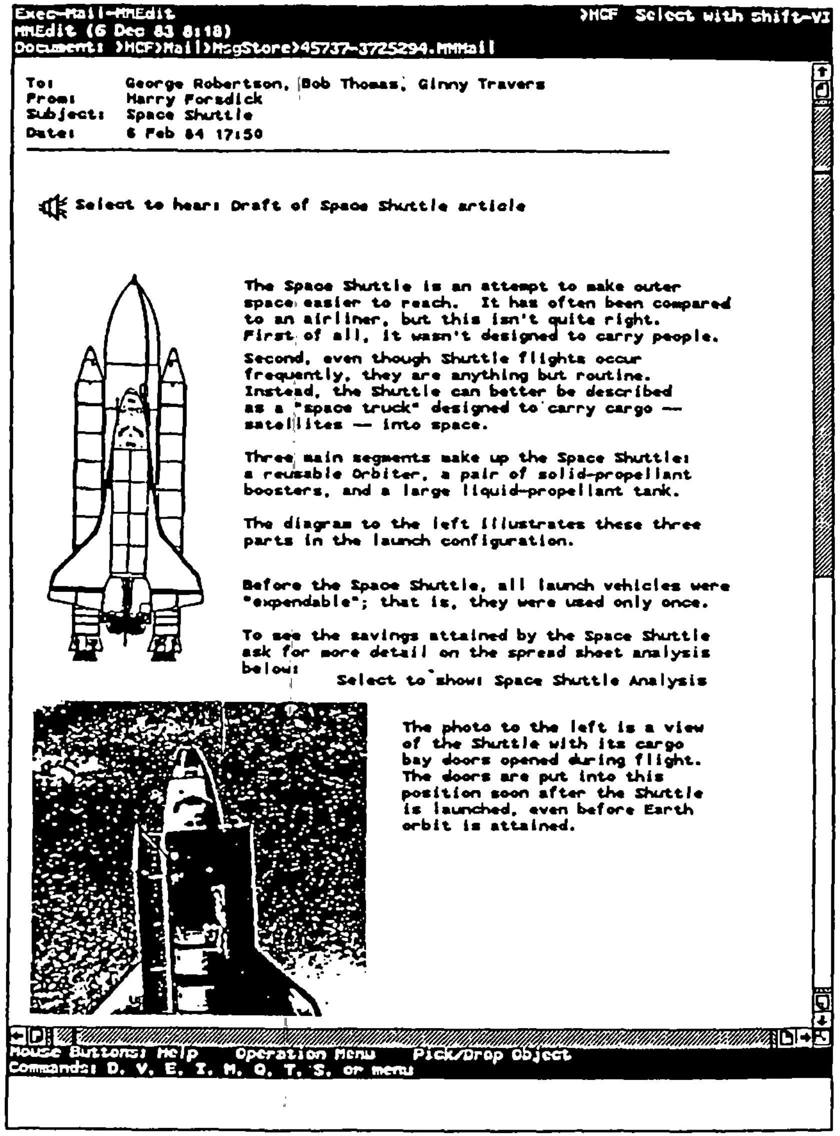


Figure 1. Document represented in Experimental Model.

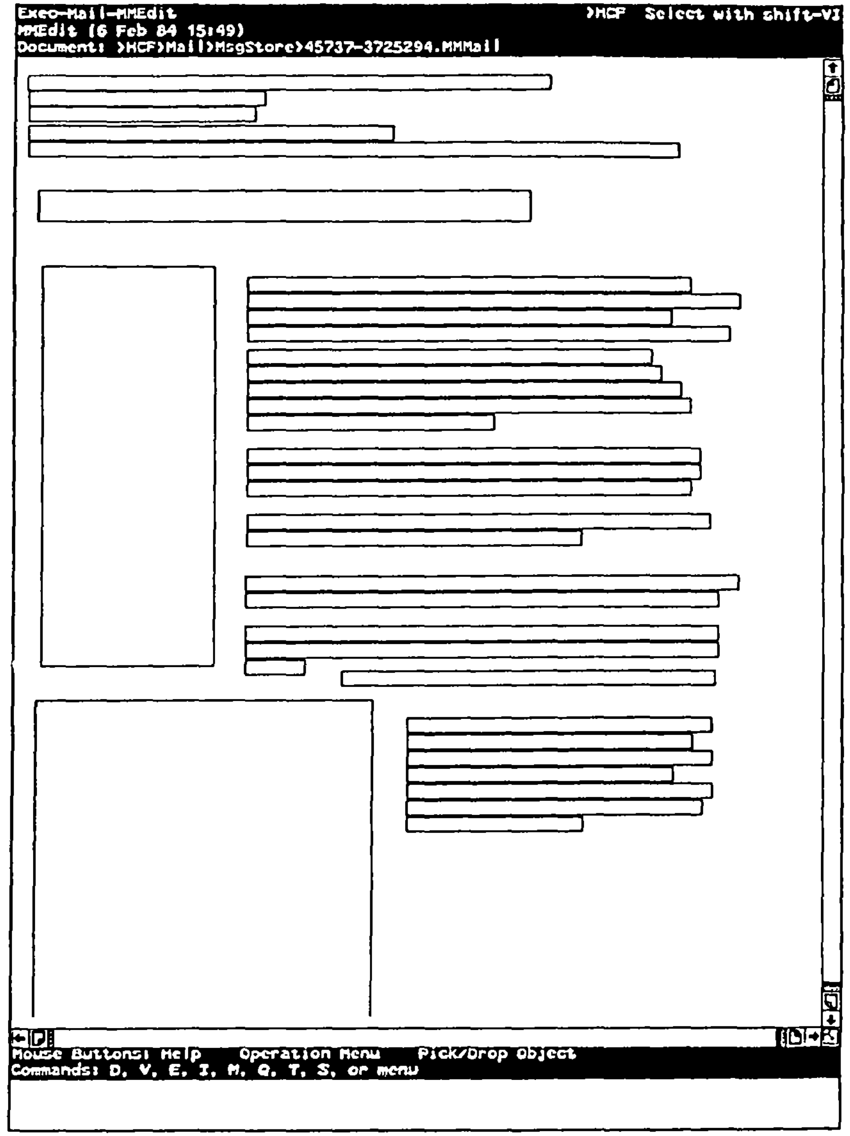


Figure 2. Underlying structure of the document in Figure 1.

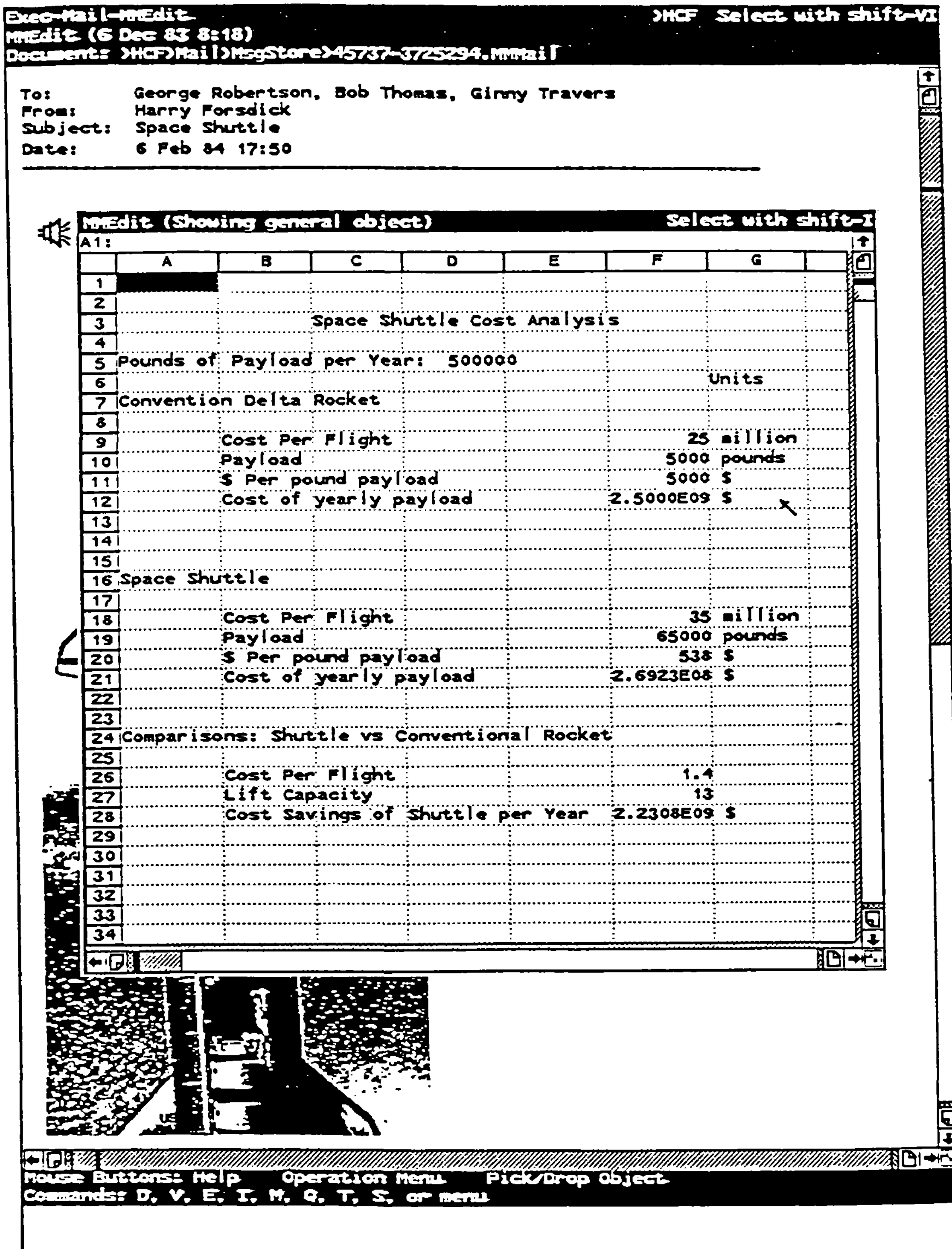


Figure 3. The presentation of a spreadsheet general object.

supporting the Experimental model is that the visual integrity of the document is destroyed by having to display general objects in separate windows. Consider, for example, a document that contains an electronic spread-sheet and an explanation, in text, of the spread-sheet; the explanation (a text object) cannot be viewed simultaneously with the spread-sheet itself (a general object).

A major flaw of the Experimental model of multimedia documents is that relatively little information about the objects that make up a document and the interrelationships among them is maintained. This manifests itself in several ways. For text, it is difficult to change or reformat blocks of text once they have been entered into the document. After text is entered there is no provision for grouping lines into blocks which may be subsequently formatted or edited. For images, the manifestation is somewhat different. A variety of editing operations are provided including cropping, rotating, and scaling (reducing and enlarging). Some of these operations are information lossy in the sense that when an image object is reduced and then enlarged by the same scale factor the result is loss of resolution. Another problem is there is no convenient mechanism for controlling the overlapping or grouping of objects and so, unusual and unpredictable interactions between objects on the display can occur.

The deficiencies of the Experimental model and its implementation are most evident by the difficulties of editing partially completed documents. However, as simple as it is, complex and sophisticated documents can be expressed using this model.

3. DARPA Internet Model

The DARPA Internet model [6] is the basis for a standard for representing multimedia documents in a machine independent manner for purposes of exchange among machines and document systems of possibly dissimilar architecture. In this model, objects, called "Presentation Elements", are organized hierarchically into a single composite document. The types of objects currently supported by the model include Text, Scanned Images, and Voice although there are plans for adding several additional types including Graphical Line Drawings.

To preserve the machine and device independence in the representation standard, certain attributes of documents are abstracted from their concrete (usually machine dependent) representations. As a result, the specifications for these attributes tend to be qualitative or relative in nature as opposed to quantitative or absolute. For example, the positions of the objects that comprise a document are not specified explicitly. Instead, the objects are organized into groups, and the presentation of objects within a group is specified as being "sequential", "simultaneous" or "independent". The interpretation of these descriptions is not precisely specified by the Internet standard in order to permit a wide variety of implementations. A possible implementation of "sequential" is to divide the display surface into horizontal bands and to present the objects in sequence, one per band. A possible implementation of "simultaneous" is to present the objects side-by-side within one horizontal band. Figure 4 is an example of a document that could be encoded in this model. The underlying structure of the document in Figure 4 for the DARPA Internet Model is shown in Figure 5.

Work to refine this model [3] has produced conventions for expressing common formatting styles for text, such as paragraph, enumeration, and itemization as well as for unformatted text (i.e., formatted explicitly by the user).

4. Diamond Multimedia Documents

The model of multimedia documents used in the initial version of Diamond is based on experience with the Experimental and DARPA Internet models. The major differences between the Diamond model and the other two are:

- o There are several alternative means of specifying positions of objects in a document including absolute and relative positioning.
- o Graphical drawings and electronic spreadsheets are supported as an explicit object type rather than through extension mechanisms (such as the general object mechanism supported by the experimental model).
- o Color is supported in a general way for all object types.

This model represents an improvement over the Experimental model and an extension of the ideas in the current DARPA Internet model.

In practice, the expressive power that can be attained by a document model is largely determined by the software (e.g., the editor) used to produce documents according to the model. The current document editor for Diamond limits certain degrees of freedom possible in the model. It also automatically performs certain operations not addressed in the model in order to achieve a balance between expressive power and simplicity. These restrictions and additions include:

- o Different objects cannot overlap. This makes it easier for users to edit documents. Since objects don't overlap, when a user points to an object as part of an editing operation, there can be no ambiguity concerning the object to be manipulated. Interrelations between objects can be represented explicitly by a linking connection (represented by an arrow) from a feature in one object to a feature in another object.
- o All objects that have a meaningful visual representation are displayed

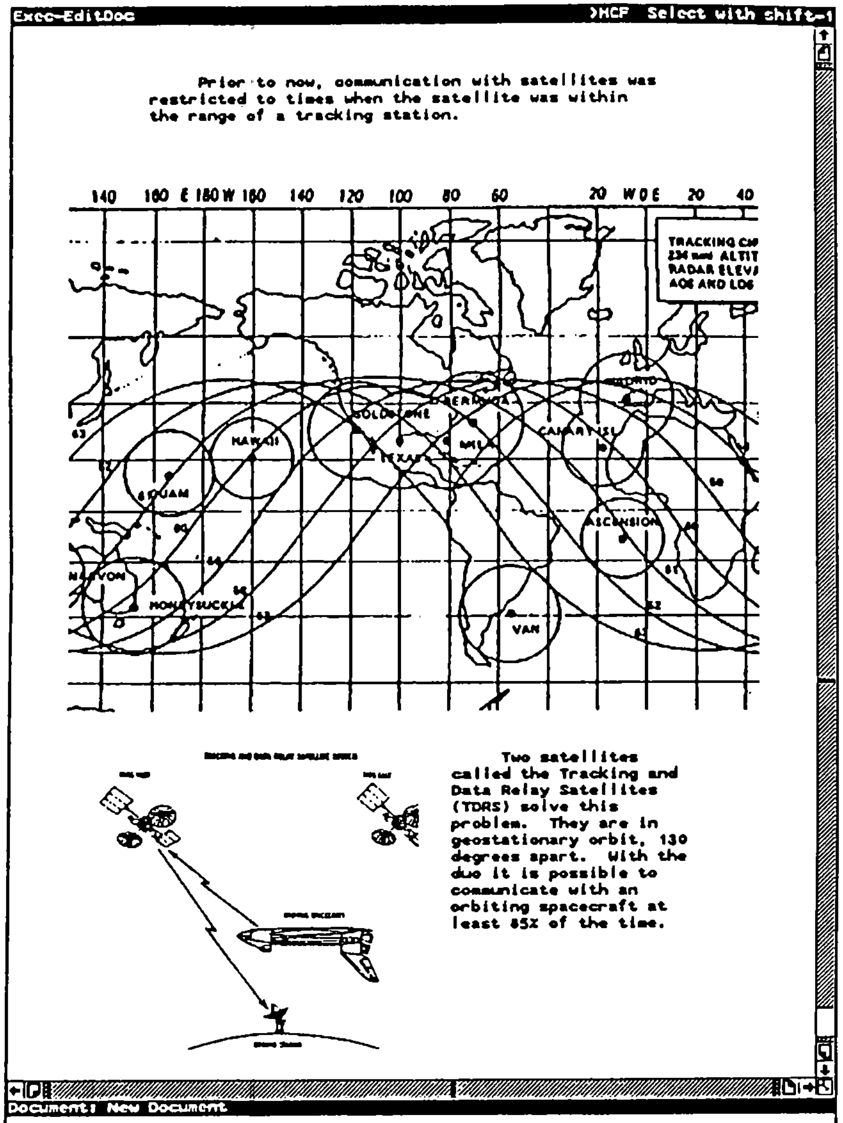


Figure 4. A document expressed in the DARPA Internet Model.

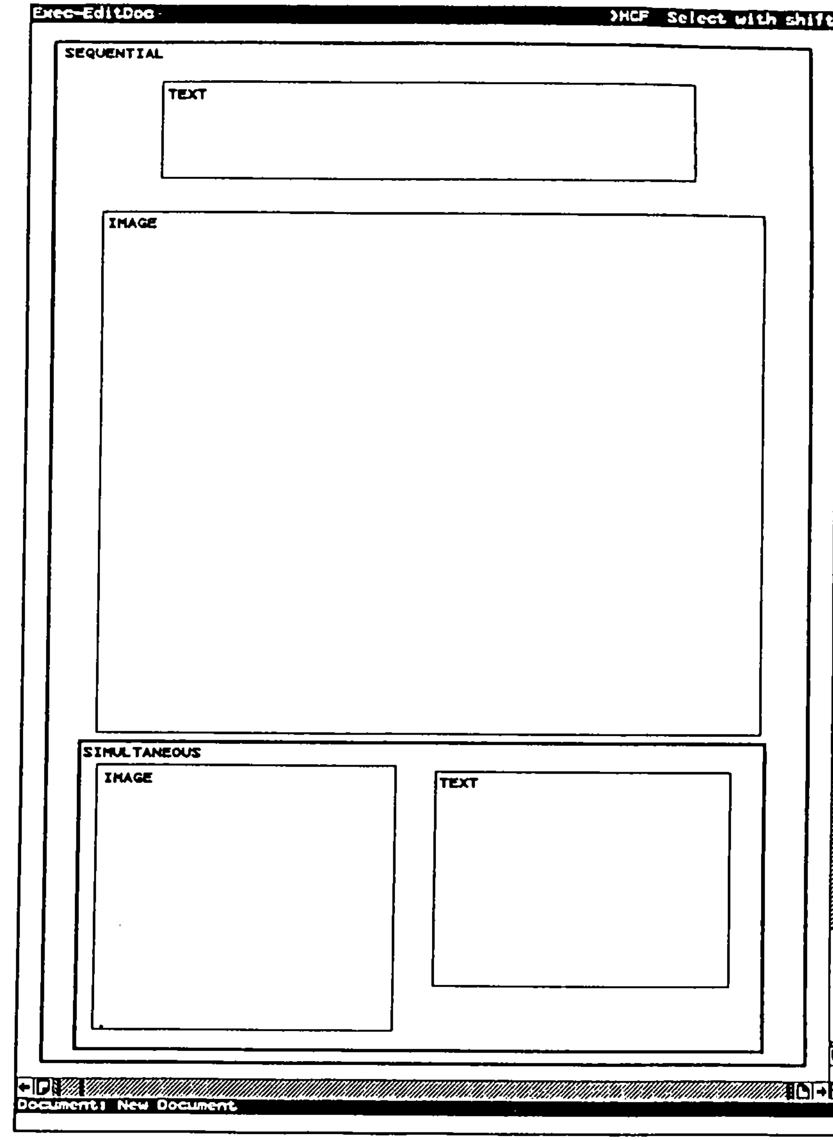


Figure 5. Underlying structure of the document presented in Figure 4.

directly on the display surface. The distraction and confusion that is caused when some objects are directly displayed and others require additional user action to be seen (as was the case with the Experimental Model) has been eliminated.

- Automatic formatting in several different styles is provided for text objects.

- Assistance is provided for automatically formatting a composite multimedia document by adjusting the positions of objects so that they conform to standard margins and pre-determined inter-object spacings.

The style of a Diamond document (see Figure 7) is similar to documents that appear as published books and journal articles, with a few significant differences. These differences include:

- **Voice**. Books and journals have no means of incorporating voice. When displaying documents that contain voice, Diamond represents the voice passages by icons on the display screen, and provides means to playback the vocal passages.

- **Annotations**: Footnotes provide a means for an author to annotate a book or journal article. With formal publications however, there are no convenient means for readers to share their comments about a document with each other and with the author. Shared annotations to a document are feasible with electronic documents. Diamond supports annotations by allowing users to "attach" comments (which are themselves documents) to a document. In the initial version of Diamond annotations are represented by icons displayed in the margin of the original document. The user must explicitly request to see the contents of an annotation by pointing to its icon and requesting "MoreDetail".

- **Document Layout**: Document formatting in publishing is a fine art involving a large amount of human judgement in the way the parts of a document are laid out. The initial version of Diamond does not automatically generate document layouts of the sophistication that a graphic artist can produce. However, Diamond provides means for users to control the layout of documents.

- **Resolution of Displays and Printers**. The devices currently used in Diamond for displaying and printing documents have relatively low resolution (in terms of the quality of lines, characters and images they can represent) compared to the devices used for high quality published material

A Diamond document is a collection of objects which may be represented either directly (e.g., text, images, graphics) or indirectly by means of icons (e.g., non-visual objects such as voice) on a two-dimensional surface such, as a display device or a piece of paper. The types of objects that may appear in a Diamond document include.

- **Text**: ASCII text passages similar to the contents of current electronic text messages [2]. Diamond supports multiple text fonts, and a variety of styles of formatted text, including paragraphs, itemization (indented, marked lists of points), enumeration (indented, numbered lists of points), and verbatim (as entered by the user).

- **Graphics**: Drawings including lines, geometric figures (rectangles, circles, ellipses, etc.), and text strings. Closed regions may be shaded with arbitrary textures (regular bit patterns that fill the regions). A macro capability is supported which permits groups of objects to be treated as a single object.

- **Images**. Digitized images of drawings, maps, photographs, and other pictures. Images may be represented in black and white as well as shades of gray and color. Although it is possible to send text and graphics as image data, because conversion of text or graphics to image form generally results in loss of information and expansion of data, image data is most suitable for visual information that cannot be represented in any of the other forms.

- **Voice**. Voice passages encoded by a vocoder. The most natural use of vocal objects in a document is probably as a comment or as an annotation to other objects in the message. However, because Diamond places no restrictions on the use of voice in documents, the major information content of a document may be one or more voice objects. Currently, Diamond uses LPC algorithms [7] for vocoding.

- **SpreadSheets and Charts**: Electronic spreadsheets and charts of selected spreadsheet data. The underlying spreadsheet model is stored along with the spreadsheet data in the document. This permits recipients of Diamond messages that include spreadsheets to interact with the underlying models if they choose.

- **Connections**: Linkages, represented by lines with arrows, that connect a point within one object to a point within another. For example, it is possible to compose a comment about a small feature of an image using speech and have a line drawn from the comment to point at the feature.

The Diamond software and the internal representation used for documents are designed to permit introduction of new media types.

In Diamond information about the individual parts of a document (paragraphs, captions, labeled fields, line drawings, images, vocal passages, etc.) is preserved in the document. This structural information facilitates the standard presentation of documents (e.g. on different types and sizes of display surfaces) as well as editing documents that are evolving. The underlying structure of the document in Figure 7 is presented in Figure 8.

For composition, editing and viewing purposes, a Diamond document is organized as a set of non-overlapping boxes. Each box can contain source material of a given media type, plus connections that relate points in one box to points in other boxes. The boxes are positioned on a quarter plane. Although it need not be, the width of a document is usually bounded by the width of standard sized paper to conform with the "paper world". A box is specified by its width and height as well as its position[6]

---

[6]Positions and distances are expressed in pixels, and every document contains the resolution of the environment in which it was created in pixels per inch.

The positions of boxes can be specified in absolute or relative terms. With absolute positioning, the coordinates of the upper left corner of the box are described. For relative positioning, the relationship between box A and box B is described in terms of sets of positioning descriptors such as Above, Below, Right Of, Left Of, Top Aligned, Bottom Aligned, Centered On, etc. Relative positioning is used to facilitate formatting a document so that it can be shown on display surfaces with different shapes and sizes, this is useful since most modern window display systems support windows of various shapes and sizes. For example, Figure 6 shows how a document whose object positions are specified in relative terms would be displayed in two different shaped windows. The relative sizes of objects whose presentation is flexible (such as text or graphics that can be scaled) can be adjusted so that the presentation of the composite document is roughly the same, regardless of the shape of the display surface.

Boxes are also used to group objects into collections of source material of diverse media types[7]. The purpose of grouping the contents of a document in boxes is to help the author distinguish one part of the document from another and to bind several distinct objects together so that they can be treated as a single object for purposes of positioning and editing. The reader sees minimal evidence of the box structure of a document. For example, in Figure 7, boxes appear in the document, primarily for the visual appeal of showing the reader the boundaries of the images objects. Figure 9 shows the view seen when editing the same document. In this case, most of the boxes are shown, although an attempt is made to minimize the clutter and confusion by showing only the most relevant ones. For example, in this document there are three text boxes grouped together (paragraph, enumeration, paragraph) although only the outer grouping box and the inner text enumeration box are shown explicitly. Displaying the enumeration box shows its extent and helps the author position the cursor if for example, an additional point is to be added.

5. Conclusion

This paper has described three models for multimedia documents that were investigated during the initial development of the Diamond multimedia document system. The Experimental model is adequate for expressing the appearance of multimedia documents laid out on a fixed size display surface. It[8] does not maintain sufficient

---

[7] In this special case, the enclosing box completely overlaps the boxes it groups together.

[8] More precisely, the software that implements it.

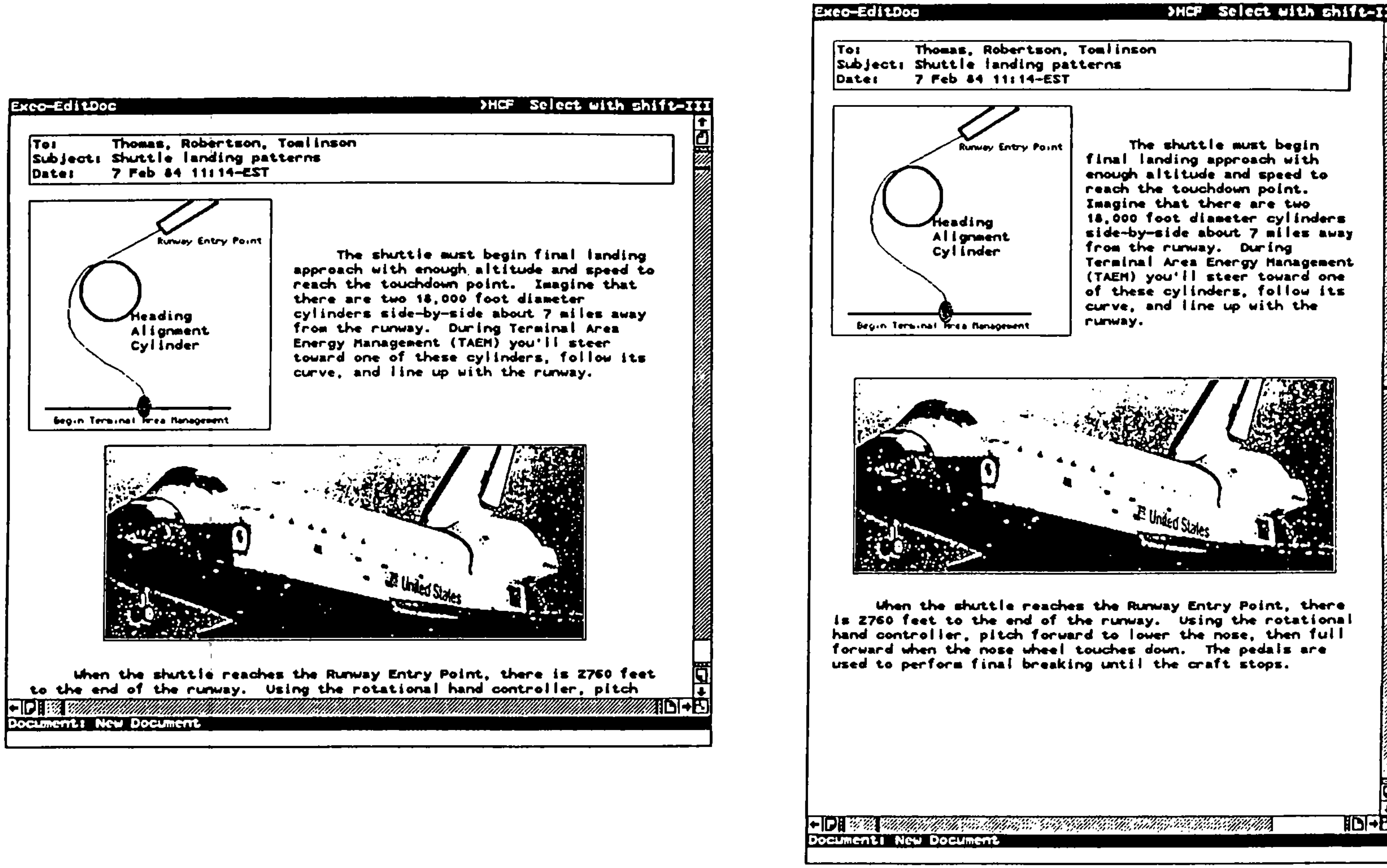
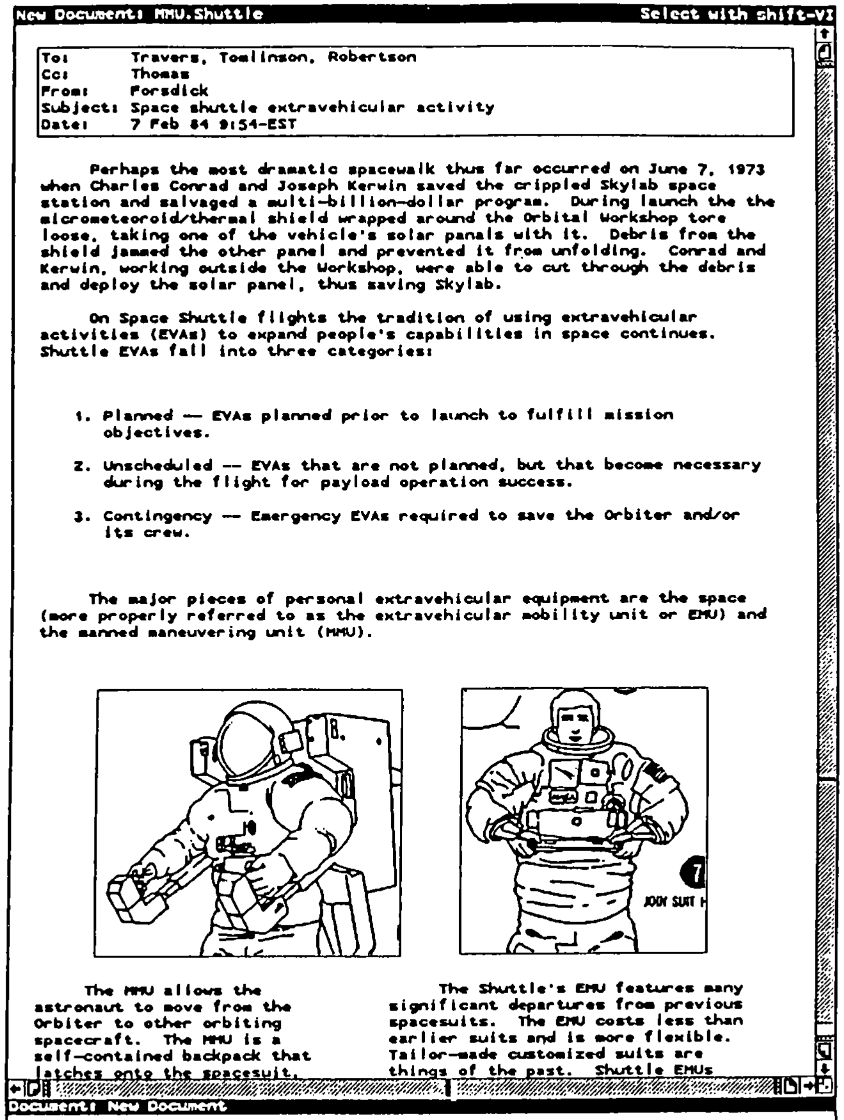


Figure 6. The same document displayed in different shaped windows.



New Document: MMU.Shuttle Select with shift-VI

To: Travers, Tomlinson, Robertson
Cc: Thomas
From: Forsdick
Subject: Space shuttle extravehicular activity
Date: 7 Feb 84 9:54-EST

Perhaps the most dramatic spacewalk thus far occurred on June 7, 1973 when Charles Conrad and Joseph Kerwin saved the crippled Skylab space station and salvaged a multi-billion-dollar program. During launch the the micrometeoroid/thermal shield wrapped around the Orbital Workshop tore loose, taking one of the vehicle's solar panals with it. Debris from the shield jammed the other panel and prevented it from unfolding. Conrad and Kerwin, working outside the Workshop, were able to cut through the debris and deploy the solar panel, thus saving Skylab.

On Space Shuttle flights the tradition of using extravehicular activities (EVAs) to expand people's capabilities in space continues. Shuttle EVAs fall into three categories:

1. Planned — EVAs planned prior to launch to fulfill mission objectives.
2. Unscheduled -- EVAs that are not planned, but that become necessary during the flight for payload operation success.
3. Contingency -- Emergency EVAs required to save the Orbiter and/or its crew.

The major pieces of personal extravehicular equipment are the space (more properly referred to as the extravehicular mobility unit or EMU) and the manned maneuvering unit (MMU).

The MMU allows the astronaut to move from the Orbiter to other orbiting spacecraft. The MMU is a self-contained backpack that latches onto the spacesuit.

The Shuttle's EMU features many significant departures from previous spacesuits. The EMU costs less than earlier suits and is more flexible. Tailor-made customized suits are things of the past. Shuttle EMUs

Document: New Document

Figure 7. A Reader's view of a Diamond Document.

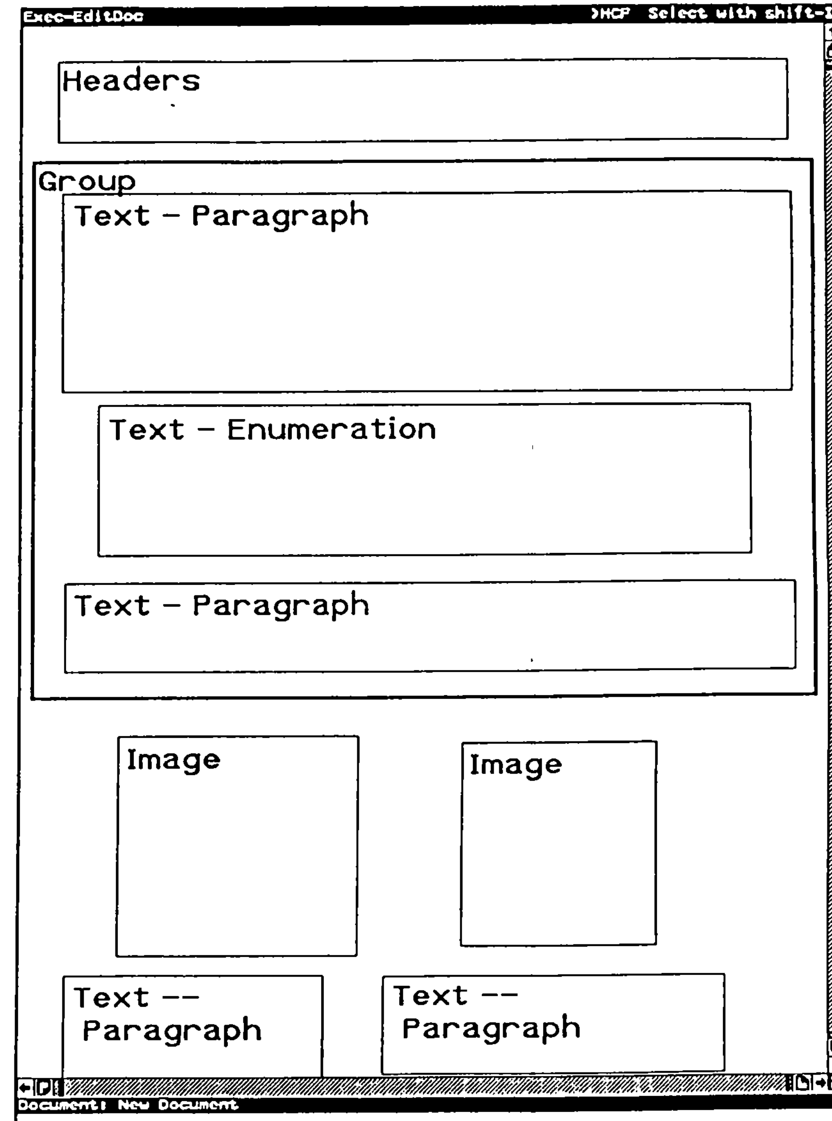
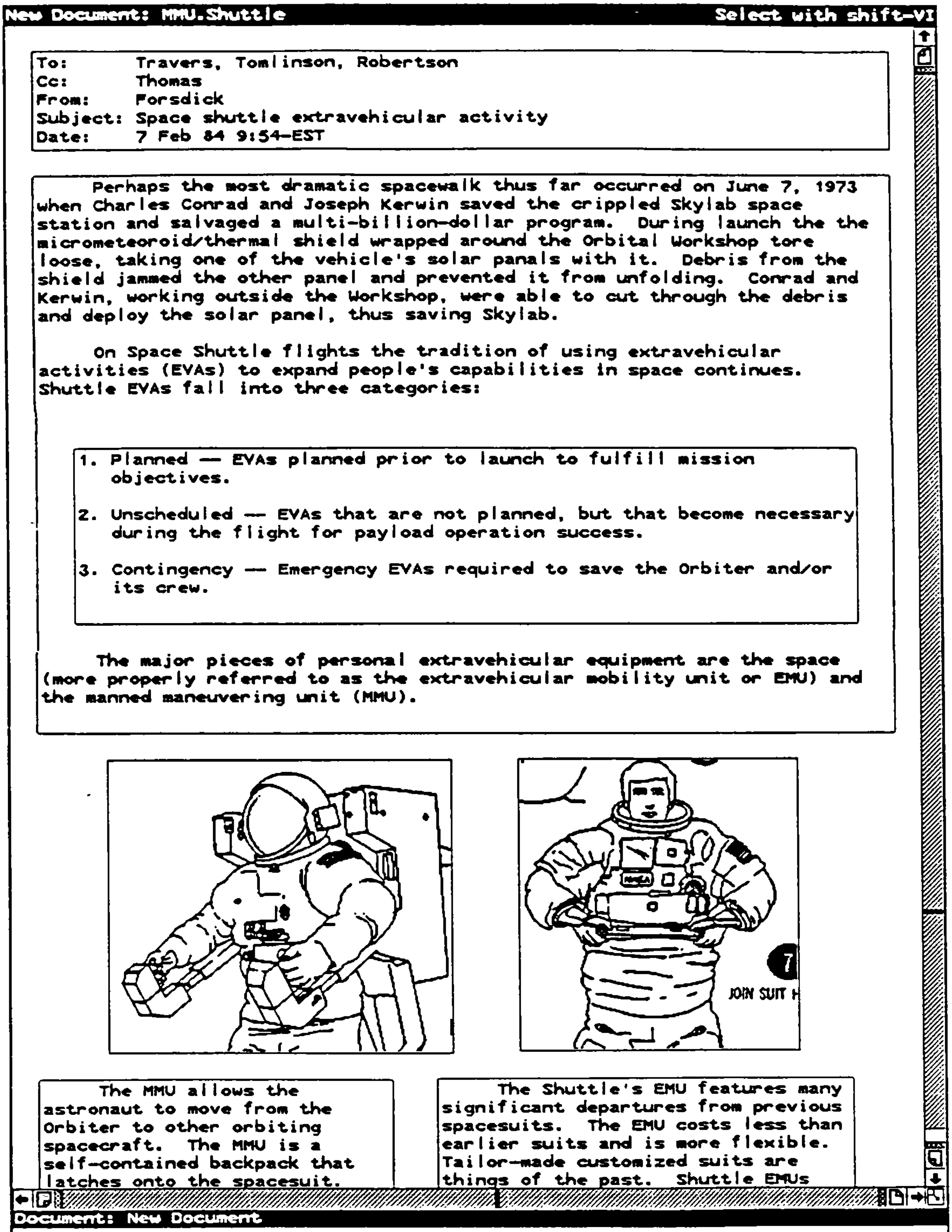


Figure 8. Underlying structure of the document presented in Figure 7.



New Document: MMU.Shuttle Select with shift-VI

To: Travers, Tomlinson, Robertson
Cc: Thomas
From: Forsdick
Subject: Space shuttle extravehicular activity
Date: 7 Feb 84 9:54-EST

Perhaps the most dramatic spacewalk thus far occurred on June 7, 1973 when Charles Conrad and Joseph Kerwin saved the crippled Skylab space station and salvaged a multi-billion-dollar program. During launch the the micrometeoroid/thermal shield wrapped around the Orbital Workshop tore loose, taking one of the vehicle's solar panals with it. Debris from the shield jammed the other panel and prevented it from unfolding. Conrad and Kerwin, working outside the Workshop, were able to cut through the debris and deploy the solar panel, thus saving Skylab.

On Space Shuttle flights the tradition of using extravehicular activities (EVAs) to expand people's capabilities in space continues. Shuttle EVAs fall into three categories:

1. Planned — EVAs planned prior to launch to fulfill mission objectives.
2. Unscheduled — EVAs that are not planned, but that become necessary during the flight for payload operation success.
3. Contingency — Emergency EVAs required to save the Orbiter and/or its crew.

The major pieces of personal extravehicular equipment are the space (more properly referred to as the extravehicular mobility unit or EMU) and the manned maneuvering unit (MMU).

The MMU allows the astronaut to move from the Orbiter to other orbiting spacecraft. The MMU is a self-contained backpack that latches onto the spacesuit.

The Shuttle's EMU features many significant departures from previous spacesuits. The EMU costs less than earlier suits and is more flexible. Tailor-made customized suits are things of the past. Shuttle EMUs

Document: New Document

Figure 9. The Author's view of a Diamond document.

information about the structure of a document to permit documents to be edited in a convenient way. Some of the more experimental features of this model (e.g., the extensibility made possible by general objects) were omitted from the initial Diamond model in favor of a simplified style of presenting documents. The DARPA Internet document model is successful in recording structural information about objects in a document, although in an effort to be machine independent, some important layout characteristics, such as object positioning, are left loosely specified. The initial Diamond model, which combines features from the Experimental and DARPA Internet models, extends the previous models in ways that facilitate editing and object layout for presentation purposes

All three models are relatively simplistic, and must be regarded as initial attempts to deal with multimedia electronic documents. More sophisticated models will evolve as experience with such documents is acquired. We plan to extend both the DARPA Internet model and the model currently supported by Diamond based on experience with the Diamond system.

References

[1] Bricklin. D.. Frankston, R.
VisiCalc (TM) Computer Software Program.
Personal Software. 1979.

[2] Crocker, D. H.
Standard for the Format of ARPA Internet Text Messages.
Technical Report RFC 822. DARPA Network Working Group, August, 1982.

[3] Finn. G. G., Katz, A. R., Postel, J., Forsdick. H. C.
Proposal for the Body Item for a Multimedia Message.
Technical Report. USC Information Sciences Institute and Bolt Beranek and Newman, Inc.. April. 1983.

[4] Forsdick. H. C. Thomas, R. H.
The Design of Diamond: A Distributed Multimedia Document System.
Technical Report 5402. Bolt Beranek and Newman. Inc., October, 1982.

[5] Forsdick. H.. Thomas. R.. Robertson, G., Travers. V.
Initial Experience with Multimedia Documents in Diamond.
In *Computer Message Service, Proc. IFIP 6.5 Working Conference*. pages 97-112. IFIP. 1984.

[6] Postel. J.
Internet Multimedia Mail Document Format.
Technical Report RFC 767. DARPA Network Working Group. March. 1982.

[7] Tremain. T. E.
The Government Standard Linear Predictive Coding Algorithm: LPC-10.
Speech Technology :40-49. April, 1982.

An Experimental Multimedia System for an Office Environment

S. Christodoulakis

Computer Systems Research Institute
University of Toronto
10 King's College Road,
Toronto, M5S 1A4

1. Multimedia Messages

We call the unit of multimedia information a *multimedia message*. Multimedia messages are composed of attribute text image and voice information. Some of the functions that systems for multimedia messages may provide are filing of multimedia information, content addressability of multimedia messages, and multimedia message transmition and reconstruction in a different site.

In a possible scenario an office worker uses the multimedia filing capability to find some information relevant to the interests of his company. He uses the extraction unit to extract some of this information, and the comparative interface unit to compare some images. When he selects an image with some statistical information he may want to alter its presentation form and/or further edit it. He uses the information extraction unit for it. Some information may be in a paper form. He uses an image digitizer and an extraction capability to extract the relevant information. Finally he may want to use all the information that he has selected so far to compose a report. The report may be transmitted through communication lines to another station.

A conceptual framework for multimedia messages has been presented in [Christodoulakis 84]. Multimedia messages have a type, a set of attributes, a text part, an image part, a voice part, and an annotation part. The text part is further subdivided into sections, paragraphs, sentences, words and parts of words. The image part is composed of an image type, a vector form (used to represent the picture as a collection of easy to store primitive objects like lines, polylines, circles, ...), a raster form (used to represent the picture as an ordered set of pixels), a statistical part (used to represent any statistical information contained in images), and a text part (which can be associated paragraphs, caption information, and words appearing within the picture). The image type can be graph, pie chart, histogram, table, statistical (any of the previous), or picture (anything else). More than one statistical object (graph, pie chart, ...) may exist in the same image. The annotation may be text annotation or voice annotation. Annotation is a further informal explanation about the contents of a message, paragraph, word, image or image object.

The presentation form of the constituents of a message may be different from the internal representation of the message.

The internal representation of an image does not have to have both an object form and a raster form. It may only have one of the two. An example of an image where both forms exist in the internal representation is a photograph where objects have been identified and stored in the object form for enhancing content retrieval. The actual photograph may be stored in high capacity devices like videodisks or directly addressable microfilm while the extracted information may reside in a disk and be used for enhancing content addressability. An example of an image having only a raster internal representation is an uninterpreted photograph. An image having only an object form as internal representation can be an engineering design. (At the presentation level however, the object form may be used to display the design in a raster display.)

The internal representation of the object form of an image is a collection of objects. With each object is stored information related to its type (polygon, circle, ...), its name, name display specifications (font, size, position of display), shading information, and the coordinates of a set of points or other information specific to object type (radius,..). This information enables the reconstruction of the set of points which compose an object.

The internal representation of statistical type images (graphs, pie charts, histograms, tables) is a collection of tables. This information is not usually displayed and it is in fact a duplication of information since the information about the objects composing the presentation of these images in a specific device is also maintained. However, the information duplication is not very large, and the approach facilitates both answering queries on the image contents and presenting the image in a different form, or the same form but with different parameters (different coordinate system say), at a later point in time. In addition it provides a means of displaying the statistical information in devices which do not have graphics or bitmap display capability.

The presentation of a multimedia message in an output device is called a physical message. With a physical message we associate some default information (such as font, size, line spacing, ...) which is used for displaying the message in an output device. A physical message is divided into physical pages. Each physical page is composed of rectangles. A rectangle can be a text rectangle or an image rectangle. Rectangles are identified by their location within a physical page and their size. Image rectangles correspond one to one to images of a multimedia message. Text rectangles may contain some information that is used for displaying messages in an output device (alternative font, alternative size,...). Since sequences of words may be displayed in a different way we also use word sequence rectangles which are contained within text rectangles.

Finally the voice message and the annotation message part of a multimedia message are not displayed in the physical message. However, the voice part of the message, voice annotation sections, and text annotation sections are mapped one to one to image rectangles and paragraph rectangles of the physical message. An indication of their existence is a special symbol associated with the relevant rectangle, which may be optionally displayed in the output device. The indication symbol can denote voice message, voice annotation section, or text annotation section.

A descriptor is associated with each created multimedia message. The descriptor indicates the parts of the message, the internal form for each part and its mapping to a physical message.

2. Content Addressability

Multimedia messages are retrieved by specifying message content information instead of a unique message identifier. The user will have some idea of what is the content of messages that he wants to see (or not see) and he will specify this information in his query. The system will try to return to him all relevant messages.

Content addressability in the attribute and text part of the message can be achieved by allowing the user to specify conditions on attribute values and words appearing within the text part of the message.

Image content addressability can be achieved by specifying conditions on the image text part, on the image statistical part, as well as similarity relationships among image objects. Retrieving messages based on conditions on the image text part is logically and physically different than specifying conditions on the text part of the message. The former specifies that the user wants to see a *message that has an image* related to the condition specified while the latter specifies a *message* related to the condition specified. It is also physically different because the search is limitted to the image text part. Retrieval based on the text part of the image is a very powerful primitive for content addressability in office information systems. In addition to image caption and related text paragraphs, in several cases words which appear within the image itself may be very usefull. Virtually every diagram, engineering design, or CAM design contains words within the image that could potentially be usefull in content addressability.

An image may contain a number of statistical objects (graphs, pie charts, histograms, tables). Each one of those has an internal representation in the form of a table. The user can focus his attention to only one of the statistical objects at a time. We do not allow content addressability based on relationships among tables. We follow this approach because we believe that it will be confusing to the user to remember which statistical objects belong to the same image. In addition, conditions on single tables may be very selective so that the size of the response is limited. However, the presentation of a message allows that more than one statistical objects (graphs, tables, ...) appear in the same picture.

Finally similarity and structural relationships of objects in pictures may be found usefull in restricting the size of responce for non-statistical images (pictures). Structural relationships of objects within an image (like contains, intersects, above, right of ...) provide a set of powerful primitives which can be used to enhance content retrieval in a given context (application environment). The internal object representation allows us to answer queries on spatial relationships among objects. Fuzzy retrieval in images can also be based on these primitives. We plan to provide such capabilities for our system in the future.

In some cases the user will have previously seen the particular message that he wants. He may remember some physical characteristics of the message. For example he may remember that the particular statistics that he wanted were presented in the form of a graph in the particular message. He may also remember the approximate location of this graph within a physical page of the message. We allow the user to enhance his retrieval capability by specifying conditions on the presentation form of the message [Christodoulakis et al. 84].

In the future other types of content retrieval like voice or special signal recognition (coming from particular stations) should be considered.

3. User Interface and Query Reformulation

The important task of the user interface is to support in a uniform and integrated way the various data forms and activities. In order to ask a query the user has to specify a filter. The specification of the filter is done using menus in a by-example fashion. This approach allow the user to specify his selection non-procedurally using a set of options and thus it presents advandages for non-expert users.

The screen is divided into two regions. The left region displays a message template. A message template has two main components: 1) A set of fields that corresponds to the attributes of the message, 2) the message body. The various components of the template are filled in during the process of query formulation. The right region of the screen is the menu area and displays the available options for definition of restrictions on the voice and image part of the message.

For restrictions on voice the options that can be specified are present, absent, or no restriction.

For image restrictions several options are available. These options are specified using menus in a hierarchical fashion. The interface for specifying conditions on the statistical part uses options that are particular to the image type (e.g. graph, pie chart, ...). However, all conditions on the statistical part are examined on the internal (table) representation of the image.

In a multimedia information system environment it may often be the case that the user cannot exactly describe the information that he wants. This is not typical of a data base environment where the information is well structured and named, and attributes take values from a fixed set of attribute values. An example from a text retrieval environment which demonstrates that this may not be true in a more general information retrieval environment is synonyms, words with similar meaning.

The user may find it even more difficult to specify queries on the image part of messages [Christodoulakis 84]. In addition, the information extraction process (when used) may fail to name or extract information for all the existing objects within an image. The system should allow a dynamic query reformulation.

It is possible that the user will feel the need for query reformulation at some point in time as he browses through the messages. One reason is that something that he saw in these messages may have prompted his memory to a better specification of his query. Another may be that he has decided that he is receiving too many documents. The query reformulation may restrict the number of qualifying documents further, it may expand the query with a disjunctive term, or it may completely change the query. We allow options for query expansion using an environment dependent thesaurus, query modification (more restrictions) and continuing the search forwards, or changing the query and restarting without seing the documents selectd so far.

4. Access Method

Multimedia messages coming in a station are stored in general files. At a later point in time a user of the station may want to view these messages or extract some information from these messages to form a new message. An access method based on abstractions is used to achieve fast responce time in user queries. An abstraction of the multimedia message is much smaller than the multimedia message itself and restricts the attention to a small number of qualifying messages.

Information stored in the abstraction file contains abstractions of text, image and voice data. The text abstraction scheme is based on superimposed coding [Christodoulakis and Faloutsos 84]. A fixed length block signature is created for each logical block of text data. Originally all the bits of the block signature are set to zero. The signature is constructed by taking each non-trivial word in the text message splitting it into successive overlapping triplets of letters and hashing each triplet into a bit position within the block signature. These bits are set to one. If the word is too short, additional bit positions are set to one by using a random number generator, which is initialized with a numeric encoding of the word. Thus a constant number of bits corresponds to each non-trivial word. The size of the signatures and the number of bits per word have been determined in such a way that the performance of the system is optimized [Christodoulakis and Faloutsos 84].

To examine if a given word appears within a logical block of the message, the signature of this block is examined. The same transformation is performed on the word and the bits determined by the transformation are examined. If they are all one, the word is assumed to appear in the text message. This access method retrieves supersets of the qualifying messages. The browsing capability described before allows the user to pinpoint the relevant messages. Parts of words can also be specified in queries. More complicated query patterns (including conjunctions and disjunctions of words) can be examined versus the signature in an obvious manner. Information related to attribute values is also abstracted using a signature technique. The only difference is that order preserving transformations are used in order to answer inequality queries.

Some important advandages of the technique are that it can easily handle parts of words in queries, it is suitable in the case that errors may exist in the text (which may be frequent in this particular application environment), as well as that it requires simple and uniform software and it can easily accomodate other types (like queries on images or on presentation form) [Christodoulakis and Faloutsos 84]. Some further evaluation showed that the approach is more appropriate for an information system environment than word signatures [Tsichritzis and Christodoulakis 83] or rigid indexing techniques like in IBM STAIRS.

Important information regarding images like the image type and approximate location is also inserted in the abstraction file. In addition information related to the objects of a picture as well as an abstraction of the image text part and the statistical part is also inserted in the abstraction file. Finally the only information related to voice that is inserted in the abstraction file is information related to the absence or existence of a voice section in the message.

As with text information, the information used for answering queries involving attributes, pictures, and voice which is stored in the abstract file garantees that a superset of the qualifying messages to a given request is retrieved. The blocks of the access file are accessed sequentially. The sequentiality of access, the use of large blocking factors, and the small size of the access file result to a small cost of the access method.

5. Status and Future Research

Our approach is both anaytical and experimental. We have already implemented two versions of a prototype for multimedia messages [Christodoulakis et al. 84], [Tsichritzis et al. 83]. Most of the system capabilities described above have been already incorporated into the prototypes. Our current research seeks to evaluate the various ideas that have been implemented as well as enhance the capabilities of the system.

References

[Christodoulakis 84]

S. Christodoulakis: Framework for the Development of an Experimental Mixed-mode Message System, Proceedings ACM-BCS Research and Development in Information Retrieval, Cambridge Press, 1984.

[Christodoulakis et al. 84]

S. Christodoulakis, J. Vanderbroek, J. Li, S Wan, Y. Wang, M. Papa, E. Bertino: "Development of an Information System for an Office Environment", Proceedings VLDB 84.

[Christodoulakis and Faloutsos 84]

S. Christodoulakis and C. Faloutsos: "Performance Analysis of a Message File Server", IEEE Transactions on Software Engineering, March 1984.

[Tsichritzis and Christodoulakis 83]

D. Tsichritzis and S. Christodoulakis: "Message Files", ACM Transactions on Office Information Systems, 1,1, 1983.

[Tsichritzis et al. 83]

D. Tsichritzis, S. Christodoulakis, P. Economopoulos, C. Faloutsos, A. Lee, J. Vanderbroek, C. Woo: "Multimedia Office Filing System", Proceedings VLDB 83.