

## A Multimedia Office Filing System

D. Tsichritzis, S. Christodoulakis, P. Economopoulos, C. Faloutsos  
A. Lee, D. Lee, J. Vandebroek, C. Woo

Computer System Research Group  
University of Toronto, Canada

### Abstract

The paper outlines an Office Filing System which can be used for multimedia messages. The system uses signature techniques for fast filtering. It uses miniatures, voice excerpts and a game environment for effective browsing and selection of the desired messages.

### 1. Introduction

Our objective is to design and implement a facility for filing office objects. With the advent of the widespread use of message systems such a facility is very much needed. As people exchange text messages, voice messages, records and facsimile they will need to file them and retrieve them in a flexible manner. Such filing activity serves two purposes. First, it enables the users to store and retrieve the information relevant to their own work. Second, it enables the system to retain information from which it can feed corporate data base to augment the "corporate memory" [MoRo79].

We will refer loosely to messages as the office objects related to filing. A message can be a *data base record*, a *text message*, a *voice message*, or an *image*. It can also be any combination of the above. For instance, a message may consist of:

- a) attribute values, e.g., date, sender
- b) text part, e.g., letter contents
- c) voice part, e.g., voice annotation
- d) image part, e.g., digitized photos

A message consists of a header which has a unique id and perhaps rules related to its behavior [HMGT83]. The contents consist of different sections of attribute values, text, image, and voice. We want to provide a facility for filing and retrieving such multimedia messages.

A simple way of filing and retrieving messages is in terms of labels. Each message is labeled with a name and it is stored in a separate file. It is retrieved through

a search of the file directories, e.g., UNIX hierarchical file directories. The approach is effective irrespective of the nature of the message's contents. It is equally applicable to data, text, voice, and images, or any combinations. The management of names, however, becomes difficult for the user and it does not work well in the presence of many messages.

Another simple approach is to file all messages sequentially and to search them sequentially to select the needed messages. It is the method used when doing a library search using a microfiche reader. The ordering of the messages can facilitate the search, e.g., alphabetic, chronological, etc. The method works well when we don't have many messages and/or the messages have an order which is very meaningful to the user. However, it is time consuming to sequentially scan all messages, when we have many messages and want to access them in many different ways.

A third approach is to abstract some properties of the messages and encapsulate them in attribute values. The approach is used in Information Retrieval when doing keyword searches. The search is effected in terms of a selection of attribute values. The selection filter is specified through a query involving a Boolean expression of simple, *attribute <op> value*, conditions. The method is effective when the attribute values adequately represent the properties of the message and when the environment is static. These requirements are not easily fulfilled when messages have pictures, or voice. It implies a priori knowledge of the properties which are important for searching purposes.

A fourth approach is to retrieve messages according to a pattern present in them [Salt80, FIUI80, Hask81, AhKW78]. This approach works well for text. The text part of a message can be qualified according to a regular expression of strings (words, combinations of words) present in them. For voice and pictures, however, patterns are not easy to define and they often require complicated and time consuming pattern recognition techniques [Redd76, BaBr82, EHLR80]. Note that what can be a natural pattern for the human eye/ear is not as easy to pin down in terms of computer-oriented messages.

Finally, a fifth approach to retrieving messages is to encapsulate their properties in terms of abstractions which are easy for the users to recognize. The users proceed searching for the messages with the aid of these abstractions. These abstractions can be closely related to the message, e.g., a *miniature* image of the message. Abstractions can also be *unrelated to the exact contents* of the message, e.g., a particular type may identify a person or an icon for an idea. An association easily recognized by the user relates the seemingly independent abstraction of the message to the message

itself.

In this paper we would like to deal with multimedia messages. We will use, therefore, a combination of the above techniques for flexible message retrieval. In this way, the facility would be effective for each medium of communication and it will be especially suitable for combinations of data, text, voice and pictures.

Information retrieval facilities consist usually of two parts; a *filtering* capability and a *browsing* capability. Filtering enables the user to specify what he would like to see, or equivalently the messages which he does not like to see. The browsing capability enables the user to pinpoint from the filtered messages the ones which he actually wants. In many systems the browsing capability is only an after-thought (especially true for data base systems). It deals only with the presentation of the selected messages to the user. It is not considered an integral part of the selection. In addition, the filtering and browsing are considered as two independent and consecutive steps without any relation to each other. In the case of office filing the browsing capability is very important. We will consider it as important for selection purposes as the filtering capability. This approach is needed because the user filters are rather vague. The user does not adequately remember what he is looking for. Filtering alone cannot pinpoint the desired messages. In addition, voice and image filtering according to contents is difficult to implement because it may imply pattern recognition. In this case it is advantageous to use browsing rather than filtering.

We believe that the browsing aspect is a dual method to the filtering for selection purposes. We provide therefore, "play" methods supporting browsing in the same way that we have access methods supporting filtering. We also allow filtering and browsing to be interleaved. That is, while browsing we can modify the filter for selection of the messages we are currently browsing. In this way filtering and browsing proceed concurrently enabling the user to pinpoint the appropriate messages. The additional advantage of this approach is that the dynamics of the interaction between the user and the system are greatly improved. The user does not get bored waiting for the filtering, nor swamped with its results when they come in bursts. Instead the user is provided with a continuous stream of filtered messages from which he can select, by advanced browsing methods, the messages he wants. The browsing is also implemented as an interesting game to further appeal and retain the interest of the user.

## 2. General Design

Messages in our Office Filing System consist of a unique id and a number of *fields*. Each message has a *date* field, a *sender* field and a *subject* field which are attribute fields. Attribute fields have a maximum length and they take single values from a domain of values. In addition, a message has fields which are unstructured and of variable length. These fields consist of text, image, and voice annotations. Images include graphs, tables, captions, bar charts, pie charts, diagrams, and pictures. Images may appear anywhere in the message. Voice annotations are parts of the message that are used to clarify and enhance it. For instance, they can be verbal comments about the message or an utterance to attract the attention of the reader.

All incoming messages are filed into a general message file. The user searches for the required messages guided by a vague recollection of the contents of the messages and by a vague image of what the messages look like. The user provides initially a partial

specification of the message contents [Zloo75]. This partial specification of the desired messages acts as a filter. The filter restricts the attention of the messages in the message file to a manageable subset. The filter can be changed dynamically by tightening its specification.

The filtering capability is by no means an exact one. The user seldom specifies an accurate filter. His specification will allow more messages to qualify in addition to the ones he absolutely wants. These additional messages are eliminated in the browsing mode by the user. To assist the user in identifying the appropriate messages, *miniatures* and *fasttalk* are provided. Miniatures are realistic visual abstractions of the messages which are displayed for the user during browsing like in [FeND81]. At the same time when the miniature appears in view to the user, the fasttalk can be heard. The fasttalk is a voice excerpt associated with the message which highlights the message's meaning. On the basis of what the user sees and hears, he can decide if the message is one of the ones he wants retrieved. If so, the message corresponding to the miniature is displayed to the user along with a playback of the voice annotations for the messages.

To select the miniature to be viewed in full, the user identifies the particular miniature by shooting it down with a cannon. In the current implementation, this cannon is stationary and the miniature is hit when it appears within the target area of the cannon. Using miniatures and fasttalk rather than the messages themselves, the user can play many more filtered messages. In this way his browsing capability is enhanced. In addition, the presence of many more message abstractions enables the user to spend more time on the more interesting candidates. We also plan to allow the user to control the speed that the abstractions are played.

## 3. User Interface

The screen layout of our system appears in Figure 1<sup>1</sup>. The status of the display depends on the current mode. There are four modes:

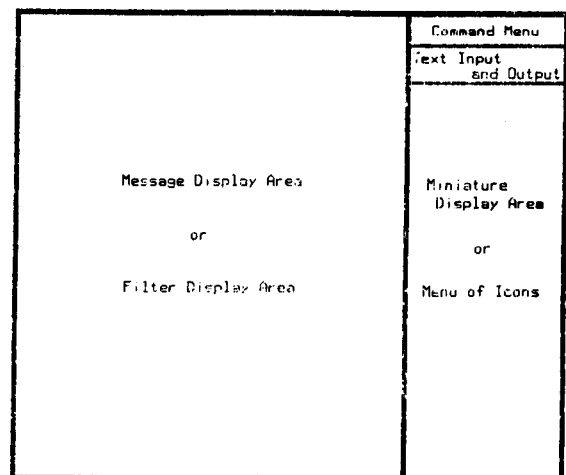


Figure 1: Screen Layout

Our implementation environment consists of a SUN computer which provides one and a half page of bit-map display. The left part is a whole page of message, while the right is half a page of screen real estate used for menus and miniatures.

- (1) **Create/Append** - the user creates or appends to the filter. This state can be recognized by the appearance of the filter template and the contents of the filter (in the case of Append mode) on the left and the icon<sup>2</sup> menu area to the right of the screen.
- (2) **Browsing** - the user is playing the abstractions (miniatures and fasttalk) for the messages that are filtered by the system. Miniatures are scrolled on the right hand and the filter remains on the left part of the display.
- (3) **Viewing** - the user has just frozen the browsing of the miniatures to view one of the messages in more detail. The expanded message appears on the left while the right screen remains frozen.

The *filter* used by the searching process is the conjunction of all the restrictions on the data, text, voice, and image values of a message. The template of the filter and its menu appears in the Filter Display and Icon Menu areas as shown in Figure 2.

Restrictions on attribute and text values are provided as values and patterns. For each of the attribute and text fields, a field restriction may be defined. Each field restriction is a conjunction of conditions. The syntax of the field restriction is given by the following pseudo grammar:

```

field_restriction = element | field_restriction '&' element
element = string | element '|' string
string = word | wordpart | string word | string wordpart
word = letters
wordpart = 'X'|letters | letters'X', 'X'|letters'X'
letters = digit | char | symbol | letters digit |
          letters char | letters symbol
  
```

A field restriction need not be provided for each of the data and text field. In such cases a null field restriction for this field is assumed. The field restriction is entered in the appropriate field entry on the filter using a by example approach [Zloo75].

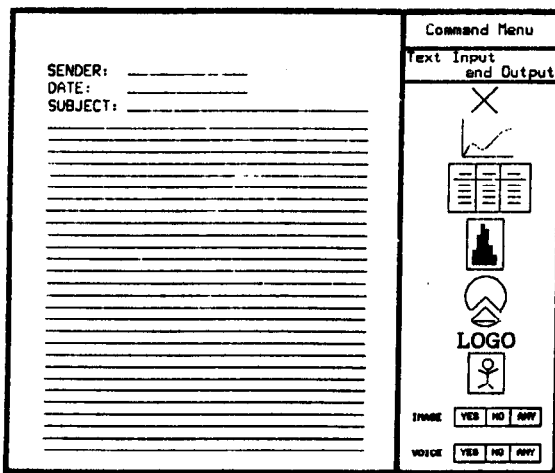


Figure 2: Screen Layout of Create/Append

<sup>2</sup> The icon is a graphical representation of an object [Lod483]. These icons are similar to those found in the Star [SIKH82] and the Apple Lisa [WILL83].

Restrictions may be placed on images in the form of:

- (1) There are images present in the messages being sought. The user specifies the approximate location of where these images reside in the message and their type.
- (2) There are no images present in the messages being sought.
- (3) A null restriction in which the messages being sought may or may not contain images.

The selection of the appropriate condition is facilitated by selecting the appropriate light button beside the image entry in the filter.

If the user specifies that there are images in the desired messages, then the positions and image types are identified by dragging the 'x' icons or object icons (e.g., graph, pie chart, etc.)<sup>3</sup> and positioning them in the appropriate place on the filter template. The 'x' icon (representing any type of image) and the object icons are picked up from the menu. Selecting an object icon implies that the desired image is represented by this object icon<sup>4</sup>. Within a filter create or update session, the user may drag any of the positioned icons (those placed in the same session) into the menu area to remove that image from the filter.

The user specifies the voice restriction by selecting the appropriate light button (for voice) corresponding to whether voice annotations is present, absent, or null for don't care.

Commands for the Office Filing System are located in a command line area at the bottom of the screen. The appropriate command is selected using the appropriate light button. Different command menus appear depending on what mode the user is in. The command menus for the various modes are:

#### Create/Append Mode

Return to  
Browsing

#### Browsing Mode

Exit	Create Filter	Append Filter	Scroll Down	Fasttalk Off
------	---------------	---------------	-------------	--------------

#### Viewing Mode

Exit	Return to Browsing	Create Filter	Append Filter	Play Voice
------	--------------------	---------------	---------------	------------

To exit the Office Filing System, the Exit button is selected while in the Browsing or Viewing mode. When the Create Filter light button is selected, it indicates that the user is to provide a new filter. A blank filter template appears to the left part of the screen and the

<sup>3</sup> The icons appearing in Figure 2 represent the image types recognized in the Office Filing System. They are, from top to bottom:

- 1) any image type
- 2) line graph
- 3) table
- 4) bar chart
- 5) pie chart
- 6) logo or letterhead
- 7) picture (i.e., any other image type)

<sup>4</sup> In our implementation, an image is a single indivisible object (e.g., pie chart).

user is in the Create mode. After all the restrictions are added, the Return to Browsing button is selected. When the Append Filter light button is selected, it indicates that the user wants to augment the existing filter with more restrictions. The user can now edit in the restrictions and Return to Browsing when finished. The Scroll Down light button when off indicates that the miniatures are displayed from the bottom to the top versus top to bottom (Scroll Down is on)<sup>5</sup>. The user can turn the playing of the fasttalk on and off by turning the Fasttalk Off light button off and on, respectively. As long as this light button is on, no fasttalk is spoken<sup>6</sup>. In the Viewing mode, the user can play the voice annotation turning the Play Voice light button on<sup>6</sup>. To return to the other two modes, the user selects the appropriate return light button.

#### 4. Abstractions from Messages

Information abstracted from the messages are:

- (1) Signatures
- (2) Miniatures
- (3) Image Description
- (4) Fasttalk

In our system, we use a signature technique as an access method for attribute and text values [TsCh83]. The method is based on superimposed coding [ChFa83]. A fixed length signature, which is a bit string, is created for the attributes. A separate signature is created for each block of the body. These signatures within the block signature are determined by taking each non-trivial word in the body or in the attributes, splitting it into successive, overlapping triplets of letters and hashing each triplet into a bit position. If the word is too short, additional bit positions are created 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. These bits are set to one. 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.

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 message. Otherwise, the message is skipped. This access method retrieves supersets of the qualifying 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.

The miniatures for the messages are formed by first taking each word within the message and representing it with a variable line thickness (i.e., to account for the ascenders and descenders in the letters of the word). Then, the bit-maps of the images are extracted and an "n" factor reduction is performed (i.e., reduce every "n" bits into one bit). This reduction is sensitive to bits that are on. That is, if a majority of the "n" bits are off then the one bit is turned off. Otherwise, the bit is turned on. To complete the miniature, the reduced bit-maps are merged accordingly with the textual portion of the message.

<sup>5</sup> The speed of the scrolling is limited to the length of the fasttalk and the Rasterop operation.

<sup>6</sup> In our current implementation, voice annotations and fasttalk are played back at normal speeds.

Simple image descriptions can be abstracted from the message, such as the image types present in it (e.g., graph, table, bar chart etc.) and their positioning. This information will be automatically gathered, since it is reasonable to assume that the image creation will be conducted with the aid of specialized image editing tools that are aware of the image type being created.

In our current system, the fasttalk is created manually by the user. It contains a short (one to two seconds of talk) description of the message's contents, or an excerpt of the message. In the future, we are hoping to use automatic techniques to obtain a fasttalk which highlights the voice annotation of a message.

#### 5. Implementation

The implementation of the Office Filing System is divided into three processes.

The insertion process is used to add new messages to the message file. In addition, this process generates the appropriate search aiding entries for the new messages. The search process will search for messages satisfying the search filter. The user interface process, as we described earlier, is concerned with the specification of a tighter filter and allows the user to browse the miniatures through a game playing environment and viewing a message in more detail.

Communication between the user interface and the search process are limited to two uni-directional pipes. The user interface process passes the filter information, the directory where the files are, and the commands to change the search direction to the search process along pipe 1. The search process in turn passes to the user interface process the messages that meet the restrictions of the filter along pipe 2. In the rest of the paper we will elaborate on the insertion and searching capabilities of the system.

The user provides four files to the insert process. These four files contain various pieces of the message (i.e., text and attributes, images, voice annotations, and the message's fasttalk<sup>7</sup>). We plan to use automatic techniques in the future to split a multimedia message into its media components. The insert routine processes the information contained in the four files and appends them to the appropriate files and relations of the message file.

The message file is really seven files and one data base relation. The four input files are appended to the corresponding four files of the message file. The remaining files and relations of the message file are created by the insertion process. These are described as follows:

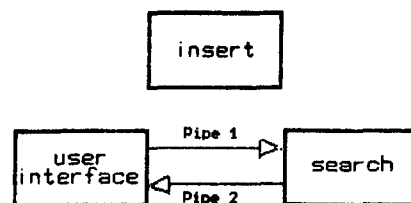


Figure 3: The Office Filing System Implementation

<sup>7</sup> At this stage of the implementation, we assume a text editor has created the text and images portion of the message and a simple voice editor has created the voice files.

- An ASCII file is provided as input which contains the text and attribute components of the message. The insert routine generates the signature entries for the text and attributes components of the message and places it into the signature file of the message file.
- A file containing the position and size information for the images present in the message is provided as input. It also contains the bit maps for these images and information about the image type. Using the contents of the ASCII file and this file, the miniature is created and placed into the miniatures file of the message file. The positioning information along with the information about the image types is stored by the insert routine into a database relation of the message file.
- Two files are provided that contain the voice annotation and the fasttalk portion of the message, respectively. For the initial implementation, the files contain analog signals of the corresponding voice annotation and fasttalk, and are stored on a separate direct addressable audio storage device [Inst82].
- The last file of the message file is a pointer file. It contains information about the other six files of the message file. Each entry in this file gives the location and size of each portion of the message in the corresponding files.

The "insert" routine consists of the three subprocedures that update the files and relations of the message file based on the input files (refer to Figure 4). It also updates the pointer file according to the information returned by them.

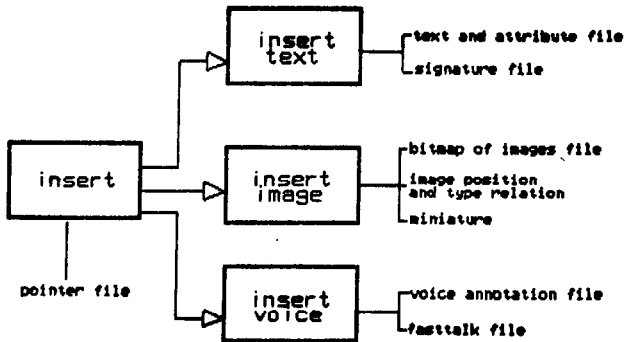


Figure 4: Insert Procedure Outline.

The "search" routine calls three search procedures to progressively isolate the messages that satisfy the filter conditions. The sequence of calls and inputs shown in Figure 5.

Each message is passed through the three search procedure sequentially and the respective medium restriction specified in the filter are checked. The messages that finally pass 'search text' are those that qualify under all the medium restrictions contained in the filter including the one for the text and attributes.

The 'search image' routine first checks the entry in the pointer file to see if the message contains the minimal number of images that are required. If so, the routine verifies that the positioning of the image is indeed within the minimum rectangle for the images and the types of the images match those specified in the filter by checking the corresponding database relation. If the absence restriction is specified, the search process merely checks to see that there are no images in

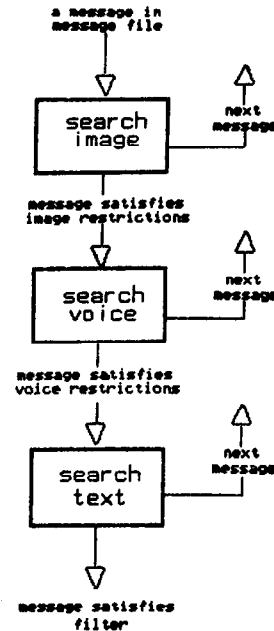


Figure 5: Search Procedure Outline.

the message.

The 'search voice' routine requires only a search for the presence or absence of voice annotations in the messages.

The 'search text' routine retrieves the signatures for the incoming message and examines if the message satisfies the restrictions on the text and attribute values.

The results of the searcher is a filtered subset which is passed to the the user interface for browsing. The miniatures for these messages are displayed and the fasttalk are played. When the fasttalk of a given messaged is played, the corresponding miniature of the message is highlighted by a box on the screen.

## 6. Concluding Remarks

The Office Filing System outlined in this paper has been implemented using UNIX on a SUN computer [ThR178]. The SUN [Sun 82] is a MC68000 based system that combines high-performance graphics, processing, and networking capabilities in a desk-top workstation. It has a high resolution (1024 by 800 points) bit-map display that can show two pages of text and graphics of a reasonable resolution. A hand-held pointing device called a mouse facilitates input of graphical information. The SUN UNIX operating system is based on the Berkeley 4.2bsd version of UNIX and the ARPA IP/TCP protocols [Sun 82]. The Ethernet local area network connection allows SUN workstations to share resources and to access such services as electronic mail, file storage, and printing.

We are using the Instavox RA-12 Rapid Access Audio Unit [Inst82] for storing voice messages. Voice messages are stored on 5-1/4 inch diskettes, each of which can contain about 27 minutes of speech. However, voice messages are represented in analog form and, as such, changes in playback speed are not allowed. We are planning to incorporate the capability of

increasing the playback speed of fasttalks so that a faster scanning rate of the messages can be achieved.

We use signatures as an access technique to speed up text search. We are also considering the use of special processors to support our system. A basic requirement is that the processor must be able to handle both structured and unstructured files. We are planning to use the Intel IDBP database processor [Inte82] because it provides facilities for managing both structured and unstructured files. For instance, it provides 'JOIN' operations for relating structured files together and 'CONNECT' operations for relating structured files and unstructured files together [Lowe82]. Therefore, separate structured and unstructured files (e.g., memo heading, text, image and voice parts) can be CONNECTed together and manipulated as an integrated message. Although the IDBP can be used for full text search, we can also employ signatures to further enhance the performance. Finally, the IDBP can be used as a file server connected to an Ethernet in order to allow sharing of the facility by several users.

The following are in our opinion the highlights our system.

- a) It stores and retrieves mixed media messages.
- b) It interleaves filtering and browsing for flexible message selection.
- c) It uses signatures as an access method for text selection.
- d) It uses miniatures and fasttalk as abstractions to aid the user in pinpointing faster the desired messages.
- e) It uses information about images in terms of their type and their positioning.
- f) It uses a game to improve the user interaction and retain his interest.

It should be noted that the selection in terms of attribute values and text patterns is mainly based on filtering and appropriate access methods. The selection in terms of voice and images is mainly based on playing fast abstractions in a game environment. In this way, the user's ability for fast and effective browsing is enhanced.

#### Reference

- [AhKW78] Aho, A.V.; Kernighan, B.W. and Weinberger, P.J. *Awk - A Pattern Scanning and Processing Language* (2nd edition). September 78.
- [BaBr82] Ballard, D. and Brown, C. *Computer Vision*. Prentice Hall, 1982.
- [ChFa83] Christodoulakis, S. and Faloutsos, C. Design Considerations for a Message File Server. *IEEE Transactions on Software Engineering* (to appear).
- [EHLR80] Erman, L.D.; Hays-Roth, F.; Lesser, V.R. and Reddy, D.R. The Hearsy II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computing Surveys* 12, 2, June 80.
- [FeND81] Feiner, S.; Nagy, S. and van Dam, A. An Integrated System for Creating and Presenting Complex Computer-Based Documents. *Computer Graphics* 15, 3, August 81.
- [FIUI80] Floyd and Ullman. The Compilation of Regular Expressions into Integrated Circuits. *Proc. 21st Symposium on Foundations of Computer Science*, October 80.
- [Hask81] Haskin. Special Purpose Processors for Text Retrieval. *Database Engineering* 4, 1, Sept. 1981.
- [HMG83] Hogg, J.; Mazer, M.; Gamvroulas, S. and Tschritzis, D. An Intelligent Mail System. *University of Toronto Technical Report CSRG-150*, June 1983.
- [Inst82] Education and Information Systems, Inc., Operating Manual for the Instavox RA-12 Rapid Access Audio Unit. Education and Information Systems, Inc., Champaign, Illinois, 1982.
- [Inte82] Intel Corporation, Intel Database Processor (IDBP) System Summary. Intel Corporation, Austin, Texas, May 1982.
- [Lodd83] Lodding, K. Iconic Interfacing. *IEEE Computer Graphics and Applications*, April/March 83.
- [Lowe82] Lowenthal, E. Multiuser Microprocessor Systems Get a Data-base Manager. *Electronics*, 55, 13(June 1982), 113-117.
- [MoRo79] Morgan, H.L. and Root, D. A Concept of Corporate Memory. *Proc. of the NYU Symposium on Office Automation*, May 79.
- [SIKH82] Smith, D.C.; Irby, C.; Kimball, R. and Harslem, E. The Star User Interface: An Overview. *Proc. NCC*, 1982.
- [Redd76] Reddy, D.R. Speech Recognition by Machine: A Review. *Proceedings of the IEEE*, 64, 4(April 1976), 501-531.
- [Salt80] Salton, G. Automatic Information Retrieval. *IEEE Computer*, 13, 9(September 1980), 41-56.
- [Sun 82] Sun Microsystems Inc., The Sun Workstation Architecture, 1982.
- [ThRI78] Thomson, K. and Ritchie, D. The Unix Time-Sharing System. *Bell Technical Journal*, 57, 6 (1978).
- [TsCh83] Tschritzis, D. and Christodoulakis, S. Message Files. *ACM Journal on Office Information Systems* Feb. 1983
- [Will83] Williams, G. The Lisa Computer System. *BYTE*, February 1983.
- [Zloo75] Zloof, M.M. Query By Example. *Proc. NCC*, 44 May 1975.