Human Factors Challenges In Creating a Principal Support Office System—The Speech Filing System Approach

JOHN D. GOULD and STEPHEN J. BOIES
IBM Thomas J. Watson Research Center

This paper identifies the key behavioral challenges in designing a principal-support office system and our approaches to them. These challenges included designing a system which office principals would find useful and would directly use themselves. Ultimately, the system, called the *Speech Filing System* (SFS), became primarily a voice store and forward message system with which users compose, edit, send, and receive audio messages, using telephones as terminals. Our approaches included behavioral analyses of principals' needs and irritations, controlled laboratory experiments, several years of training, observing, and interviewing hundreds of actual SFS users, several years of demonstrating SFS to thousands of potential users and receiving feedback, empirical studies of alternative methods of training and documentation, continual major modifications of the user interface, simulations of alternative user interfaces, and actual SFS usage analyses. The results indicate that SFS is now relatively easy to learn, solves real business problems, and leads to user satisfaction.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems; H.4 [**Information Systems Applications**]; H.4.1 [**Information Systems Applications**]: Office Automation; H.4.3 [**Information Systems Applications**]: Communications Applications

General Terms: Human Factors

Additional Key Words and Phrases: Principal support system, office of the future

1. INTRODUCTION

1.1 Purpose of this Paper

The intent of this paper is to highlight the key *behavioral* challenges we faced in designing a principal-support office system, the approaches we took toward addressing them, and our general results. (Principals are business people, including executives, managers, professionals, and salespeople who rely upon secretaries to assist them in their work, especially through typing.) This paper describes the close coupling between continually on-going behavioral research and the way this motivated changes in the design and development of a prototype office system, called the *Speech Filing System* (SFS). The significance of focusing upon behavioral challenges is that we believe behavioral factors, and not technical factors, limit the arrival of the excellent offices of the future. Useful functions must be identified and easy-to-use human interfaces must be created. Both can be imple-

Authors' address: IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

mented, tested, and improved upon with the behavioral methodology and systems technologies already in existence (see [11]).

1.2 Key Issues

The goal of the work, which began in 1973, was to create a useful computer-based principal-support office system. We had the intuitive belief that this was possible, and that successful solutions would not only lead to more productive and happier principals but would, in the long run, move toward reducing stagnating U.S. productivity and health-related work-stress problems.

Issue 1. *What should the system do*? Computing was being increasingly extended to offices through word processing, database, and electronic mail systems. Secretaries were receiving new tools, but principals were not. The key reason was not the lack of economic opportunity; indeed, 15 million U.S. principals represented an attractive marketing potential. Rather, nobody could figure out what principals needed. Traditional analytic attempts to identify isolated, repeatable principal activities which could then be automated were unsuccessful. On the useful side, activity analyses revealed that principals spent nearly all of their time in communicating interactively and that the second largest amount of time was spent in composing and reading documents (e.g., [16], [19]). These findings and our own initial behavioral analyses of principals' activities, needs, and irritations suggested the importance of improving the efficiency and quality of document composition and principal interpersonal communication. To do this, we focused first upon enhancing dictation as a method of composition and, second, on creating a noninteractive voice communication system.

Issue 2. *Could dictation be significantly enhanced*? We observed that handwriting was the main method by which principals composed. We had the intuitive belief that dictation was a potentially superior method of composition. Dictating is potentially five times faster than writing, on the basis of estimates of maximum writing and speaking rates when composition is not required (see [6]). Dictation may also be qualitatively superior: potentially faster transfer of ideas from limited capacity human working memory to a permanent record may reduce forgetting attributable to interference or decay.

Issue 3. *Would noninteractive communication be useful*? At the beginning there was no strong evidence that it would be. In 1973 noninteractive communication involved delays, usually of days, in communicating and feedback. It involved printed material (e.g., the U.S. mail). It was, and still is, often a formal medium, with emphasis upon historical record. Principals, on the other hand, communicate interactively and informally [16]. There were no audio noninteractive communication systems (except that people occasionally mailed audio cassettes), and computer-based electronic mail was barely around and not used by office principals. (In universities, electronic video mail systems were just beginning to come into use, e.g., ARPANET mail, HERMES, EDUMAIL, and EIES, see [25]).

Issue 4. *Would principals actually—and directly—use a computer system, even though it was useful to do so*? Office principals did not use computers in their work in 1973, and there was a general belief that they would not. The challenge was to get principals to directly use SFS, rather than have their

secretaries use it for them. This was important if content messages were to be sent regularly, if the maximum potential of SFS was to be realized, and if there were to be clear displaceable costs (as opposed to only value-added, or "more productive principals," see [10]).

Issue 5. *Would principals send content messages*? Telephone "yellow slip" messages rarely contain content—only the name and number of the person who called. Further, people generally do not leave content messages on traditional telephone answering devices. We were concerned that principals might use SFS only as a secondary or backup system (i.e., only when they could not get hold of a person on the telephone). Perhaps even here a user would not send a content message, but only ask a recipient to call back. We believed that content messages were important, however, if noninteractive communication was to be useful. A content message in a user's own voice and words would be very useful, we thought, since this would eliminate the potential communication errors introduced by intermediaries and the need for transcription, as well as eliminating telephone roulette.

1.3 What Is SFS?

SFS is a voice store and forward office system intended to augment communication among principals—which is their main activity [16]. With SFS a principal can create a voice message by calling a designated telephone number, keypressing his or her last name and password, and speaking into the telephone (see [10] for a fuller description). Principals can listen to and edit their messages by keypressing commands on the telephone, and can send them to other users or to predetermined distribution lists by keypressing those names. Feedback and prompts have been developed over the years to make SFS easy to learn and use, to eliminate mistakes, and to make users feel comfortable. The recipient hears the message in the sender's own voice, with the actual prosody and inflections. Unfilled pauses are automatically compressed. A principal can send or listen to messages from any pushbutton telephone anywhere, as illustrated conceptually in Figure 1. A principal can review and annotate messages, have them typed, and control what recipients can do with them.

SFS contains enhanced dictation facilities for creating, formatting, reviewing, editing, and listening to audio documents.

SFS is one of the few office systems designed to be used directly by office principals. The services it provides can increase white collar productivity, substantially decrease the time delays associated with many office communications, and reduce the frustrations commonly experienced when attempting to interact with one's colleagues. Telephones are the terminals that make SFS convenient to use from just about anywhere and at almost any time. However, these services are only valuable if the principal is willing to use the system directly for important business communications.

1.4 History of SFS

SFS started as a research project at the IBM Thomas J. Watson Research Center in 1973, under the leadership of Stephen Boies. The first running prototype was

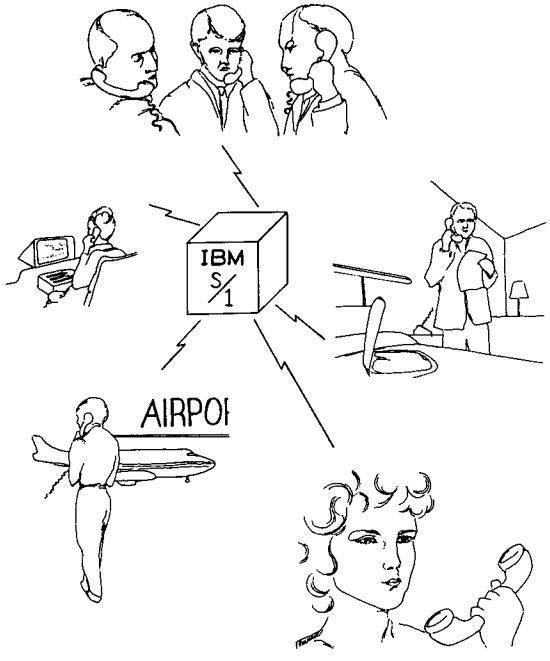

Fig. 1. Conceptual diagram of SFS. Users can send and receive messages using standard pushbutton telephones. Messages are stored in an IBM S/1 computer, which is connected to the telephone network and can be dialed from anywhere. Starting from the lower left and going clockwise, users can get or send messages from telephone booths, from their offices, from conference rooms (where several people can use SFS at once), or from motels and homes. A user's secretary can, with the secretary password, assist the principal via ADS.

made available to users in 1975. Four people could use it at once. From 1975–1981 about 750 IBM principals, mainly in the U.S., used various SFS prototypes in their daily work. For seven years we modified and enhanced prototypes on the basis of informal observations, interviews, studies of users and through helping users with their problems. These SFS prototypes ran on an IBM System/7 computer attached to VM370 for additional storage. A unique feature of the SFS project was the interdisciplinary nature of the research group, with emphasis upon behavioral expertise. (Half of the group of ten had PhDs in experimental psychology.)

We began work on a Series/1-based prototype in 1978. In September 1981, IBM announced this project as the *Audio Distribution System* (ADS). Each ADS can support up to 1000 principals (with up to ten people using it at once). In this paper we describe our work up to, and including, the ADS product. During 1982 we unplugged the SFS prototype and now use ADS in our work. Other companies, including AT&T, Electronic Communications Systems (ECS), and Wang, have announced audio systems which have some functions in common with ADS.

1.5 General Methodology

Our approach was fivefold:

(1) Our initial design relied upon our intuition and behavioral analyses of principals' work activities.

(2) From 1975–1981 we conducted informal field studies of how principals used SFS, with the view of learning what was needed to make SFS more useful, and then implementing these requirements, regardless of the difficulties.

(3) At the same time we investigated related issues in controlled laboratory experiments on composition [4, 5, 8, 9, 10]; listening [20]; and impression formation [26].

(4) We often simulated possible new versions of SFS in informal laboratory studies, sometimes with an experimenter sitting behind a curtain and pretending that he or she was SFS. At other times (1978–1980) we used the VM Simulator because it was easier to change the messages and user interface control flow on it than on the small IBM S/7 (or, later, S/1). This allowed us to simulate a user interface before an actual new one existed.

(5) During the same period we gave talks and live demonstrations to thousands of potential users, and received their reactions about what they believed should be included in SFS to make it useful as a noninteractive message system. Here we also began to understand better some necessary commercial factors, that is, the need for displaceable costs.

While working on this project, we followed (and recommend to others) several principles of system design [11]:

(1) Focus initially on important user characteristics.

(2) Make intended users part of the initial design team.

(3) Empirically measure, through simulations and prototype experiments, how people use the system, with the aim of modifying the user interface, training procedure, and reading material.

(4) Create a system architecture so that the user interface can be changed easily on the basis of empirical results. The user interface should be as easy to change as the documentation. This iterative design philosophy may seem expensive, but, with the present state of understanding of user interface design, it is the only way to ensure excellent systems. It is not just a trivial, expensive matter of "fine-tuning," but a basic design philosophy to be contrasted with other principled design philosophies [11].

(5) Put one person in charge of all aspects of usability—user interface, reading materials, training approach, hot line, etc.

Although these principles may seem intuitive, results show that they are not typically recommended or followed [11].

1.6 Limitations of the Data

The work reported here is mainly informal, and lacks the rigor of controlled laboratory experiments. Often, we studied a question only until we became convinced of a better alternative, then implemented and tested it. We publish

Table I. SFS Function Compared to Traditional Telephone-Answering Device Functions*

Recording	Filing System
+Voice activated	Selective retrieval
Insert	Statistics
Delete	Scan descriptions
Move, Copy	Able to group messages
Review	**Access**
Combine messages	Name-oriented
Erase selectively	+Remote
Audio markers	Automatic notification
+Messages from anyone	
	Other
	User profile
	+Dictation
Listening	Multiple simultaneous users
Sped speech	Call divert
Review (replay)	Usage analysis
Skip around	Automatic roll-over
Annotation capability	Password security
+Erase	++Message light
Selective erase	++Music when busy
	Growth
	Uses digital computer
Distribution	Integrate to PBX
One person	Integrate with text, pictures
Several people	Data entry and processing
Distribution lists	Combine with speech recognition
Timed transmit	User networks
Personal reminder	**Present Estimated Costs**
Status of messages	$200-2000/user total cost
Reply	$10/user/month

* SFS functions are listed in this table. The SFS functions which typical telephone-answering devices have are preceded by a + sign. Functions which telephone-answering devices have and SFS does not have are preceded by ++.

this work because perhaps no office system has been studied as extensively as SFS and because we would like to contribute to a beginning in the publication of such case histories.

2. SFS FUNCTIONS

The main functions of SFS are shown in Table I. They are summarized under composing (recording), listening, filing, and distributing documents. As described more fully in [10], SFS is voice activated, recording only when a user is speaking. While recording a message, a user can insert, delete, move, or copy speech signals. A user can attach audio markers, through key presses, to various parts of the document, which then provide listeners with clues to the document's format and structure and allow listeners to skip to a new paragraph, or listen only through a particular topic. While listening, a user can review, scan, or skip around in a message. A user can annotate a message, return it to the sender and/or send it to someone else also. A user can send a message to one person, to several people, or

to user defined distribution lists. A user can send personal reminders, requesting a telephone to ring at a designated time. A user can inquire whether his or her messages have been heard. A user can file and retrieve a message on the basis of the sender's name, date, or topic. Users send messages to people's names, not to their telephone numbers or other arbitrary designations. Each user has several telephone numbers at which SFS will call after new messages arrive. Each user has not only a personal password but also secretary, family, and guest passwords—each of which has limited, appropriate functions. A user can alter these and other items of personal profile information from pushbutton telephones anywhere. SFS can be used with coded information as well, for example in remote data entry or in service repair applications.

3. TERMINALS

3.1 Why Telephones?

Our behavioral analyses suggested that telephones should be the terminals for a principal support system rather than, for example, typewriters or CRTs. Telephones are located nearly everywhere and are available 24 hours per day, which is important because principals work in many places and at many different times of the day. Telephones were chosen because many people must be on a communication system in order for it to be useful; the fact that all principals have telephones makes it possible for this to happen just as soon as SFS is installed. New users can be immediately joined to SFS without having to cost justify new terminals or wait for delivery, wiring, and installation. Telephones were already present and cost justified for other purposes. Further, telephones, not video terminals, were the terminals principals were actually using.

The challenge was to map the many functions of SFS onto the 12-key pushbutton telephone so that SFS would be easy to learn and use. Laboratory development and marketing people were skeptical that we would ever be successful. For example, they would watch demonstrations, be impressed with the useful function, but volunteer that most people would never bother to learn the finger dances required. There was a strong opinion that a special purpose terminal with a different, labelled key for each of the 40–50 functions was needed, perhaps with a visual display. To this end, we verified that commercially available Touchamatic telephones could serve as a special purpose terminal [7]. (Touchamatic telephones are "one-button dialing" devices which contain a panel of buttons each of which when pressed ordinarily causes a different telephone number to be dialed.) We labelled each of the thirty separate keys and set each to execute a different SFS function.

We resisted this one-key-one-function approach for two reasons: the ubiquity of the telephone, allowing for remote, convenient use and its very low entry cost. We knew that the user interface problems would be much harder to solve using the 12-key pad, however.

System messages that are heard raise issues which do not come up with system messages that are read. A voice system requires pronouncing the names of symbols which ordinarily go unnamed. Should the #-key on the telephone be called the "number sign key," the "tic tac toe sign," the "hash mark," or the

"pound sign" key? Should the *-key be called the "asterisk key" or the "star"? We settled on the last alternative in each set after years of observing what other people called them and of trying various alternatives ourselves. Should a person be told to "enter," "dial," "type," or "keypress" his or her last name? We settled on the last alternative after trying the others as well. Should voice messages be of the form "To insert here, press the 1-key" or, alternatively, of the form "Press the 1-key to insert here"? Initially we composed all messages in the former way, since this first identified what a person wanted to do and then said how to do it. However, we gradually switched most of the messages to the latter format, which seemed to lead the less novice confusion. The alphabetic labels on the 12-key pad omit the letters Q and Z. When these letters appear in a user's name (e.g., Quinones, Zeheb), users keypress the 7 key for Q (PQRS) and the 9 key for Z (WXYZ).

3.2 Keypressing vs. Voice Commands

Why not voice commands? The state of the art of speech recognition over telephone lines was not good enough when we started. Errors would have been made in recognizing voice commands in the stream of other words a user was saying; such errors could rarely occur in recognizing pushbutton tones in the stream. Each user would probably need his or her own stored set of commands for speech recognition to work reliably. Results of controlled experiments satisfied us, however, that voice commands do not interfere with composing [9]. Over the years, users did not ask for voice commands, even though this would have reduced the inconvenience of using dial telephones when pushbutton telephones were not available.

3.3 Results

We learned over the years that the telephone was an excellent choice as a terminal, particularly as we improved ease of use (see Section 4). Field studies and interviews with potential customers indicated that they would be less disposed to put large groups of people on SFS if they had to cost justify terminals (in addition to the system itself). Principals used SFS from many different places and at all hours of the day and night. Our observations indicated that prospective users felt the need for hardcopy, but once they became actual users they rarely asked for this, or used the available facility to get it. Uhlig [27] has also noted the value of the telephone in being able to send a message whenever and from wherever one wants to. His project's solution, however, was to have people carry with them "very lightweight portable computer terminals" that attach to telephones so users could get and send printed or video messages.

4. USER INTERFACE

4.1 Multifunction Mapping

Modes. Our basic approach to the multifunction mapping problem was to organize SFS functions into command modes. Partitioning a command language into modes involves a trade-off. The benefits include simplification, reduced (human) memory load, and a helpful structure with which users could think about system

Simple ADS User Language Accomplishes Many Things

*R (ecord)	*T (ransmit)	*G (et)	*L (isten)
	N Gould		
	00 (Send it)		

Compose Messages	Receive Messages
Voice Activated	Select Messages
Review	Relisten
Insert	Reply
Start Over	Annotate
Make File Copy	
	Nonuser Capability
Send Messages	Receive from Nonusers
To One Person	
To Several People	Enhanced Dictation
To Distribution Lists	
Personal Reminders	
File Messages	

Fig. 2. Basic ADS commands are shown at the top, and the functions they accomplish are listed next. The command language is designed to minimize the number of commands required to accomplish the most frequently used SFS functions, as identified over the years by observation of people's usage patterns. Users compose messages by recording (*R), send messages by transmitting (*T), and receive messages by getting (*G), and listening (*L) to them.

functions. The potential costs are that users must keep track of which mode they are in, and they are limited in what they can do in that mode.

Initially we had 13 modes. This worked fine to identify the functions users preferred, but our field studies showed that this number was too many for novices' preferences. We gradually reduced the number to four basic modes. This approach has worked well. However, even with only a few modes, novices occasionally made mistakes, for example, in trying to send a message from the Listen mode instead of the Transmit mode.

As shown in Figure 2, the basic mode commands of ADS are now *R, *T, *G, and *L. There is also a Customize mode (*C). With these mode selection commands one can execute the basic functions shown in the lower part of Figure 2.

Within each mode there is specialized function. For example, as shown in Figure 3, in Transmit mode a user can Classify a message (C-key), send a message to a Distribution list (D-key), or to a person's Name (N-key). Inside each mode each key can stand for a different function. Users were taught to think about these intramode commands as prefixes or alternative selectors (see Section 4.2).

We made the mode approach easier to use by reducing needless distinctions, adding mnemonics, improving feedback messages, adding the concept of selector keys, developing partially and fully prompted user languages, and adding a help system. These were added and continually modified and improved on the basis of user requests, feedback, and simulations.

1-key	Press 1 Immediate Delivery, Quick-Ring Notification Special Notification Normal Notification	2 Time Delivery Enter Time Enter Date	
C-key	Press 1 Classify Messages Unclassified Internal Use Only Confidential Personal Personal and Confidential	2 Limit Access Unlimited Append and Forward Listen Only	3 Limit Retention Retain 180 Days Retain 90 Days Retain 60 Days Retain 30 Days Retain 2 Weeks
D-key 4-key 5-key N-key	Precedes Name of Distribution List Undefined Key Undefined Key Precedes Recipient's Name		
R-key	Press 1 Request Reply Reply Requested Specify Deadline Time Enter Time Enter Date Use Standard Deadline	2 Request Approval Approval Requested Specify Deadline Time Enter Time Enter Data Use Standard Deadline	
T-key	Press 1 Call Me when Received Specify Notify Time Enter Time Enter Date Use Standard Notify Time	2 Call Me when Action Taken Specify Notify Time Enter Time Enter Date Use Standard Notify Time	
W-key	Answers Who, What, Where Questions		
*-key	Changes Modes		
#-key	Activates Help System		

Fig. 3. Specialized function in Transmit mode.

Reduction of Some Distinctions. In initial versions of SFS it was thought necessary, for example, to have an "Edit" mode separate from a "Record" mode. Another example was having separate commands for an unformatted Listen mode and a formatted Listen mode. In the formatted Listen mode, beeps played out where audio format markers were located, whereas in the unformatted Listen mode, beeps did not play out [24]. Through empirical studies we learned that this and several other distinctions were not important to users, and we eliminated them.

For several years the SFS command language had a neutral command (the user pressed the *-key), which explicitly took users out of the mode they were in and put them in a neutral state. Figure 4 is an example scenario, as shown in an early (1977) Examples-of-Use card. We learned this neutral concept could be dropped by combining the * with the next command mode to be selected. For example, the user learned to press *R to record a message, rather than pressing * to leave a mode and then pressing R to activate the Record mode. Figure 5 shows an example scenario of this improvement. The important point here is that in both cases users pressed exactly the same keys—they just thought about it differently in the two cases. Whereas in the first case they thought of neutral

To Compose and Transmit a Message (with 1977 User Interface)

*	Return to Neutral
NEUTRAL	Confirming Message
71	Select Record Mode
RECORD	System Prompt
"John, I will be in Washington this week."	
*	Return to Neutral
NEUTRAL	Confirming Message
82	Select Transmit Mode
TRANSMIT	Confirming Message
5 Gould ##	Define Recipient
JOHN GOULD	Confirming Message
ADDED TO LIST	Confirming Message
00	Send Message
BEEP, BEEP	System is working
AUDIO SLATE TRANSMITTED	Confirming Message
*	Return to Neutral
NEUTRAL	Confirming Message

Fig. 4. Example of a user composing and transmitting a message with the 1977 user interface. Note the lack of alphabetic mnemonics and the separation of the * and mode commands. (Compare to Fig. 5 to see the improvement in the command language.)

To Compose a Message and Transmit It (Final User Interface)

*R
RECORD
"Bill, what is the part number that. . ."
*T
TRANSMIT
N
ENTER RECIPIENT'S NAME
notz
BILL NOTZ
00
MESSAGE TRANSMITTED TO BILL NOTZ

Fig. 5. Example of a user composing a message and sending it to Bill Notz. (This may be compared with an earlier version of the user interface shown in Fig. 4.)

and mode selection as two separate commands, in the second case they thought of mode selection as only one command (in which they pressed two keys).

To be able to make these changes consistently and to evaluate them quickly meant that we had to introduce them simultaneously in the user interface, in reading materials, and in the training procedure. A fragmented development procedure with no single focus for usability would not allow this easy coordination. In eliminating other needless distinctions, we dropped much general, generic terminology (e.g., "items," "parameters," "lists").

Mnemonics. Initially, all commands were numbers, that is, a user pressed the number 72 to record a message. Gradually, we began using alphabetic mnemonics

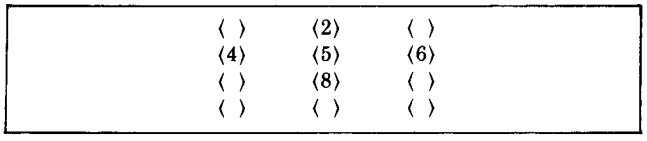

Fig. 6. Example of a spatial mnemonic, or memory aid, in the Listen mode. Imagine that your forefinger pivots on the 5-key. To go to the beginning, or top, of a message, move your finger up and press the 2-key. To go to the end, or bottom, of a message, move your finger down and press the 8-key. To skip back a little in the message, move your finger to the left and press the 4-key. To skip ahead a little in the message, move your finger to the right and press the 6-key.

(memory aids). As shown in Figure 2, users pressed *R to record a message, *L to listen to it, and *T to transmit it. We were initially worried that mixing alphabetically and numerically represented commands might be confusing to users, especially in print. This was not the case, however.

We also incorporated spatial mnemonics, and applied consistency in this regard across modes [10]. An example of spatial mnemonics is illustrated in Figure 6. SFS allows a person to skip around within a message. Imagine a person's forefinger resting on the 5 key. To go to the beginning or *top* of a message, a user presses 5 and then moves his finger *up* and presses the 2 key. To go to the end or *bottom* of the message, a user presses 5 and then moves his finger *down* and presses the 8 key. An analogous arrangement was provided for selecting messages in queues in the Get mode.

Feedback Messages. Years of empirical studies were aimed at improving SFS prompting, feedback, helping, and confirming messages. (Message consistency was of course incorporated. Consistency was not the problem that it often is, since one person was in control of all messages.) A message played out after every user action. Sometimes a small study would be directed at the exact wording of a single message. On the VM simulator we changed the message set literally thousands of times. In ADS there are about 565 messages. Customers can reword any of these voice messages—and in any language.

Selector Keys. We observed that Atari was able to design a system of several hundred video games which children (and adults) play with no instruction. One principle is that a player simply holds down a switch until he or she reaches the desired game. We incorporated this concept into SFS. For example, a user might want to classify a message he or she is sending as company confidential, and, in addition, limit in other ways what a recipient can do with it. Thus, as shown in Figure 7, the user would press the C-key (mnemonic for message Control) and then answer the questions SFS asks until the right alternatives are selected.

4.2 Partially Prompted Language

Gradually, we moved away from an entirely self-produced command language to one that under some circumstances presented users with audio menus (no more than three choices per item). This relieved some of the user's memory burden

To Classify a Message and to Restrict What Can Be Done With It

```
*T
TRANSMIT
C
PRESS 1 CLASSIFY MESSAGE, 2 LIMIT ACCESS, 3 LIMIT RETENTION
1
UNCLASSIFIED
PRESS 1 CHANGE CLASSIFICATION, 2 DON'T CHANGE CLASSIFICATION
1
INTERNAL USE ONLY
PRESS 1 CHANGE CLASSIFICATION, 2 DON'T CHANGE CLASSIFICATION
1
CONFIDENTIAL
PRESS 1 CHANGE CLASSIFICATION, 2 DON'T CHANGE CLASSIFICATION
2
CONFIDENTIAL
C
PRESS 1 CLASSIFY MESSAGE, 2 LIMIT ACCESS, 3 LIMIT RETENTION
2
PRESS 1 UNLIMITED, 2 APPEND AND FORWARD, 3 LISTEN ONLY
3
LISTEN ONLY
N
ENTER RECIPIENT'S NAME
Boies
STEPHEN BOIES
00
MESSAGE TRANSMITTED TO STEPHEN BOIES
```

Fig. 7. Example of a person using a selector key to send a confidential message that will allow Stephen Boies to only listen to the message (and not alter it or send it to someone else).

and allowed another level of function without increasing the number of key presses required.

4.3 Entirely Prompted User Language

As SFS became increasingly easier to learn—through years of observation of users and resulting improvements in the user interface, training approach, and training material—we decided to try to design the ultimate in ease of learning, namely, a user interface requiring no user training whatsoever. If we fell short of this goal, we would at least know how close we were to the ideal. One approach we took, beginning in 1979, was to design and experimentally study through simulations an entirely prompted SFS language containing only the basic subset of SFS functions shown at the bottom of Figure 2. It made command selection a recognition task rather than a recall task. Figure 8 shows an example scenario with the prompted language. All prompts are context sensitive. People were never prompted for things they could not do or that were inappropriate (e.g., prompted to get a new message when there was none). Futher, for the first time we began to use system messages that did not stand alone but relied on context, that is, contained demonstrative pronouns.

To Compose and Transmit a Message With the Fully Prompted Language

```
PRESS *G GET A MESSAGE, *R RECORD A MESSAGE
*R
  RECORD
  "Nancy, are you coming to. . ."?
  *
PRESS *L LISTEN, *T TRANSMIT, *R RECORD
*T
  TRANSMIT
  PRESS N TO SPECIFY A NAME
  N
  ENTER RECIPIENT'S NAME
  Gris
  NANCY GRISCHKOWSKY
  PRESS 00 TRANSMIT MESSAGE, N SPECIFY ANOTHER NAME BEFORE TRANSMIT-
    TING
  00
  MESSAGE TRANSMITTED TO NANCY GRISCHKOWSKY
  PRESS *R RECORD, *G GET MESSAGE, *L LISTEN, *T TRANSMIT TO ADDITIONAL
    PEOPLE
```

Fig. 8. Example of a person using the fully prompted language to compose a message and send it to Nancy Grischkowsky. (Compare this to Fig. 2.)

In informal experiments involving about 100 new users we found that most novices could use this prompted language with no training. In these experiments we simply told participants that SFS allowed them to send messages to people and to receive messages from people, and that they should try to use it to carry out some test problems which we gave them. When problems did occur, they usually involved the three concepts mentioned in "Induction of Concepts From Examples" in Section 5.3.

4.4 Three ADS Languages

Like others who design systems, we felt that novices might need a "menu" system (the entirely prompted language), but that experienced users would want a non-menu system. As a result, ADS has both. ADS has a fully prompted language containing mainly the basic functions shown in Figure 2. The motivation here was to give users a simple, basic system with a very easy to learn command language. Here, users are always given a menu of choices (three or less) after each action. The drawbacks to any fully prompted language are that (a) users must take the time to go through the menus, and (b) learning a menu-oriented language may not transfer well to learning a non-menu oriented language—which is what we believed our full function language should be and we wanted users to eventually migrate to it. We addressed the first problem by letting users interrupt the audio menus or prompts any time they wished. We addressed the potential transfer of learning problem by offering a second basic command language for new users, functionally equivalent to the fully prompted one. (Users have the option of which language they want to use.) This language appears to the user to be exactly like the full function language (in prompts and feedback), except that it contains less function. (Here, if a user presses a key which would trigger

To Check if Recipient Has Listened To Your Message

```
*G
  NEW MESSAGES
    STEPHEN BOIES
  55
  OLD MESSAGES
    JOHN GOULD
  55
  OUTBOUND MESSAGES
    NANCY GRISCHKOWSKY
    JIM SCHOONARD
    STEPHEN BOIES
  PRESS *L TO LISTEN TO THE MESSAGE TO NANCY GRISCHKOWSKY
  W
  SENT TODAY SEPTEMBER 26 AT 6:32 p.m.
  W
  TO NANCY GRISCHKOWSKY. NOT RECEIVED.
```

Fig. 9. Example of a user checking whether Nancy Grischkowksy has listened to a message the user sent to her. Note that after the user enters his Outbound Message Box, he presses the W-key to learn what has happened to the message.

advanced function in the full function language, he or she is told that the key is undefined and is then prompted as to what is allowable in that context.) Both are basic languages and are true subsets of the full function language, to maximize transfer of learning to the language with additional functions. The full function language is the third language, and is a combination of prompted and self-produced commands. (Self-produced commands are those which a user emits without being given a menu of choices.) As already mentioned, within each mode of the full function language each key generally has a specialized function (see Figure 3 for the Transmit mode; see the *IBM ADS Subscriber's Guide* [15] for a complete description of all ADS function).

4.5 Help System

To help novices get started and to help experienced users learn additional functions, we implemented a help system on ADS. We have been told by ADS customers that it contributes significantly to ease of learning the full function language, and that their users require little training and little documentation (a telephone template and a wallet-sized card containing the basic commands).

Help Key. The important concept here is the designation of the #-key as a help key. If a user is stuck, he or she can press the #-key and SFS plays out the user's alternatives in that context.

Automatic Help. In addition, if the user does nothing for many seconds, then ADS plays a prompt to the user. The length of these pauses is variable, being tuned to what the user is trying to do.

W-Key. With the W-key (i.e., the 9-key) a user can get answers to the who, what, when questions, that is, who sent the message, when was it sent, who else was it sent to, who else already listened to it and when (see Figure 9).

Other Significant User Interface Features. Table II provides a list of SFS features which users have described as "user friendly." For example, users key-in

Table II. Some SFS Features that Are Called User Friendly by Users

Name oriented
Voice activated
Mnemonic
Multiple user languages
Adaptive system
Context sensitive system messages
Safeguards against possible user errors
Anticipates likely user errors
Help key
W-key
Automatic system message playout when user is stuck
Forgiving system (e.g., dates, times)
Prompted customize mode
Multiple passwords
Optional notification when recipient listens to message

their last name rather than a telephone number or other identification code. Thus, SFS is telephone-independent, and users can send messages to others without worrying about which telephone number to send them to. As a result of years of study, SFS will anticipate likely user errors; for instance, if a user presses a zero instead of the letter "O" while keying a person's name, SFS will tell the user that. SFS safeguards against possible user errors (e.g., if a user leaves Transmit mode without actually transmitting a message, SFS will ask the user whether he or she forgot to transmit that message). SFS can notify a user when a recipient has listened to (and/or replied to) an important message, which frees the user from anxiously calling SFS to hear a reply.

5. TRAINING AND LEARNING

There was no office system in 1973 which principals themselves used. Thus, our early attempts at training had to be both motivational and instructional. We had the general belief that principals did not want to spend much time being trained, nor did they want to read much documentation. Designing a powerful audio system (using only the telephone as a terminal) to meet these requirements seemed much more difficult than designing an electronic mail video system, with large keyboards, screens, printers, and a decade of experience in using such terminals and editors.

5.1 Initial Tutorial Approach

At the beginning (1975), small groups of new users (5 people) were taught SFS through an audio-visual orientation lecture and a subsequent followup lecture. Users were given printed material to take back to their offices, including telephone templates. Some of this material was instructional, via manuals on how to use SFS, and some was motivational, indicating how SFS could help them in their everyday work.

This approach did not work well. People did not transfer what they learned in the lecture to actual use of SFS in their offices. Second, principals confirmed our beliefs about their desire for minimal training and documentation.

5.2 Training with Examples

We addressed this lack of transfer in two ways. First, we installed four telephones in a small room and asked people to actually practice using SFS for most of the first hour of training. Second, we designed an Examples-of-Use card which contained printed examples of the most frequently used SFS scenarios, such as those shown in Figure 5. Novices would follow these during their hour of training and then take the card back to their offices.

Hands-On Training. The approach of having new users practice using SFS during most of the first hour of training worked well. Seeing other participants achieve success and/or make mistakes was helpful to novices and reduced their anxiety. We were able to learn of problems novices were having with the user interface (e.g., pressing zero instead of the letter "Oh" when keying a last name such as Owens) and with SFS concepts. We mixed secretaries and principals in the same group. Although we were sometimes warned not to do this, it worked well.

The first hour of training provided many insights into how to modify the Examples-of-Use card, our tutorial about SFS, and ultimately the user interface itself. For example, we modified the layout and order of examples based upon how people searched for and used them.

Users came back in about two weeks for a second hour of training in advanced SFS functions. This second hour was dropped after about one year because of users' lack of interest and because we increasingly learned what the most typically used functions were and covered them in the first hour.

Examples-of-Use Card. By 1977, our Examples-of-Use card and a telephone template were the only documentation we gave most users. The Examples-of-Use card was a 4-fold fan card that conveniently fit in a suit-coat pocket. It contained about 20 examples of SFS use, plus one page of additional information. By combining the card with hands-on training, the transfer problem, that of novices leaving the training session and being actually able and willing to use SFS on their own, was substantially solved.

Principals did indeed use the Examples-of-Use card in everyday business. In individual interviews conducted by John Conti with about twenty high-level principals who had used SFS for a year or more, each one reported using the card in everyday business. As a further test, each was asked where his card was. To assess the accuracy of recall, each principal was then asked to show the interviewer where the card was, and each one was successful at finding it quickly.

Gradually, the Examples-of-Use approach became the way we thought about adding new function. That is, it became part of the design process. We would work out new scenarios and rationalize them with already existing examples. If a new scenario appeared to be cumbersome, we then redesigned the relevant part of the command language, which had already been established. Ultimately, about 100 Examples-of-Use served as an initial functional specification in 1979 for what became IBM's ADS.

5.3 Results

The important point here is that SFS was initially usable, requiring much less time than that required to learn today's text editors (e.g., [17]). But we wanted to

do much better. By 1978 novices typically learned to use SFS with about one hour of training. Grishkowsky [12] provided recommendations for such training. By 1982 the need for this much training was reduced. ADS customers now indicate that formal training classes are often not needed. New users are simply told that they have been added to ADS, and are given documentation, sometimes consisting of only a wallet-sized card or a telephone template. If a learning problem arises, it is sometimes handled by using the third-party add-on facility of a PBX. ADS is added to a telephone conversation between a novice and the system supervisor. Either party can control ADS with pushbuttons.

This is a far cry from usability results in 1975. The biggest gains over seven years have been in the improvement of the user interface, so that now novices can experience immediate success, and in the careful integration of the help system, the minimal printed material, and the improvements in the user interface. Clearly, any development process that separates education, training materials, and interface design could not have worked as well. Recent reports show that even with the improved computer-based electronic mail systems of today, users require 1 to 2 hours [25], at least several hours [27], about one day [21], or more [18] of introductory training to do basic functions. In addition, users are also given much documentation. Even after training, a significant fraction of office principals do not *directly* use electronic mail systems, but instead use them through intermediaries [2, 13]. (Usability should be little affected by the number of users on a network.)

Induction of Concepts From Examples. We designed the Examples-of-Use card with the hope that users would inductively derive SFS from it. In general, this approach worked very well. However, it fell a little short of complete success for some first-time users with no training. When some of these people made a mistake in following the examples, they were not always able to infer the proper recovery action. They could press the *-key to escape from the mode they were in, or start over, but this fell short of a magic "undo key" which they would have preferred.

A few concepts seemed to be the sticklers. We learned that these had to be explained to some users. We found that a 8-page comic book (written by John Conti) together with a 1- to 2-page printed description was satisfactory.

One sticky concept was that of the audio slate, which is a temporary work or storage area where the current message being composed or listened to resides. Novices did not need to be told about this until they made a mistake. Often, however, an understanding of why this temporary storage area existed was required for a novice to recover. Involved here was the concept that two copies of the same message existed when a person was listening to a message: one on the audio slate and the other in the user's Old Message Box.

Another difficult concept was how to retrieve and relisten to a message which a user had listened to somewhat earlier in that session. This involved the concept of the audio slate, as well as the concept that this message had been transferred from the user's New Message Box to his or her Old Message Box as a result of being listened to. The following example illustrates what makes this problem particularly complicated for the novice. Assume the novice has four messages called A, B, C, and D in his or her New Message Box; and two messages called E

and F in his or her Old Message Box. Assume that the novice has listened to message A. Now the Old Message Box contains A, E, and F, in the order last in, first out. Assume the novice is now recording a message of his own. He wants to relisten to message A. To do this he must get into the Old Message Box, which requires skipping over the remaining new messages. Further, the novice wonders what will happen to his partially recorded new message.

The third difficult concept was that of modes. The notion that specific actions could only be done within specific modes (e.g., send a message from Transmit mode only) had to sometimes be explained to a first-time user.

6. SYSTEM USE

6.1 What Was SFS Used For?

Principals could use SFS to replace or displace five types of communications: (1) dictated documents intended to be typed and read; (2) "yellow slip" messages; (3) memos and letters; (4) interactive telephone conversations; and (5) face-to-face conversations. Results show that SFS was used mainly as a voice message system. New users typically used it when they were unable to reach a person on the telephone, and to respond to SFS messages that others sent them. However, they always left content messages, rather than simply asking that person to call back. A typical user then used SFS to gradually replace some memos, letters, and interactive conversations.

This usage pattern points at displaceable cost savings, separate from any claim about improved principal productivity. Three normal operating costs can be reduced: long-distance telephone costs, because ADS customers can use a single 800-number; typing costs due to fewer written memos; and human telephone message handling costs due to fewer missed calls. We estimate the costs savings here to be about $90 per user per month in a typical office. This cost savings is based upon the assumption of 6 fewer long-distance calls ($15 savings), 8 fewer memos ($50 savings), and 100 fewer telephone messages ($25 savings) per user per month. Assuming that ADS costs about $10 to $15 per user per month, the net cost saving is about $75 to $80 per user per month. If these estimates are approximately right, ADS pays for itself in less than one year, which is very unusual for a computer system.

People did not much use SFS for dictation. This was true even when users were told that they would receive a typed version within an hour or so after they dictated their letters (and their own secretaries would not have to do the typing). (Most did not use any other dictation system either). Our laboratory studies began to show the superiority of noninteractive speaking over other methods of composition. (With speaking an author assumes the recipient will listen to the resulting letter, whereas with dictation an author assumes the recipient will read the letter.) Speaking was faster than dictating and writing [4, 5, 7] and using text-editors [9], with no reduction in quality [4]. People composed spoken messages in laboratory experiments [7] about four times faster than did users of a video electronic mail system (EIES), who composed at a rate of 6 to 8 words per minute (wpm) [14]. Actual users of SFS compose even faster, typically at about 120 wpm [23]. People found speaking easier because listeners are more tolerant of mistakes,

poor syntax, false starts, and inexact wording than are readers, and no spelling is involved. In laboratory experiments [7] new SFS users did not experience the same anxieties as did people new to dictation [5].

Users typically sent short messages on SFS (often less than one minute long [23]). Laboratory experiments showed that the pause-compressed speech on SFS was at least as comprehensible as extemporaneous speech, based upon tests of subjective understanding, listenability, and comprehension [20]. The important listening features turned out to be those valuable for listening to short messages (e.g., automatic pause compression). Thomas [26] conducted experiments suggesting that an automatic pause-compression facility would lead listeners to form more positive impressions of callers than if callers were heard to talk slowly with many pauses. Audio quality and reliability had to be, and now are, as good as the telephone system's. SFS surpasses the telephone system in some matters of convenience that are important to users, for example, SFS answers after one ring; (almost always) has no busy signals when a user calls it; and plays out messages quickly when it calls a user. Compared to electronic video mail systems, composing and listening on SFS appear to be more efficient and easier.

6.2 Noninteractive Communication

The value of noninteractive SFS communication has exceeded our expectations of a decade ago. If fulfills basic communication needs of principals by giving them the ability to communicate with people when they want to, to communicate across time zones, to communicate without knowing where a recipient is, to communicate without need for irrelevant conversation, to communicate under a variety of circumstances (which otherwise would be prohibitive), to formulate a reply without the pressures of interactive communication, to eliminate telephone roulette, and to control interruptions.

We began to realize that a significant percentage of all interactive telephone calls (and some interactive conversations) were "one way," and thus were easily displaced by SFS. Our initial view of the usefulness of noninteractive communication was colored by the technology then available. But, relatively rapid noninteractive communication through SFS (or printed electronic mail systems with terminals convenient to principals see [21] and [22] for reviews) has altered our view.

Hardcopy Record and Audit Trail. Before people became users of SFS, they generally believed that a hardcopy record would be essential for them to use SFS. Sometimes this belief was based upon the suspicion that they would rather read their SFS messages than listen to them. At other times it was based upon the tradition of having a printed historical record from noninteractive communication. Once people became users, however, this perceived need disappeared. From 1976–77 users could have any of their audio documents typed by sending them to a user called "typist," but they rarely did so. For the most part, users do not even retrieve an audio message which they filed earlier [23], although some want the possibility of doing so.

A related perceived need was for an audit trail indicating the history of each message, who listened to it, and when. We have found that users do want to know the current status of a just sent message (e.g., has the recipient listened to

it yet, or taken action?) We have incorporated a detailed audio audit trail into ADS, and we await the results of its usage.

6.3 Content Messages

Principals did send content messages. In seven years of field studies and interviews we did not learn of a single example of a user sending a message which only asked the recipient to call back. With experience, users began to use SFS as an alternative communication means, often using it rather than calling a person directly. For example, users sometimes suggested in conversation that a person should send them a SFS message, rather than call them, to resolve some matter. Users did not receive junk mail from other users, as happens with video electronic mail systems (e.g., [14]). The tone of the messages we have heard was almost always businesslike. We were initially concerned that principals might not use SFS for important content messages because they might be afraid that their emotions would show through in their voices. However, users have not expressed this concern and have themselves heard only a few emotionally intoned messages.

Many factors contributed to principals sending content messages.

(1) The composing facility was easy to learn and use, as already discussed.

(2) SFS contained useful functions, and users perceived these as making a positive contribution to their worklives. Table III summarizes some of the positive statements made by users about SFS. Users believe SFS saves them time, makes them more effective, and happier.

In a field study conducted by Epstein [1] on 80 principals in a corporate headquarters location, most users were satisfied with SFS, felt they were more productive with it, and felt that they would be unhappy if it were taken away (see Table IV). The same principals indicated that their satisfaction and productivity would improve more if up to five more key people of their choice were added to the system. One of the most effective ways of learning about the value of a discretionary use system is to take it away from people. Seventy-five per cent of these principals reported they would be dissatisified if they were removed from SFS—which is perhaps the ultimate test of user satisfaction. We found that, over the years, if SFS was unavailable for a short time, even in the middle of the night, at least one user would complain to us the next day. From 1977–1981 we had a waiting list of several hundred people who wanted to be on SFS.

Electronic mail (video or printed media) systems also report increased principal happiness, effectiveness, and productivity. Uhlig [27] reports that users communicate more frequently and more effectively, and that they like the ability to communicate whenever and from wherever they want. A survey of 210 users of another electronic mail system (Darcom) showed that, as a result of using it, over half the managers and professionals felt they were more productive and had better long distance communication [21]. Most managers, but not professionals or secretaries, felt they had more flexibility in their working hours as a result of using an electronic mail system.

(3) Principals sent content messages because they quickly developed confidence that recipients would listen to their messages. Several factors contributed to this confidence. First, SFS was reliable. Almost perfect reliability was necessary

Table III. What Users Volunteer that SFS Does For Them

Saves My Time
5:1 faster composition time
Fewer handwritten memos
Less proofreading
Less telephone roulette
Reduction in looking-up telephone numbers
Able to and does leave content messages
Able to get content messages
More control over working environment
Reduced need to interrupt on-going activities to communicate with people
Less need to organize activities around other people's (presumed) schedules
Adds useful time to workday
More powerful, flexible dictation
Makes Me More Effective
Solves real problems
Does not interrupt regular office procedures
More timely information
Faster communications
Reduces errors in messages
Provides better telephone coverage
Fewer missed telephone calls
Controls interruptions
No need to hunt around for people
Solves time zone communication problems
No need to locate people to communicate effectively
Able to send messages from any place at any time
Able to get messages at any place at any time
Unique advantages in multinational companies
Makes Me Happier
Satisfies major annoyances
Reduces interpersonal communication problems
Less need to come to office during off hours

not only during transmission but also while composing (so that a half-composed message would not be lost at this stage). Second, the message was available to recipients immediately after it was sent, unlike some other electronic mail systems. Third, users addressed a message with a recipient's own name, rather than with the recipient's telephone number or some other arbitrary number. This gave users confidence that the right person would listen to the message (rather than just anyone at that telephone number). Fourth, users generally expected recipients to listen to their messages soon after they sent them, and this usually happened. If a message was sent in the morning, the median amount of time until a recipient listened to it was 4.3 hours [23]. Recipients usually called SFS, but if they did not, and they had new messages, then SFS called them. Fifth, users could easily check whether a recipient had listened to a message. Figure 9 shows an example of this. Users could also be automatically notified when a recipient had listened to a message. Sixth, users were given some *control* over when the recipient's phone would ring. They could make a recipient's telephone ring

Table IV. Results of a Survey of User Satisfaction*

User Statisfaction (in Percent)	
Very dissatisfied	1
Dissatisfied	0
Neutral	29
Satisfied	58
Very satisfied	9
No response	4
How Users Would Feel if SFS were Taken Away (in Percent)	
Satisfied	5
Neutral	19
Mildly dissatisfied	28
Somewhat dissatisfied	19
Very dissatisfied	28
No response	2

Estimated Productivity Gains (in Percent)	*Users Estimating Productivity Gains Due to SFS (in Percent)*	*Users' Estimates of Their Productivity Gains if Five People of Their Choice Were Added to SFS (in Percent)*
Negative	5	1
Zero	35	15
1–3	24	24
4–10	20	26
11–20	11	23
Over 20	3	1
No response	1	1

* Results from a survey of 80 principals who used SFS in a corporate office for 1 to 4 months during a field study [1]. The entries are the percentages of the 80 users who selected various alternative answers.

immediately, if desired, or at any other time. Seventh, users wanted SFS to be available 24 hours a day, and it was.

(4) Users sent content messages because SFS was perceived as being secure. Users frequently sent sensitive messages, for instance about personnel matters. We learned from our demonstrations to thousands of potential users that they were more concerned about security when they were potential users than when they became actual users. Security included password features, message classification features, and name-orientation. The password system on SFS was typical of the security on most computer systems, but users were given multiple passwords, each of which had a different power associated with it. For example, the secretary password allows one's secretary to obtain the names of people sending a principal new messages, but not to listen to the messages' content. The message classification system lets a user tag a message, for example, as personal or confidential. In addition, users could limit what a recipient could do with a message, that is, not alter it.

(5) Users sent content messages because SFS had to be—and was—sufficiently different from traditional telephone answering devices so that users would (a) like SFS and (b) leave content messages. One reason many callers do not leave *any* message (content or otherwise) on traditional telephone answering devices is

that they *expect* to talk with the person directly. With SFS the "caller" has already made the decision not to talk directly with the other person. Traditional telephone answering devices have only a few SFS functions, as shown in Table I.

(6) We, as systems administrators and trainers, were supportive of user concerns.

(7) Users sent content messages because a top-down sociology set in, whereby senior managers sent messages to people reporting to them, thereby encouraging them to use SFS similarly (see also [23]). (It was also possible that workers and professionals would be the first users of SFS, and then convince their managers of its value, i.e., "bottom-up" sociology.) In our early field studies we let the senior manager of a new group of users assign those who should be on SFS. He or she would assign as many new users as we would allow. This, however, did not work out as well as assigning a subset of users initially and then letting the net grow. The latter approach insured that a higher percentage of users would opt to take advantage of SFS.

6.4 Direct Use

Principals used SFS directly. We did not find a single example where principals asked a secretary or other support person to send a message for them. SFS was not viewed by users as a computer but, rather, as a tool to help them communicate. This is in contrast to the findings that some principals use electronic video mail systems by having their secretaries send the messages for them [2, 13].

7. GENERAL CONCLUSIONS

A voice store and forward system, as implemented in SFS, contains useful functions, solves real user problems, leads to user satisfaction and to the user perception of increased productivity, and is relatively easy to learn. New users used it when they were unable to reach a person on the telephone, or to respond to messages that others sent them. They used SFS directly, and always sent content messages. A typical user gradually used SFS in place of some written memos and interactive conversations.

A second set of conclusions has to do with methodology. We developed a process which worked successfully in identifying useful systems functions and in making these functions easy to learn and use. This process includes four critical steps:

(1) early focus upon the characteristics and needs of the intended user population (in this case, office principals);

(2) participative design in which the intended users become part of the design team;

(3) empirical and experimental measurement of how simulated and early prototypes are actually used;

(4) iterative design during which the user interface, training procedures, and reading material are modified based upon earlier and on-going measurements, and are then measured again.

This design philosophy is not an expensive, unprincipled fine-tuning, but is a principled approach which is necessary if progress toward significantly easier to

learn and more useful systems is to be achieved. While we thought we had an excellent system several years ago, we concede that ADS is better today as a result of continual iterations of this methodology. Indeed, we continue to use it in our research on new office systems.

ACKNOWLEDGMENTS

This paper does not describe just the authors' work. SFS was a team effort. Stephen J. Boies was the leader throughout. The other group members who contributed significantly over several years are John Gould, Nancy Grischkowsky, Bill Notz, John Richards, and David Zeheb. The authorship reflects who wrote the paper.

We thank Mary Beth Rosson, Bob Mack, Jim Schoonard, Nancy Grischkowsky, and Clayton Lewis for commenting on drafts of this manuscript, and two anonymous reviewers for their thoughtful comments on the manuscript originally submitted.

REFERENCES

1. EPSTEIN, A. A user survey of SFS. Personal communication, 1981.
2. GOODWIN, N.C., AND HOSMER, S.W. A user-oriented evaluation of computer-aided message handling. MME Final Rep. volume VI, part 1, MTR 3920, Mitre Corp., Bedford, Mass., 1980.
3. GOULD, J.D. An experimental study of writing, dictating, and speaking. In *Attention and Performance VII.* J. Requin, Ed., Lawrence Erlbaum Associates, Hillsdale, N.J., 1978, pp. 299–319.
4. GOULD, J.D., AND BOIES, S.J. How authors think about their writing, dictating, and speaking. *Hum. Factors 20* (1978), 495–505.
5. GOULD, J.D. How experts dictate. *J. Experimental Psychol. Hum. Perception Perform. 4,* 4(1978), 648–661.
6. GOULD, J.D., AND BOIES, S.J. Writing, dictating, and speaking letters. *Science 201* (1978), 1145–1147.
7. GOULD, J.D. Writing and speaking letters and messages. *Int. J. Man Mach. Stud. 16* (1982), 147–171.
8. GOULD, J.D. Experiments on composing letters: some facts, some myths, and some observations. In *Cognitive Process in Writing.* L. Gregg and I. Steinberg, Eds., Lawrence Erlbaum and Associates, Hillsdale, N.J., 1980, pp. 98–127.
9. GOULD, J.D. Composing letters with computer-based text editors. *Hum. Factors 23* (1981), 593–606.
10. GOULD, J.D., AND BOIES, S.J. Speech filing—an office system for principals. IBM Res. Rep. RC-9769, Dec. 1982.
11. GOULD, J.D., AND LEWIS, C. H. Designing for usability—key principles and what designers think about them. In preparation, 1983.
12. GRISCHKOWKSY, N.L. Training users on the Audio Distribution System. IBM Res. Rep. RA-136, 1981.
13. HERSH, H.M. Electronic mail usage analysis. In *Proc. Human Factors in Computer Systems.* Sponsored by the Institute for Computer Sciences and Technology, the NBS, and the Washington Chapter of the ACM, 1982, pp. 278–280.
14. HILTZ, S.R., AND TUROFF, M. The evolution of user behavior in a computerized conferencing system. *Commun. ACM 24,* 11 (Nov. 1981), 739–751.
15. IBM AUDIO DISTRIBUTION SYSTEM SUBSCRIBER'S GUIDE. SC34-0400-1. (Can be obtained from IBM, 4111 Northside Parkway N. W., Box 2150, Atlanta, GA 30056).
16. KLEMMER, E.T., AND SNYDER, F.W. Measurement of time spent communicating. *J. Commun. 22,* (1972), 142–158.

17. MACK, R., LEWIS, C., AND CARROLL, J. Learning to use word processors: problems and prospects. *ACM Trans. Office Inf. Syst. 1*, 3 (July 1983), 254–272.
18. MILLER, D.G. Military message experiment—final training report. Mitre Tech. Rep. MTR 3919, Mitre Corp., Bedford, Mass., 1980.
19. MINTZBERG, H. *The Nature of Managerial Work.* Harper and Row, New York, 1973.
20. NIX, D. Two experiments on the comprehensibility of pause-depleted speech. IBM Res. Rep. RC-6305, 1976.
21. PANKO, R.R. The EMS revolution (a survey of electronic message systems). *Computerworld* (Aug. 25, 1980), 1–12.
22. RICE, R.E. The impacts of computer-mediated organizational and interpersonal communication. In *Annual Review of Information Science and Technology, 15*, M. Williams, Ed., Kowledge Industry Publications, White Plains, N.Y., 1981, pp. 221–249.
23. RICHARDS, JOHN T. The IBM Research Speech Filing System: analysis of the System 7/S 370 Prototype. IBM Res. Rep. RC-9114, 1981.
24. SCHOONARD, J.W., AND BOIES, S.J. *Speech Filing System Reference Manual.* Unpublished manuscript, 1975. (Can be obtained from the authors at the IBM Research Center, Yorktown Heights, NY 10598).
25. TAYLOR, J. Evaluation of the terminals for managers (TFM) program at Stanford University. A report from the Center for Information Technology, Stanford Univ., Stanford, CA 94305, 1981.
26. THOMAS, J.C. Office communications studies: I. Effects of communication behavior on the perception of described persons. IBM Res. Rep. RC-7572, 1979.
27. UHLIG, R.P. Human factors in computer message systems. *Datamation 23*, 5 (1977), 120–126.

Received January 1983; revised April 1983; accepted May 1983