FACULTY OF ENGINEERING AND APPLIED SCIENCE

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

The Design and Evaluation of an Intelligent Front End Interface to an Electronic Mailbox System with the Hearing and Hearing Impaired

by

Keith Ian Watson

A thesis submitted for the degree of
Master of Philosophy

UNIVERSITY OF SOUTHAMPTON
1986

ACKNOWLEDGEMENTS

There are a number of individuals and institutions who I would like to thank for the invaluable assistance that they have provided during the course of the project.

Firstly I would like to thank Dr. A.C. Downton and Mr. J.A. Pickering for their supervision, particularly for their guidance, constructive criticisms and helpful comments, throughout the project.

I would particularly like to thank Dr. Hani Dabbagh, whose comments and discussion were of great value, and all the other members of the Man-Machine Systems group, during the period from October 1981 to July 1984, for their comments, advice and regular discussion. I would also like to thank the Electronics and Information Engineering Department for providing administrative help and assistance.

I am particularly grateful to Dr. S.M. Lewis, of the Faculty of Mathematical Studies, for her considerable help with the statistical theory involved in the experimental design and to Mr. R.G. Baker, now of the Department of Adult Education, for his assistance in the design of the questionnaire and his psychological expertise.

I am indebted to the people who gave unselfishly of their time and effort to participate in the evaluation. I am especially grateful to all the hearing impaired people who have given me encouragement and advice. I hope my work can be of particular assistance to them, many of whom are now my good friends. I would also like to thank the RNID for allowing me to conduct some of the experiments at their headquarters and to Mr. B. Judson, the system manager of the RNID mailbox group on Telecom Gold, for his comments and assistance.

I am grateful to the Science and Engineering Research Council for funding my research under grant award number 83/308125.

I would especially like to thank my parents, my sister and my wife for their support during this research. Last, but not least, I would like to thank someone who has been supporting me every day and has been a source of constant refreshment in my life, my Lord and my God, Jesus Christ.

> 'And with all his abundant wealth through Christ Jesus, my God will supply all your needs. To our God and Father be the glory for ever and ever! Amen.'
> Philippians 4 v 19-20.

"But if everybody obeyed that rule," said Alice, who was always ready for a little argument, "and if you only spoke when you were spoken to, and the other person always waited for you to begin, you see nobody would ever say anything."

- Lewis Carroll, *Through the Looking-glass*

for my parents, my sister and my wife

who have all been a source of constant encouragement in my studies

with all my love and gratitude

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
ELECTRONICS AND COMPUTER SCIENCE
Master of Philosophy

THE DESIGN AND EVALUATION OF AN INTELLIGENT FRONT END INTERFACE TO AN ELECTRONIC MAILBOX SYSTEM WITH THE HEARING AND HEARING IMPAIRED

by Keith Ian Watson

The main objective of this work is to demonstrate the feasibility of imposing an alternative dialogue style and structure on existing mailbox systems using a local intelligent interactive interface. The investigation of the development of better user interfaces to such systems is of benefit to both the hearing and the hearing impaired. The problem is analysed in a conceptual, as well as a practical manner involving the investigation of previous text communications aids, mailbox standards and user interface design principles.

The various standardisation issues involved in implementing an intelligent front end interface to a mailbox system are discussed with particular reference to the CCITT X400 mailbox model. One particular approach, that of using an intelligent interface as a mediator between the user and the central host dialogue, is proposed as the most flexible in terms of bandwidth limitations and interface development.

The design of a prototype intelligent front end interface, based on a simple microcomputer and modem is described and the various practical problems involved in matching the local interface state to the central host state are mentioned. Particular emphasis is placed on development of the prototype human-computer interface using practical guidelines.

The development of a suitable experimental evaluation model is described which makes possible the comparative evaluation of the efficiency of message transactions using the existing interface and the intelligent interface to carry out several typical mailbox tasks. Performance was measured quantitatively by the time to complete these tasks, and qualitatively by means of a questionnaire.

Analysis of the experimental results showed that the naive subjects found the intelligent interactive interface quicker, easier and more satisfying to use. On average, tasks took only half the time to complete when using the intelligent interface than when using the existing interface. For the experienced subjects there was no evidence of any significant difference in the performance between the two interfaces. The results have substantial implications for the designers of human-computer interfaces to both CCITT X400 and non-CCITT X400 mailbox systems.

The use of an intelligent front end interface to a mailbox system shows much potential and with the improvements suggested seems to justify further development as an aid for hearing and hearing impaired users.

C O N T E N T S

CHAPTER 1

INTRODUCTION

1.1 Electronic Communications Systems

In the past ten years the home computer has become increasingly popular and now more than a million families in the U.K. own some kind of computer. Previous use of these computers has mainly been in the leisure or 'games' market and the education field. Recent trends, however, suggest that people are now purchasing home computers 'with a purpose' such as word-processing, home finance or telecommunications.

In the same period advances in the fields of large scale integration, time shared computer systems, computer networks and communications, word processing and information management have come together to make the growth of office automation technology, and particularly mailbox systems, prolific. There are problems, however, which originate from this technological background. To appreciate this more it is appropriate to examine some of the elements of this evolutionary process in more detail.

Historically electronic transmission of messages has been around since the telegraph was invented by Samuel Morse in the 1840's. Since then most electronic message systems have been developed to overcome the problems with traditional communications links such as face-to-face conversations, the postal mail service and telephone conversations.

There are two types of electronic communications systems, asynchronous systems where communication is deferred, and synchronous systems where communication is simultaneous. Figure 1.1 shows a hierarchy of electronic telecommunications systems. Asynchronous text communications has many advantages such as reducing 'telephone tag'. (Here if a person is not in to receive a telephone call a message is left for the call to be returned. This also happens when the call is returned etc.) Also since messages are already in electronic form there is no problem with media transformation. (Montgomery and Benbasat 1983).

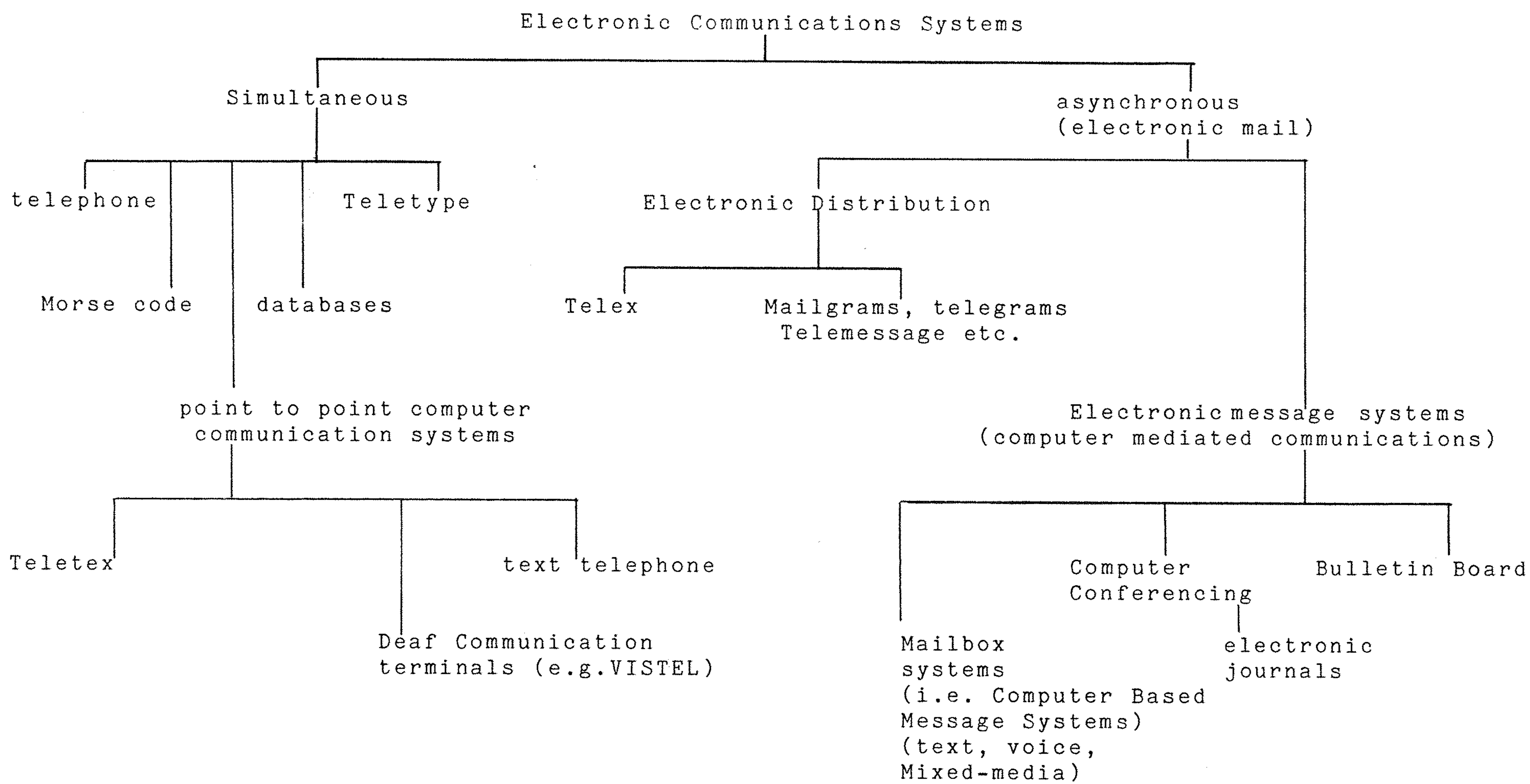


Figure 1.1 Types of Electronic Communications Systems

One definition of an asynchronous electronic message system is,

> 'the use of teleprocessing technology and services to facilitate the movement and management of information among people' Miller and Vallee (1980).

This, however, covers a wide range of facilities and would include all electronic communications systems such as telex, mailgrams, communicating word processors, etc.

A better definition is,

> 'a system of communication whereby a human at one end sends a message to a human at another end, in an asynchronous fashion, where some or all of the transmission is carried out electronically.' (p31) Anderson (1978)

Thus the interrogation of a computer data base is not electronic mail (since it is not person to person) nor is the telephone or morse code transmission (since it is not asynchronous). Since telex and mailgrams are only asynchronous due to the fact that they are delivered by hand it is better to define telex, telegram etc. as a subset of electronic mail called electronic distribution.

The subsection of asynchronous communications of interest to us is computer mediated communication which could be defined as,

> 'communication by help of a computer system which provides an assortment of services in the preparation, distribution and retrieval of information.' (Sandberg 1979).

This group consists of noticeboards (bulletin boards), computer based message services (CBMS) or mailbox systems, and computer conferencing (CC). Computer conferencing systems allow users to share a common file 'inside' which they communicate. The main feature is that the user does not send a message to a person but to a conference although in some computer conferencing systems, users can send personal messages to other people.

A mailbox system allows people to communicate with each other using sections of the host computer memory. Transmission involves sending messages, not to the recipient or to a location, but to a centralised host computer. Here the messages are stored in a section of memory rented by the recipient and he may access these messages at a time and location that is convenient to him. It is convenient to think of the messages as being in a conceptual mailbox which is based on the idea of the office desk (Miller and Vallee 1980, Sandberg 1979, Welsh and Wilson 1981).

Most present mailbox systems implement a simple direct mode of dialogue (user-initiated) which is typical of many computer mediated communications systems which have evolved from the mainframe computer system accessed by dumb terminals. Further there has been little incentive to improve the dialogue design because of the requirement to maintain concise communications in order to maximise information flow along low data rate telephone lines and ensure fast use of the central mailbox processor. While this type of dialogue is fast for experienced users, such systems require significant training and regular use to maintain proficiency and impose a high memory load even on intermittent expert users.

1.2 Project Definition

If computer mediated communications systems like mailbox systems are ever to achieve wider acceptance then simpler computer initiated dialogue styles will be required. These will make mailbox systems attractive both to naive users and for more expert users who cannot justify memorising a complex set of commands.

Much of the present work in the area of mailbox systems has centred mainly on the standardisation of the internal architecture and protocols of the message transfer system. There has been relatively little work on the human-computer interface design to mailbox systems. It seems as if designers have been keener to have the system operational and earning money than on developing prototype interfaces and evaluating the success of the design. What is required now is an investigation of how user interfaces could be improved and especially how new technology, such as local intelligence, can be used to mediate the access to mailbox systems.

One area where using mediated access to mailbox systems will have an impact is in the area of aids for the hearing impaired. It is ironic that Alexander Graham Bell invented the telephone while modelling the human ear in an attempt to improve communications for deaf people. Text communications in its many forms has a great potential for use by hearing impaired people but use of such systems has been inhibited by system cost and unfriendly interfaces. The aim of this project was to demonstrate the feasibility of imposing on existing systems an alternative, better designed dialogue style for the projected user population. This is particularly challenging in many areas. Firstly we will restrict ourselves to terminals affordable by the potential user population. Secondly we will attempt to develop a design based on some recognised user defined principles and thirdly there are many interesting technical problems involved in mediating the access to a mailbox system.

1.3 User Population

It is not the purpose of this thesis to give a definitive report of the various medical facts and figures about deafness since no definition of deafness has been accepted as standard. The potential hearing impaired user population of text communications systems is, however, important.

There are two main characteristics of hearing loss important to this report: age of onset of deafness and degree of deafness. People who have become deaf after acquiring linguistic skills are called post-lingually deaf (or hard of hearing) whereas those who become deaf before this are pre-lingually deaf (the deaf). This distinction is mainly social in nature and bears no relation to the degree of deafness, for example, a hard of hearing person can be profoundly deaf. The pre-lingually deaf, who are the smaller group, are less likely to benefit from text communications because the hearing impairment is compounded by poor literacy skills. The people who may benefit from some form of text telecommunications system are obviously those who have difficulty using the telephone.

To obtain a rough estimate of this group let us propose that this group will be similar to those who have difficulty hearing a person unless they are in a quiet room: this relates to a hearing loss of > 52 dB (Denmark et al. 1979). Recent up-to-date figures from a National Hearing Study conducted by Nottingham University (Institute of Hearing Research 1981, 1982, Haggard et al. 1981, Davis 1983) suggest that the prevalent rates of hearing loss among deaf adults are 17% (> 25 dB), 8% (> 35 dB) and 4% (> 45dB).

Thus a rough estimate of about 2.5 million people in the population may benefit from the text communication systems. Older people, who may show a tendency not to adapt to the new technology, account for 30% of this population because prevalent hearing loss shows an exponential increase over age.

The earlier work on text communications aids failed to satisfy the requirement of the deaf to be able to communicate with the wider hearing community. Mailbox systems could help solve this problem and since research on the improvement of user interfaces for deaf people will also have advantages for the hearing community, this thesis will concentrate on the potential benefits of this type of communications to the wider population.

1.4 Specific Objectives of the Project

Using an intelligent terminal to access mailbox systems is not a new area of work. It became apparent, however, from a review of work by others, that there was very little work on the use of local intelligence to mediate the access to mailbox systems for the mailbox user. Previous efforts have been in developing tools for the expert user, for example in the uploading and downloading of text.

The aim of this project is to demonstrate the feasibility of imposing an alternative dialogue style and structure on existing systems, to make possible the comparative evaluation of the efficiency of message transactions using each of the two different user interfaces to carry out standard tasks.

It was obvious from the previous work that the human-computer interface to mailbox systems had been designed in an ad hoc manner, where the design was based on intuitive ideas, by experts for experts. A systematic approach was adopted in this project which analysed the problem from a conceptual, as well as a practical, viewpoint. This involved the investigation of existing mailbox standards and human-computer interface design guidelines.

Another problem with previous interfaces is that the designers have not approached the potential user population during development. By using an evaluation of a practical prototype of a system this should enable the designer to test the reactions of users to the interface design features prior to a full implementation. This approach will help designers to produce more usable human-computer interfaces in the future.

1.5 Structure of the Thesis

We begin with a critical review of the various types of distributed and centralised text communications systems (Chapter 2). These systems are then categorised in terms of the various approaches to the design of aids for the disabled. The advantages and disadvantages of each type of design philosophy are discussed. One type of modified standard system, based on an intelligent front end interface to a mailbox system, is suggested as being potentially the most beneficial to hearing impaired and hearing users.

Chapter 3 describes the various standardisation issues involved in implementing an intelligent terminal to a mailbox system with particular reference to the CCITT X400 standards. It is shown that one of the three possible approaches to implementing an intelligent front end interface offers the most flexibility in terms of bandwidth limitations and interface development and so all subsequent effort was turned towards this approach.

A description of the design and implementation of a prototype interactive intelligent interface to a mailbox system follows in Chapter 4. The interface design characteristics used in the development and the various practical problems involved in matching the state of the local interface to the central host are mentioned.

Chapter 5 reports the initial design and development of the evaluation experiment carried out to compare the performance of naive and experienced users accessing a mailbox system via the conventional dumb terminal, with the performance when access to this system was mediated by the intelligent interface. Performance was measured quantitatively by the time to complete several typical mailbox tasks, and qualitatively by means of a questionnaire.

In Chapter 6 the results of the practical trials are reported and the implications in the design of both user interfaces are given. Chapter 7 contains a discussion on the implications of the results for interface designers and includes suggestions and recommendations for future work. The conclusions of the project and a final summary are presented in Chapter 8.

CHAPTER 2

SURVEY OF PREVIOUS WORK ON TEXT COMMUNICATIONS WITH THE DEAF

2.1 Introduction

Text communications can be divided into two main areas, distributed (usually synchronous) and centralised (usually asynchronous) communications. Distributed communication is the point to point system exemplified to the hearing person by the telephone whereas centralised communication is based upon sending messages via a central computer. This is not a comparison of the two types because each method has its advantages and disadvantages. The reader is referred to appendix 1 for a comparison of distributed and centralised systems. Below the various sub-categories are described and then there is a general discussion on the shortcoming of each method and how they have been, or can be, of use to the hearing impaired.

2.2 Distributed Communications Systems

2.2.1 The Teletype (TTY) Network

The emergence of the TTY network in North America marked the beginning of text communication for the deaf. In the mid 1960's a coupler was developed which made communication between teletypewriter (TTY) machines possible over the telephone line. Some years later old Baudot teleprinters were distributed through an organisation called the Teletypewriters for the Deaf Incorporated (T.D.I.) This organisation was based on a committee set up to negotiate the release of surplus teleprinters from the American Telephone and Telegraph Company (A.T.&T.). (Ward 1974, Stoffels 1980, Derfler 1982)

The main disadvantages of the TTY were the expense (both parties must be online to communicate) and poor reliability which served as a big deterrent to enlarging the network, however eventually a network of several thousand units were scattered throughout America. In the U.K. use of such equipment only proliferated on a small scale (King-Beer 1973) and only about 165 units were sold.

2.2.2 Desk Top TTY

By the early 1970's the use of microprocessors had enabled manufacturers to develop smaller, cheaper and more efficient versions of the TTY, which are still in use today. The main features are a 3-row QWERTY keyboard, soft one line display and inbuilt acoustic modem. They are compatible with TTY's and some can communicate with ASCII terminals. In America this equip-ment was sold under brand names such as AMCOM 1 (American Communications), PORTAPRINTER (Krown Research), and VUPhone (Plantronics). Other facilities that were sometimes included were printer hardcopy, cassette storage and local memory storage. (Stoffels 1980, RNID 1980).

2.2.3 Portable TTY

Neither the desk-top TTY nor the teletype were really portable and so deaf people were cut off from phone communication when away from their base. In an attempt to alleviate this problem Levitt (1981) suggested using an off-the-shelf pocket computer (Radio Shacks TRS-80) with a special telephone interface which could convert the signals being read in and out of the computer into a train of audio pulses that could be sent down a telephone line. A modem could be used for more reliable transmission over the telephone line. This device operated like a pocket teletype, with the additional advantages of memory and logic. Messages of short length could be pre-prepared and called up during transmission.

None of TTY examples have ever been used much in the U.K. as there is no extensive teletype network (RNID 1980). The main disadvantage was the isolation caused by the deaf having to use different equipment to hearing people. Other problems were the cost of connection and use, there was no facility for automatic connection and the acoustic modem was prone to line noise difficulties.

2.2.4 The VISTEL

The VISTEL (VISual TELephone) was launched in 1981 from collaboration between British Telecom, a charity called Breakthrough Trust, and a firm called Kegwain. It is a portable Deaf Communications Terminal

(DCT) for person to person communication. The VISTEL is based on the desk-top TTY's but is simpler to use.

It has its own protocol different from the American DCT's and has been designed to communicate with a PRESTEL terminal by emulating the PRESTEL host computer protocol (RNID 1981) but there is no evidence that this facility actually works. A 20 character moving illuminated strip displays the simplex communication between two VISTEL terminals. Also there is a voice over facility for people with good speech in order to economise on line time. Like some desktop TTY's it has local message storage (about 800 characters) and allows incoming message capture. The VISTEL is still in use today by about 700 deaf people however this approach has all the shortcomings identified with custom built aids (see later).

2.2.5 Text Telephone Systems

In recent years many countries have developed their own deaf telephone system and some of these are described below.

a) Diatext (manufactured by TELI in Sweden and other Nordic Countries) uses the CCITT V21 (300/300) protocol. It has facilities for pre-prepared messages and to save incoming messages, display tone indication, auto answer, speech input, uses the ASCII character set, QWERTY keyboard and has a 9" screen (Wilhelmsson 1984).

b) Text Telephone 2000 (Germany) has two rows of 40 LCD characters, tape-saving facility, television interface, uses an acoustic coupler and is portable. A hardcopy printer can be added and it has memory for preprepared messages and to recall the previous phone message. It uses the EDT standard CCITT V21, 110 baud (and so is compatible with text telephones in Austria, Belgium, Switzerland and Spain) but can also communicate with the Netherlands DTMF standard and the Sweden Diatext. (Krause 1984)

c) Italian Text Telephone which is similar to the above and uses CCITT V21 protocol at 110-300 baud and the Annuaire Electronique which uses CCITT V23 (1200/75 - duplex) and CCITT V25 (1200/1200 half duplex). (Dullemond 1984)

2.2.6 Back to Back Computer Communications

Numerous examples of back to back computer communications software exists from those to allow mainframes to commmunicate globally (e.g. for the Unix operating systems) right down to those for desk top microcomputers (e.g. Torchmail for the TORCH range of computers). Such communication has been improved recently by the development of more easy to use software and software-controllable multi-baud rate modems however this still requires a fair amount of technical expertise to operate and so is mainly limited to experienced computer users.

2.2.7 Touch Tone Solution

Johnson and Hagstad (1981) and Glaser (1981) both suggested using the fact that once a phone call has been established further touch tone pulses are propagated over the American telephone network as speech is. When the phone buttons are depressed specific tones are sent down the line and these can be decoded at the other end. Each combination of button presses representing an ASCII character. The sending party only requires a touch tone phone and the deaf person requires a phone, personal computer and modem/decoder. Obviously if the sending party is deaf he also will require a phone/decoder instrument. However in the U.K. this is not of any use because there is no touch tone system, except internal to some PABX's.

In the Netherlands the Teksttelefoon text telephone system (manufactured by the Nederlandsche Standard Electric Mij. under the name ITT-TEXTPHONE) also uses the DTMF (dual tone multi-frequency) method. Instead of using the telephone keypad, however, a prototype typewriter-like apparatus was developed. It has similar facilities to the text telephones in 2.2.5. including two lines of 40 characters display, local memory for preprepared messages and to read back conversations, tone indication and speech communication.

2.3 Centralised Communications Systems

2.3.1 Bulletin Boards

These are electronic notice boards on centralised computer systems by which any user may post 'notices' for people to read. They can exist autonomously, as public notice boards established by companies or individual people, (e.g. Communitel) or as part of a mailbox system. The problem is that this system lacks the privacy required for personal communication.

2.3.2 Mailbox Systems

Mailbox systems, sometimes called Computer Based Message Systems (CBMS's) are systems 'for textual communication where users employ computers as an integral part of the communication to produce, distribute and process messages' (Smith 1982). CBMS's come in many forms and include turnkey systems (e.g. IBIS Videotex, IMP, XIBUS), office automation software packages (e.g. All In One, PROFS), mailbox bureau services which are usually national services (e.g. COMET, PRESTEL mailbox, Notice, Quik-Comm, Telecom Gold), voice mailbox systems (e.g. ADS, DVX) and mailbox software packages (e.g. Com/Portacom, HPmail) For a brief summary of all these systems see Wilson (1983).

An example of such a system used to some effect by deaf people is a commercial international system called Telecom Gold. Basically it allows a user to send, store and receive messages in a section of memory space (the mailbox) on a central host computer which is accessed from a local dial up terminal, i.e. it acts like an electronic version of a physical mailbox (Miller and Vallee 1980). Housman (1980), Walgreich (1982) and Breakthrough Trust (1984) mention how useful such systems have been in helping deaf people to interact online.

2.3.3 Computer Conferencing (CC)

Unlike computer mail CC is a technique to enable users to conduct a conference with each other, usually asynchronously using a computer to store their conference transcripts. The user can enter, modify and store items on a shared electronic scratch pad. The information exchanged is free form rather than structured, i.e. no defined order between items is required. Examples of computer conferencing systems are CONFER, EIES, FORUM/PLANET, etc. An excellent summary of these systems is given in Strom 1982 and Hiltz and Turoff 1978.

Electronic journals are one of the applications of computer conferencing where a CC is used 'to aid the normal procedures whereby an article is written refereed, accepted and published.' (Shackel 1982). e.g. BLEND.

2.3.4 The Telephone Bureau

This was developed to allow a telephone conversation between a deaf person and a hearing person. In the bureau the hearing person's speech is converted by a human intermediary at the bureau into text which is transmitted to the deaf person's terminal so the deaf person can read it. The deaf person will then, either speak directly to the hearing person or type the reply to the intermediary who speaks to the hearing person. (Grossfield, et al. 1982).

There is a similar system called Teletel (Besson 1984) being developed in France, however this is a non-vocal system. This experiment used simple microcomputer terminals called MINITEL's which had a display screen and a keyboard. Interconnection between MINITEL terminals is provided by a host computer located in a telephone exchange so there is no human intermediary. Other solutions that have been suggested to allow connection between these terminals is to connect users via the Videotex gateway or using an adaptor at the callers end.

2.3.5 Database Systems

Examples include PRESTEL (Micronet), Lexis, and computer abstracting services and these can be provided by public or private information providers. The problem is that database systems are not optimised for efficient text communications although some have elementary mailbox facilities (e.g. PRESTEL) and Telex links. Some include facilities specifically for computer users such as home banking, software databases and games. e.g. COMPUNET for the Commodore computer range.

2.4 Multi-function Systems

Some systems include most of the above types of communications or at least primitive versions of them. For example PRESTEL includes a database, chatline and a simple mailbox system. Many computer conferencing systems have mailbox facilities and noticeboards for general information. In the past many local terminals would only allow access to certain types of system unless certain parameters were changed manually. However newer systems based on the integrated microcomputer/software/modem combination will soon make access to all these types of systems from the one terminal easier (e.g. the IBM Mailmanager or the intelligent terminal (see later)). The main problem with present multi-function systems is the interface design, which is mainly for the business user and not the naive or casual user.

2.5 Discussion of Previous Systems

Vanderheiden (1981) suggested that there were three approaches to providing aids for the handicapped.

1) Commercially available aids which are suited to certain types of people. They have the disadvantage that they may not meet a specific need or if they do then the aid cannot be produced economically.
2) Custom-built aids which are tailored to individual requirements and so cost per function is high.
3) Modified standard aids lie between these two extremes and with the use of microcomputers may prove to be the most advantageous.

This classification, however, is confusing since some custom designed aids and modified standard aids may become commercially available. It also implies that the provider of an aid may be the designer which is not necessarily true.

This could be modified into different categories where the aids have been classified by availability and by design procedure. The latter can be split into custom-built and customised aids and the resulting hierarchy is shown in figure 2.1. The distinction is that the customisation of the aid is, to some extent, reversible and so it is possible for the aid to be used by other non-handicapped people. Now each text communications system is discussed in terms of the advantages and disadvantages of the various approaches.

i) Examples of custom-built systems are the text telephones, desktop TTY's, VISTEL and telephone bureau. These suffer from the

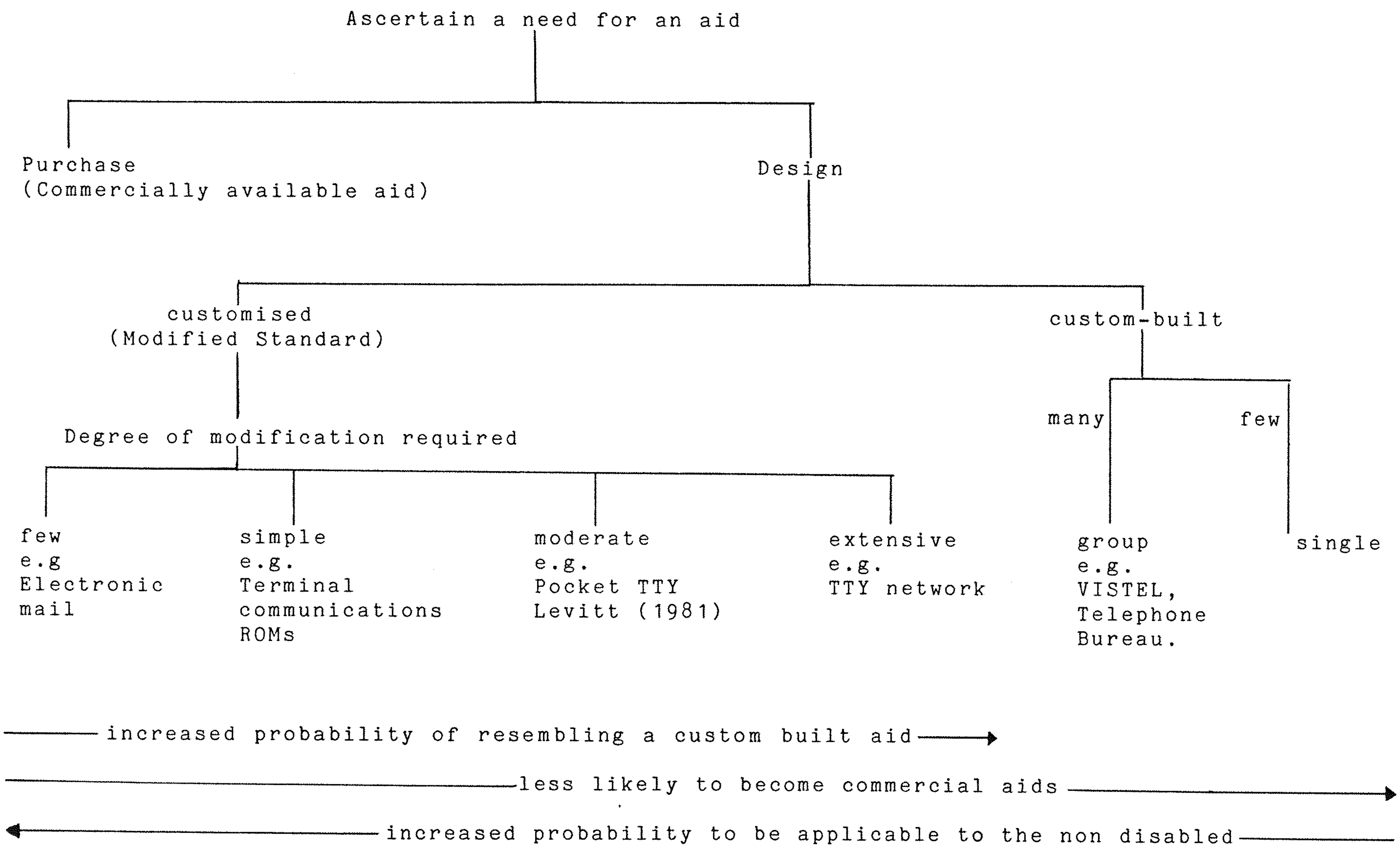


Figure 2.1 Decision making structure to designing aids for the Handicapped

shortcoming identified with any custom-built aid, namely,

a) There is a lack of possibilities to expand the facilities that can be provided. For example the VISTEL has a 20 character screen with no TV compatible output and an acoustic modem which gives poor performance on long distance telephone lines.

b) Hardware customisation is expensive and the limited market curtails demand;

c) The use of special equipment makes the deaf person feel isolated. The VISTEL was designed to be able to communicate with PRESTEL terminals and also it's own kind at a back to back data rate of 75 baud/75baud. It is incompatible with home microcomputers which communicate at 300/300 baud, PRESTEL itself and electronic mailing systems. Similarly only some desk-top TTY's can communicate with ASCII terminals. The telephone bureau enables deaf people to contact hearing people who do not have text communications systems but still makes the deaf person dependent on the human intermediary and lacks privacy.

ii) An example of a standard system is the teletype network. Such an aid has the drawback that it has not been designed for the job and so can only fulfil some of the requirements of the deaf person. This particular aid has the disadvantages of being large, expensive to run and prone to mechanical breakdown.

iii) An early example of a modified standard aid was the pocket teletype built by Levitt and this showed the potential in using a home computer as an intelligent terminal to control and enhance textual communications. The touch tone solution is another example of a modified aid and has the advantage that the hearing person does not require any extra equipment unless the deaf person does not have good speech. This idea is limited in its application in the U.K. because there is no national touch tone network and also because it cannot communicate with other text systems. Both these systems are not ideal solutions as they would still require some sort of technical expertise to operate.

The most promising modified standard aid are the mailbox systems and computer conferencing systems. A present day example is the use of a local dumb terminal to access mailbox systems such as Telecom Gold.

Walgreich reported on the use of a mailbox system, called DEAFNET, operational in the United States. The user group was a selection of deaf individuals and their families, people from deaf schools and hearing people working with the deaf. The system provided mailbox facilities, a bulletin board and a computer conferencing option. Reported average use was 20 minutes per day per person with many complex facilities being used, including the editor, showing the willingness of inexperienced users to use the system.

The Visicom project (Breakthrough Trust (1984)) was sponsored by the Department of Industry to attempt to evaluate new methods of typed visual telephone communications methods, direct and indirect. The project involved deaf people in the use of electronic mail systems (such as Telecom Gold, PSS mailstream and the PRESTEL mailbox) and direct communications via VISTEL or the Telecom Gold chat facility. 75 hearing impaired people of various ages, educational standards, literacy levels and technological experience participated in the project which lasted for a year. They were supplied with equipment of various types mainly consisting of home computers, office terminals (about 33%) or viewdata terminals (66%) and the majority were provided with Vistels.

However these simple terminals do not provide the complete solution as mailbox systems are difficult to use and the deaf subscriber must master a fairly complex set of commands to access and use the system (Telecom Gold (1983)). In addition a charge is levied for accessing the computer over and above the normal telephone charge. Though direct communication with another user on the computer is possible it is primitive and is not the normal method of communication. It is unlikely that, in its present form, most deaf people would wish to use such a system for communication because of its cost and complexity.

The common feature of all the examples presented so far is that the user interface has been difficult to use and the user must have some technical expertise to operate the systems. The complexity and cost could be effectively reduced by a) more consideration for the naive user in terms of a more easier to use interface and b) by the use of local intelligence in local terminals. The cost factor is more important when it is considered that it takes significantly longer to type a message than to speak it. A simple survey was carried out during the VISICOM project where a spoken conversation was simulated using certain typed conversations which had been recorded from TELECOM GOLD. Results showed that a typed conversation took more than seven times longer than a spoken conversation (Judson 1986).

2.6 An Intelligent Terminal as a Text Communications Aid

Various approaches to designing aids for the deaf have been examined. These aids have failed to satisfy the needs of the deaf mainly due to the ad-hoc design strategy with which they have been implemented. In the longer term, the approaches of purpose designing a telephone for the deaf like the VISTEL or extensively modifying an aid are unlikely to be cost effective because of the relatively small market for such devices. The modified standard approach, based on a standard micro-computer, is the method that will provide a low-cost solution to deaf telecommunications. The advantages of this type of method are obvious.

1) The computer can be designed to be a multi-function terminal to interface to many different types of text communications system. The intelligent terminal can deal with all the technical requirements of the telecommunications systems from a local knowledge base. (e.g. protocols, data rates etc.)
2) The intelligence of the microcomputer can be used to 'hide' the complexity of the communication from the user in a way that is not possible with a dumb terminal.
3) The local interface software (or 'protective ware') can present a standard interface to all of the text communications systems. The protective-ware is placed between the user and the existing interface and "operates rather like a communications channel in that it has at its front end an interface with the user and at its rear end an interface with the existing operating system of the machine. All messages between the user and original operating system are intercepted, monitored and if necessary interpreted." (James 1981).
4) The use of standard equipment removes the problems of lack of standardisation, increases the expansion possibilities and the user receives the benefits of large scale production.
5) The local intelligent terminal can be used to reduce the cost of communication by allowing pre-prepared messages and the

downloading and storage of text messages.

6) The separation of dialogue software from mailbox applications software (Ehrich 1983) means that personalised dialogue styles can be offered to different user types (Carey 1984) without changing the requirements of the centralised system's existing software. This is important and will be discussed in detail in the next section.

7) Another advantage of this approach is that this work is also applicable to hearing people since both groups would benefit from a better user interface.

2.7 Personalised Computer Dialogues

Disabled people need to use the same programs and accomplish the same tasks as anyone else. Yet they are sometimes prevented from doing so because the interface program takes up the space that the utility program, such as a word processing package, may require. (Vanderheiden 1982). The use of microcomputers as intelligent terminals in text systems overcomes this problem since the application program (e.g. electronic mail) can be in the communications computer and the interface program can be at the local terminal.

In the special case of deaf people the translation of one faculty for another (e.g. reading text on a screen for hearing) is an ideal task for a computer. The personal computer offers the possibility of low cost aids for the hearing community. However as Myers (1982) observes;

> 'A deaf individual requires a different set of information translations. He can, with some exceptions, use standard keyboard input and display output. Audio feedback from a key depression of an end of line warning buzzer would have to be translated to a tactile or visual form. The hearing impaired person's principal handicap is receiving voice communication...'

In a text communications system the lack of non-verbal clues can be detrimental to deaf people who use them in normal face-to-face conversation (e.g. lip-reading, facial expressions). Even hearing people using the telephone use clues unavailable to deaf people, such as voice intonation and voice level. So the local intelligent terminal can substitute visual information for audio information such as tone indication (ringing, engaged, etc.).

2.8 Conclusions

The hearing impaired do not wish to be isolated into small user groups and require a telecommunications system through which they can communicate with both the hearing and hearing impaired. Electronic mailbox systems seem to provide the most effective system which satisfies this need. At present, however, these systems share some of the problems of all text communications systems in terms of cost and complexity. The investigation of the use of an intelligent terminal to use text communications systems provides a vehicle through which to develop a cheaper and less complex aid for the hearing impaired. As long as the special needs of the hearing impaired for audio to visual/tactile feedback is recognised then the added advantage of this approach is that the conclusions can be applied to the wider hearing community, since both groups have the need for better user interface design.

CHAPTER 3

STANDARDISATION ISSUES OF INTELLIGENT TERMINALS

3.1 Introduction - Potential Problems and Solutions

It is clear from the previous chapter that there are two main problems which may hinder the use of text telecommunications systems in the future: the slow response times and poor user interfaces. The slow response time is due to two factors, the size of the data being transmitted and the rate at which it is possible to transfer data over a voice grade phone line (see Appendix 2). This creates an input/-output bottleneck at the point where the user makes contact with the text communications system. The two solutions to this are,

1) to improve the method of coding to maximise the amount of meaningful information in a given number of bits (e.g. develop modems which operate at higher data rates) and,
2) the use of a local terminal as an I/O processor.

These points have been noted by other authors (e.g. McManis 1985). However the problem of how the local processor fits into the existing standard network model and what advantages there are in each approach has not been fully investigated. This chapter examines the ideas behind the use of an intelligent terminal to access centralised mailbox systems and finally these ideas are applied to general text communications systems.

3.2 Models of a Mailbox System

Before discussing these points, it is important to ask the question, 'Should the designer begin from the user's model of the mailbox system or the system model?'. When designing an interface, it seems reasonable to assume the former, however there are problems with this approach.

> 'A classic misconception is that the user can tell designers what he needs. ..... Certainly users must be polled and seriously guarded in selecting opportunities for system application.... (but) A user cannot anticipate the form of a technology unknown to him; energy is much better spent determining the user's values and criteria for success in the context of his current work environment.' Bair (1983)

It is essential not to overestimate the user's ability to assess his own understanding of the system. Certainly the starting point should be an analysis of the user's needs (as described in the previous chapter), however the views of the user should be considered in the context of the designer's knowledge, which will be based on the existing system model. Thus it is better to begin with the correct system model and design a prototype interface which can project this model to the user. The system model should map onto the user's model where the two coincide and correct the user's model where they do not. It is easier for potential users to give valid opinions on prototype interfaces than hypothetical ones and the information gained from this can then be used to develop a user-driven design.

It would help the reader considering the systems model if he or she had some familiarity with the CCITT X400 recommendations which describes a computer based message system model. Cunningham (1984) and Wilson (1984a) gives an introduction to the actual CCITT standards (CCITT 1983). Below a summary is given which will provide some background for the following discussion. Figure 3.1 illustrates the functional entities of the mail system model.

The network which deals with the message handling network is split into two services, the message transfer network and the interpersonal messaging service. The former is made up of nodes (message transfer agents) and provides the store and forward service that allows data transfer regardless of its application using the envelope information (note - the envelope is that part of the message containing, amongst other things, the recipient of the message). The interpersonal messaging service corresponds to the conventional mailbox service as described in chapter 2.

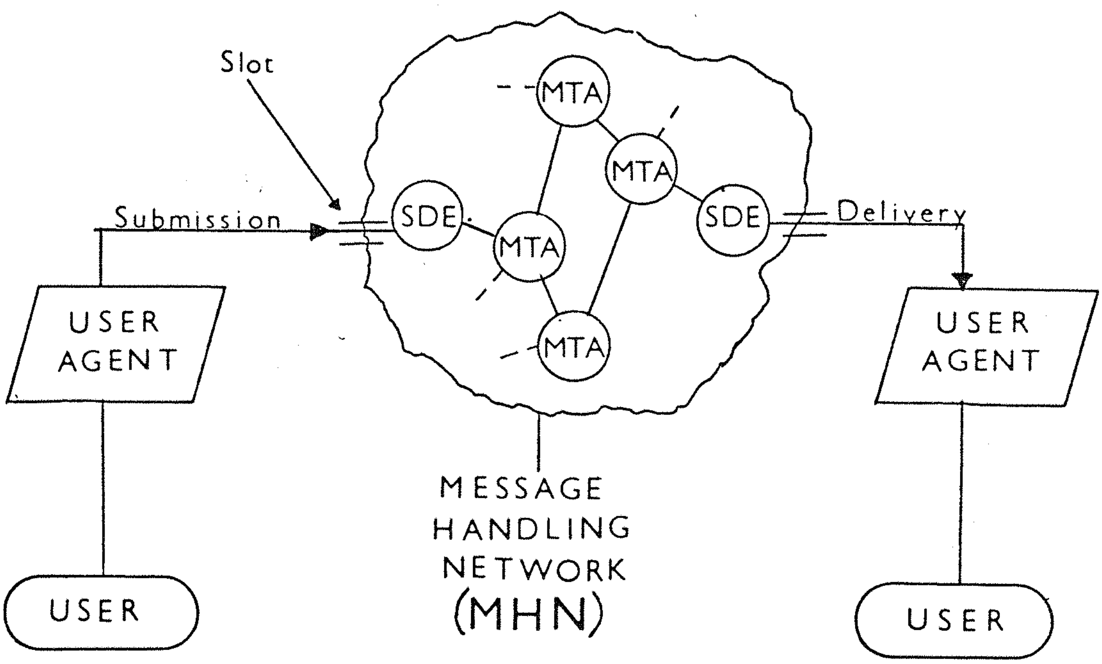


Figure 3.1 Functional Entities of a Mailbox System

Users of mailbox systems communicate with each other via mail items, consisting of an envelope and the textual content, which are based on the model of an office memo. The user prepares a mail item in the user agent which is accessed via some sort of computer terminal (usually a dumb terminal). The user agent transfers the item, through the slot, to the submission delivery entity (SDE) of the mail system and it is at this point that the mail system accepts responsibility for it: this is called the authority boundary. Receipt of the message is then defined as message transfer from the recipient's user agent to the actual recipient. The user agent only interacts with the mail system for the user and so does not provide mailing functions.

The basic problem which must be considered when implementing the intelligent interface in a commercial mailbox environment is the public to private domain transition. The user exists within this private domain whereas the mailbox system exists within the agency domain. The authority boundary is invisible to the user and must be crossed every time the user accesses the system. The three possible boundaries are illustrated in figure 3.2 a,b,c (Schicker 1982).

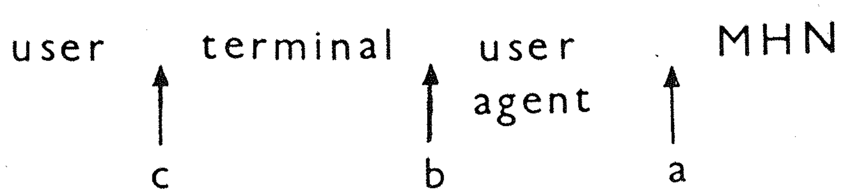


Figure 3.2 Possible Authority Boundaries in a Mailbox System

a) the user owns the user agent and so can tailor the hardware and software to their requirements. A boundary here means that a rigorous protocol must exist to use the mail system.

b) the user agent is in the agency domain so the user only requires a dumb terminal however the user can only access the services the agency provides.

c) the user agent terminal and mail systems are both within the boundary of the agency. c.f. public telephone booths.

In their efforts at standardisation the CCITT have mainly concentrated on message transfer services (i.e. delivery standards) or associated services (e.g. directory services) rather than the user interface because private user agents can be tailored to user's requirements. However with the proliferation of multiple mailboxes this creates a problem. Marshall (1985) comments,

> 'The cognitive complexities associated with multiple mailbox ownership and use will be a real problem at least until uniform, universal open system interconnection is a reality; and perhaps well beyond this due to the heterogeneous nature of open computer systems.'

At present each system provides a different interface and there is no incentive to standardise because there is no standard mailbox system. Any user who may wish to access more than one system is unlikely to have the time or inclination to learn all the command sets. This means that although the existing mailboxes have the potential for extensive development, they are at present poor communications environments. One example of this is the problem of information overload (Palme 1984) because of the ease of sending messages to many people.

3.3 The Intelligent Interface - the Alternatives

The intelligent interface is a solution to the problem of multiple mailboxes because it can act as a mediator helping the user interact with the user agent. Following the discussion on authority boundaries it is apparent that there are three possible places for the intelligent interface in the model (figure 3.3 a),b),c)).

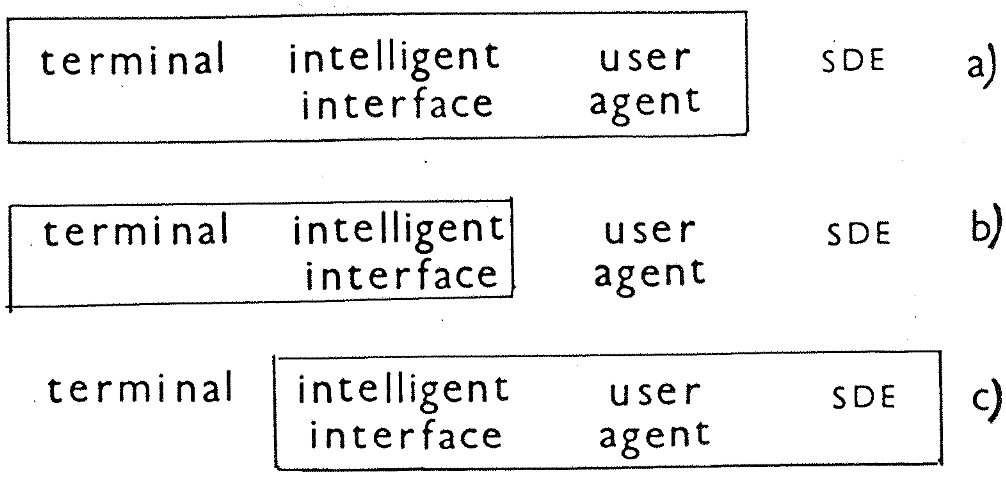


Figure 3.3 Intelligent interface possibilities

1) The intelligent interface could be an integral part of the local user agent and the mail system has the submission/delivery entity (SDE) which acts on behalf of the local user agent.
2) The local interface acts as an intelligent terminal only and the user agent is still in the domain of the mail system.
3) The intelligent interface is provided by the agency as a front end interface to the mail system.

Option 3) is only feasible if separate gateways into the mailbox system are provided for each type of user. Here the user could select the required interface by using a separate telephone line or being asked by the system on entry what interface was required. If a common interface for each user is to be defined this would require broad agreement on behalf of the manufacturers as to how the interface should be designed. This is unlikely to happen because of the facilities each manufacturer wishes to provide. The other problem is the input/output bottleneck mentioned earlier. If the interface, for example, provided extra help facilities this would cause a significant overhead in display time. So purely on the basis of the bandwidth requirement, the intelligent interface must be produced locally.

The type of system suggested in 1. is only useful in message systems where the user has all the facilities locally and the message system just provides the SDE. This has problems in that any user with a dumb terminal may not be able to access the system as no user agent functions are provided by the mailbox.

It is our opinion that option 2. is the best method for commercial systems. In this option the user agent and the SDE would be under the control of the mail system and so would the command structure and facilities. Wilson et al. (1984) proposed a similar idea where the user agent sublayer of the CCITT X400 model was further split into, what are termed, the active and passive mailbox. The role of the passive mailbox would be to collect mail for the user from the mail system and deliver any mail to the mail system. The active mailbox function would be to collect mail from the passive mailboxes.

> 'Each passive mailbox will have a different structure of mail, with different fields meaning different things, [so the active mailbox] ..must then decode the [mail item] according to its knowledge of the structure that mail system.'

The active mailbox would also have to be capable of sending letters to the passive mailboxes in the correct format to enable them to be forwarded to the mail system. Although the paper itself made no reference to the authority boundary it is clear that the boundary is implicitly placed between the active and the passive mailbox (i.e. option 2). In terms of the CCITT model the passive mailbox is, in fact, similar to what the CCITT X400 standard describes as the SDE. What is certain is that as the power of terminals increases so the facilities provided by the mailbox system will move to the local user environment and the authority boundary will migrate towards the active mailbox.

3.4 Examples of Intelligent Terminals

Sharpless (1979) developed an intelligent terminal (Micterm-2) based on a simple microcomputer to provide an extended user interface for access to a Viewdata system. Sharpless concluded that "the combination of the benefits of both (the remote data base and the local microcomputer) would seem a logical step leading to the intelligent home information terminal which would have local storage and processing capabilities of its own but would use the public telephone network for external communication and access of information." Early examples like the Micterm-2 and the VISTEL, although they had limited memory and processing power, illustrated the possible advantages of interactive intelligent terminals.

Having recognised the potential power in 'protective-ware', James (1981) decided to investigate this concept by providing a terminal at the Science museum in London which was attached to the Imperial College computing system. The terminal allowed any member of the public visiting the museum to initiate a dialogue with the computing system and to have practical experience of a few simple computer programs. The intelligent terminal intercepted, monitored and mediated the conversation between the user and the computer. The assessment of the results showed that the majority of users found the intelligent interface easy to use and had few problems with its operation.

One practical implementation of an active/passive mailbox is the Kaye and McDowell(1984) intelligent workstation. This is capable of accessing multiple CBMS's and "provides the user with a single user agent (a terminal which acts on behalf of the user) which can be integrated with his local document creation and storage facilities and mediates his interaction with the external CBMS." (Abstract) The active mailbox is the intelligent user agent, with its own in-box and out-box and this UA uses the existing mailbox systems as passive mailboxes. This user agent operates in a 'batch mode' where access to the message transfer agents occurs without user interaction. This implementation identified numerous practical problems in dealing with current non-CCITT systems (e.g. inconsistent dialogues, data corruption due to line noise) which result from the mailbox systems expectation that is it dealing with a human operator rather than another computer system.

This project proposes a prototype interactive intelligent terminal which would be designed to demonstrate the feasibility of imposing an alternative dialogue style on existing systems. An interactive system would not have some of the problems of a 'batch' terminal since it would be possible to use the intelligence of the user to simplify the interaction. This does not preclude the use of a 'batch' solution for delayed transmission of messages.

3.5 Intelligent Terminal - Summary of the Advantages

There are many advantages in using distributed intelligence in general text communications systems and these include the following.

1) Helps reduce work overload of centralised system

a) This overcomes the I/O bottleneck because the user interface is local and the mail system only handles the text transmission.

b) It reduces the problem of accessing multiple mailboxes as mentioned earlier.

c) The intelligent terminal can copy as many user agent functions as the individual user requires or rely on the mail system UA functions and so allows dumb terminals to use the system.

d) It may still be feasible to provide some sort of reduced service if the centralised systems is unavailable for any reason. e.g. point to point communications (Sommerville 1981).

e) It can be used to implement the encoding of information to decrease the average number of bits per character for example by the use of shortforms (Damper et al. 1986). Here the shortforms could be transmitted by the terminal and the whole words reconstituted at the other end.

2) Helps improve interface for user

a) Each centralised system has two parts, the low-level code to control the display and transmission of data files and the user interface. The low-level code is machine dependent but the user interface is entirely generated at the decisions of the designer. The use of an intelligent terminal allows personalised dialogue styles to be used for different user types without affecting the central applications software.

b) As mentioned in chapter 2 the intelligent interface software can mediate the dialogue between the user and the computer. This is not only true for inexperienced user interfaces but also for people who use command line interpreters. The terminal could preprocess the commands for syntax errors thus increasing the data throughput of the system and simplifying the interface (i.e. making it more forgiving).

c) The intelligent terminal can take on secretarial tasks such as sorting and filtering mail, summarising and prioritising incoming information, auto forwarding, archiving, local mailbox facilities, delayed transmission of information to make use of non-prime time rates, etc. (Marshall 1985) if required.

d) The agency which provides the system is not restricted in its interface design and can provide facilities it deems suitable.

e) Local help facilities, menus, screen editors, etc. can be provided easily without the time overhead because response time will be independent of modem times and the centralised system response time (see 1a).

Having decided that the interactive intelligent terminal is the best way to implement a user interface, the next stage was to decide what guidelines should be used in the actual design.

3.6 User Interface Design

A human-computer interface is the means by which a human enters information into and receives information from a computer and so, to the user, the interface is the system. Even if an intelligent terminal is used, it may inherit all the problems of previous interfaces unless some specific guidelines are used.

In the past the main problem has been the discrepancy between the actual needs of the naive user and their perceived needs as seen by the expert system designer. In early computer systems the focus of the designer was on the system and its financial constraints and not enough effort was directed into studying the users and their requirements. Therefore early user interfaces exhibited poor interface design characteristics such as: a terse dialogue, short counter intuitive commands, and unnecessarily complex logon procedures

(Selander 1982). This intellectual gulf between the system designer and the user has caused programmers to design interfaces which present the wrong conceptual model to the user. Bair (1983) calls these types of interfaces 'working non-solutions'.

Also, although there are some excellent texts for interface designers (e.g. Eason 1982 ,Gilb and Wienberg 1977, Jones 1978, Hiltz and Turoff 1978, Kidd 1982), these are all too often ignored with the designer relying more on intuitive design than well-chosen design principles (Hammer et al.1983).

In the future as the day to day use of the system takes up most of the cost so the user interface will become more important. A set of state of the art procedural guidelines are required by the designer but unfortunately they do not exist. In an attempt to fill this gap Ramsey and Attwood (1978) published an extensive annotated literature survey of the state of the art in human factors in computer systems. In two follow up papers (1979,1980) they concluded that, although there had been a few attempts to develop guidelines for interactive interface design, available guidelines were sketchy and virtually non-existent.

Ramsey and Attwood (1980) concluded that it was not possible to develop a 'quantitative reference handbook' in this area. However it seemed feasible to develop a 'human factors design guide' which discussed issues and methods in the context of the design process. This 'design guide' would have a structure which would parallel the major steps in the design process. Although such a guide may not at present be feasible for all interactive systems, in general, it should be possible to develop one for restricted types of systems since user behaviour and system properties tend to be highly task specific.

They suggested that the possible sections topics for such design guide could be, user types and characteristics, tasks analysis, requirement analysis and pre-evaluation, interactive dialogue, output devices and techniques, input devices and techniques and evaluation of system performance. It is on this structure that our design was based and it is hoped this work will contribute some proven guidelines to be included in a future guide.

The following sections set out briefly the design guidelines used to construct the interface. Space precludes a full description but the reader is referred to the references in each section for more information and particularly to the Ramsey and Attwood annotated bibliography papers. Since many of these rules have not been proven by extensive experimental evaluation, some designers have experienced difficulty applying them (Marshall 1984). So it was decided to apply some of them in the interface and conduct an evaluation to establish the validity of some of these design rules.

3.7 Design Guidelines

3.7.1 User Types and Characteristics

User types are described in various texts (e.g. Martin 1983, Kidd 1982) but a simple classification might be divided into naive, novice, competant and expert (in increasing levels of experience). The amount of use of text communications systems by a typical future user may be similar to use of other facilities like electrical equipment or the telephone now. It seems likely, therefore, that the majority of the user population will consist of two types: novice users who are inexperienced and require extra help and competant users who use the system intermittently and need reminders.

Also one of the ironies of a good human-computer design is that the user is liable to learn quickly and so outgrow the system. If the interface is not adaptable then as the naive user becomes more experienced he will become more dissatisfied with the interface limitations. The investigation of switching interfaces was beyond the scope of this research but the reader is referred to other texts on this subject (Alty 1984).

3.7.2 Task Analysis

Examination of the present interface to mailbox systems revealed that the options offered were too complicated for the expected user population. The best way to avoid overwhelming the user is to limit the functions available. In the case of the mailbox systems a cursory analysis shows that there are four main tasks a user performs regularly (Walgreich 1982). These are logging on, sending letters, reading letters and logging off. This segmentation is similar to that proposed by Panko (1981).

3.7.3 Requirements Analysis and Performance Evaluation

Because of the vast differences between individuals no one single dialogue will be acceptable to all users. The aim of the system designer should be to produce a dialogue which is compatible with the majority of users. It is therefore important for the system designer to work closely with the intended user population to develop the dialogue design around the particular psychological characteristics of the user (Kidd 1982) and analyse user interaction to new technology. (McCann 1983). In this design the starting point was a market analysis of the user needs as described in Chapter 2. The evaluation of the prototype is given in chapters 5 and 6.

3.7.4 Interactive Dialogue

The most common misconceptions users have when using computer systems is that either they think that the system is more intelligent than it really is (one reason not to use natural language) or that it does not perform the necessary tasks (because the information provided by the interface is insufficient). These two problems are mainly due to the user starting with an incorrect or incomplete model of the system and the system dialogue not projecting the appropriate model to enable the user to correct his model. With a user initiated dialogue, such as that provided by mailbox systems now, the conceptual model is not explicitly conveyed by the system. The designer must ensure that the dialogue enables the user to develop the correct model, this is most easily done using computer initiated dialogues (Martin 1973, Hammer et al. 1983, Stevens 1983, Kidd 1982, James 1981).

3.7.5 Output Devices and Techniques

The hardware will not be discussed here but below are some potentially important suggestions for dialogue design from the literature.

1) Feedback and response times. A maximum of 2 seconds is recommended between an action and a visible response of the computer. Although at the closure of a task the user may accept a

longer response time. This is especially important for text communications systems where response times are slow and interim responses may be required.

2) Short term memory support is important but the user should not be overloaded with too many options. The interface should only present the relevant options especially for the naive user.

3) The interface should be forgiving and encourage the user to explore the interface and reduce any anxiety about making mistakes.

4) The interface should be easy to use, i.e. efficient to use, and pleasant to use. The use of colours, segmentation, help messages, menu, forms and question/answers should all be considered.

5) The interface should be consistent and promote the correct conceptual model of the system. The conceptual model of a mailbox system is based on the office desk (Miller and Vallee 1980, Sandberg 1979, Mcquillan 1981, Welsh and Wilson 1981 and Ulrich 1980a)).

These basic suggestions were incorporated in the interface and then evaluated using naive and experienced users.

3.7.6 Input Devices and Techniques

The keyboard was considered the best present input device to text communications system. More extensive research in this area was beyond the scope of this report but future implementations should take into account the possible use of new interface tools such as the light pen, handwriting tablets, the mouse etc., especially in the case where the user is unfamiliar with the keyboard.

3.8 Conclusions

An intelligent interface is best adopted locally firstly on the grounds of the problem of the I/O bottleneck and secondly because of the proliferation of multiple mailboxes. The use of an intelligent terminal means that the user is less likely to have difficulties with many access procedures, dialogues, and nuances. These comments apply to other general text communications systems where an intelligent terminal can be used to make a generalised interface to each TYPE of system.

The solution proposed is to use an intelligent terminal as a prototype interactive user agent designed to present a unified computer initiated dialogue style to current computer mediated communications systems. The aim of this design was to demonstrate the feasibility of imposing an alternative dialogue and structure (based on certain guidelines) on existing systems, and to make possible a comparative evaluation of the efficiency of message transactions using each of the two different user interfaces to carry out standard tasks. The next chapter describes the features and characteristics of this user agent and then the results of the assessment of the two interfaces with hearing and hearing impaired people are reported.

C H A P T E R 4

AN IMPLEMENTATION OF AN INTELLIGENT FRONT END PROCESSOR (IFE)

4.1 Introduction

The ideal intelligent terminal would be designed to communicate with a standard CCITT X400 mailbox system. No existing system implemented the message submission/delivery protocol embodied in the CCITT message transfer agent at the time this work was carried out. The intelligent interface within the front end processor therefore had to communicate directly with a mailbox system by emulating input from a user at a dumb terminal and trapping and translating the responses from the mailbox system. Implementing the interface presented numerous error recovery problems since data transmission errors are frequent and no error correction facilities are normally provided as part of the data communications link to mailbox systems.

The mailbox system used in the trials is described below and some of its inherent problems are discussed. The conceptual models used in the interface are also described and related to the interface description.

4.2 Description of the Mailbox System

The Intelligent Front End (IFE) was developed to communicate directly with a commercial mailbox system. Systems such as Telecom Gold, COMET, PRESTEL, PSS mailstream and BLEND were investigated. Telecom Gold was used for the experimental work because of its rich repertoire of commands, wide availability and reasonably friendly interface. Also the use of Gold had been proliferating on a large scale - the number of Gold subscribers is expected to exceed the number of PRESTEL subscribers by 1987. Furthermore Gold had been used in the VISICOM project and a deaf user group of about 100 people still existed and provided an excellent trial group with which to compare the two interfaces.

Gold is more than a mailbox system and offers other facilities such as databases, noticeboard, telexes etc. During initial development a fairly full implementation of the facilities on Gold was attempted.

Subsequent informal analysis showed that many of the facilities were seldom used, so the intelligent interface was only developed to provide a limited, though frequently used, set of commands. For clarity these functions are described below in terms of a typical user session which is given in more detail in the user manual in Appendix 8. For a complete description the reader is referred to the Telecom Gold manual (1983).

Gold is an example of a commercial mailbox system which is based on a centralised computer network of PRIME host computers running message system software. These computers allocate memory space to each user - this is the mailbox - within which the user is allowed to compose, edit, send etc. messages. The user may access the mailbox system from a variety of computers from mainframes to microcomputers.

Access to the host is achieved by the users computer via two modems, one at each end of the communications link. By using manual or automatic dial, the user's modem connects the local computer to the host either directly or via the packet switching system (PSS). The three data rate standards used are V23, V22, or V21 (see Appendix 2). If access is via the PSS network then the PSS protocol must be transmitted followed by the Gold access protocol (which comprises the call sign, ID and password). If access is direct to Gold only the Gold access protocol is required and it is this method that was used in the assessment experiment. Once connected, the system acts like a typical mainframe accepting commands via a command line interpreter.

The user enters Telecom Gold at the system command prompt (">"). The user must first access the mailbox by entering MAIL c/r (carriage return) after which the prompt 'Send, Read or Scan:' is displayed. If the user enters the SEND c/r command, he/she enters the recipients names or numbers at the 'To:' prompt. The user then enters the title of the letter at the prompt 'Subject:'. These two fields comprise the envelope information. After entering these fields the user is prompted with 'Text:' and enters the letter content. The user terminates the letter and sends it using the command '.SEND c/r'.

The SCAN c/r command allows the user to examine all the envelopes received by the mailbox and then the user can choose to READ specific ones or all of them. Most of the commands can be abbreviated and new abbreviations can be added to personalise the mailbox commands. The mailbox session is finished either by entering the termination command ("OFF" c/r) or by dropping the line.

4.3 Problems with the Existing System

Telecom Gold is an example of a system which implements a simple direct mode of dialogue (user-initiated) and is typical of many computer mediated communications system which have evolved from the mainframe computer which is accessed by dumb terminals. The terse dialogue results from the requirement to maintain concise communications in order to maximise information flow along low data lines and ensure fast use of the central processor. While this type of dialogue is fast for experienced users, such systems require significant training and regular use to maintain proficiency and impose a high memory load even on intermittent expert users. Most users find they build up a subset of commands they know and understand (and have been able to find!) and keep to these.

If computer mediated communications systems like mailbox systems are ever to achieve wider acceptance then simpler computer initiated dialogue styles will be required. The allocation of the user interface function to the intelligent front end processor is proposed as a possible solution to this problem.

4.4 IFE Processor Hardware

The hardware was constrained to a simple microcomputer because one of the objectives was to investigate the development of an intelligent terminal affordable by the projected user population. The user interface was design and evaluated on a BBC microcomputer model B with proprietary modem (described in more detail in Appendix 3) partly because of the limited resources available but also because of its flexibility as a research machine. Although it would have been better to do the system development on a more powerful machine there was no machine available at the time that would allow the software to be transferred to the BBC later. Furthermore the BBC existed in many schools, deaf and hearing, and so this could have made the assessment easier if a complete self-contained evaluation package could be developed. However in the event this did not take place. For reasons of screen clarity and memory limitations the screen size chosen was 25 lines by 40 characters.

4.5 IFE Processor - Conceptual Models

Before designing the interface the conceptual models, upon which the design was to be based, were developed. The interface must project these models in order to correct or complete the user's model since the user's model could be different to that of the system. Some potential implementation problems are as follows.

1) The user is unlikely to be aware of the concept of being offline (i.e. only making communication with the modem locally) and online (i.e. communicating 'through' the modem with the host computer). The user does not have a model for this and so the proposed metaphor is to relate 'logging on' to opening a mailbox and 'logging off' to closing (or locking) a mailbox.
2) The office memo analogy will be inappropriate for the non-business user and so the proposed model is the idea of the post-box (to send a letter) and a letterbox (to read a letter).
3) The user's model for each type of communications system will be different. Therefore for each type of system different models must be projected. This is particularly important in the synchronous (point-to-point) model since in Gold there exists a pseudo point-to point system called CHAT where the end-users actually communicate through Gold. A possible analogy which could be suitable for the latter is where the user sees the calls being routed through an internal exchange in the mailbox system.
4) An analogy of a batch system is the active mailbox where the system is set up to transmit and receive certain letters like a human secretary. For a naive user this may be inadequate and a more suitable analogy such as an answerphone may be required.

Time precluded the full investigation of all of these metaphors and the following discussion and evaluation concentrates on the first two metaphors. To implement the required computer-initiated dialogue style, the user interface was designed around a hierarchical menu structure. This hierarchy (of which only the top levels are shown in figure 4.1) was chosen not on the basis of the Telecom Gold structure but from analysis and ranking of user's requirements within a mailbox system. Therefore the hierarchical level of activity within the local front end interface did not match directly the access level within Telecom Gold.

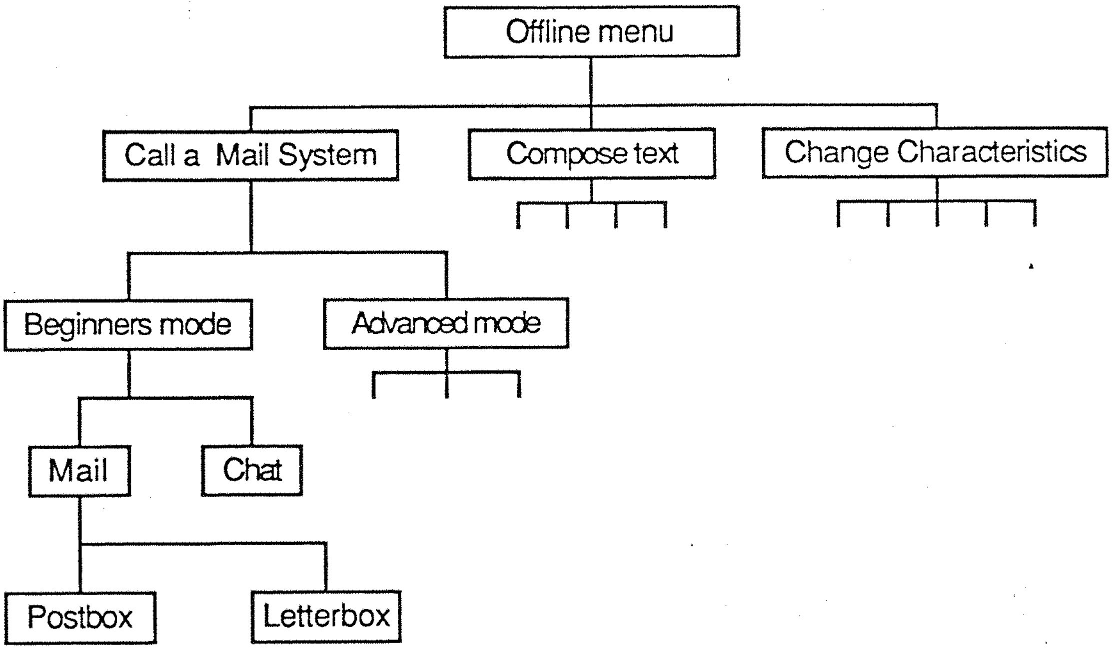


Figure 4.1 The intelligent interface menu hierarchy.

Since there are no dialogue markers transmitted by the system to indicate the level of activity a finite state machine had to be used to keep track of the dialogue state between the local and central processor. This design was general purpose and can easily be modified to drive and respond to any other mailbox system. Most of the features developed to project the conceptual models are easier to use than to implement. To illustrate the complexity of these features they are described in the following sections in terms of state diagrams and a "structured English" style reminiscent of PASCAL. The algorithms are cross referenced to a pseudo BASIC version of the actual programs in Appendix 4. In the algorithms an underlined phrase means that a separate algorithm is given in another figure.

4.6 The IFE Processor - Interface Description

In general menu selection was accomplished using a split-screen technique: the lower part of the screen displayed the allowable range of menu selections while the remainder of the screen allowed letters, envelopes, messages etc. accessed from the mail system or originated locally to be displayed. The content of the menu window depended upon the position within the menu hierarchy at that point. Menu selections were translated by the IFE interface into appropriate Telecom Gold commands, and command responses interpreted to produce subsequent menu displays. An advantage of using local intelligence is that since the menus are all stored locally, this display process is very much faster than can be achieved with a centralised menu-based system such as PRESTEL which communicate at CCITT V23. In addition to extensive use of menus, a form-filling dialogue was used for addressing envelopes to constrain and guide the user as to allowable address formats.

4.6.1 Online/Offline Model

The online/offline model provides a good model to describe the change in state when logging onto the mailbox system. If the user can be taught to accept this metaphor then this negates the need to develop a more longwinded and possibly less suitable model. To project the change in state of the system from offline to online two distinct interfaces are presented to the user. The offline mode is characterised by a menu which lists the options and shows the state of the system (this was not evaluated and so is not described here). The help messages use the bottom 6 lines of the screen and have the format,

Lines 1 and 2	double height state indicator, yellow text on blue.
Lines 3, 4 & 5	three lines of help information or instructions.
Line 6	a blank line or a line for user input.

When the mailbox has been opened a different display format is used and the status indicator 'online' is given in the bottom corner of the screen. The online messages have the format described below (e.g. figure 4.2 on page 44).

Line 1	The menu or message header is displayed in yellow text on blue background. The last 6 characters are blue on white and serve as feedback for each keypress.
Lines 2 - 5	The menu selections are displayed in blue on yellow. Later systems used red to make the keypress required

more distinct.

Line 6 The status line (white on red) to display the line state, printer state and time elapsed.

The remainder of the screen (top 19 lines) allowed information accessed locally or from the mailbox system to be displayed. This change of display format should encourage the user to associate going 'offline' with closing the mailbox. Other online messages use a slightly different format an example of which is given in figure 4.32 on page 62.

4.6.2 Dialling a Number

In the original design two functions were used to display the modem front page containing the eight telephone numbers. The only difference is that one uploaded the numbers up from the modem via the RS423 line whereas the other uploaded the previously stored page from disc, the latter version being displayed faster. The user selected the system characteristics required prior to dialling the number. To simplify the dialling task in the evaluation the interface was set up to enter Telecom Gold only. The user was required to press the space bar and the page of phone numbers was then displayed from disc.

The deaf user is helped by the cursor flashes under the number being dialled (one of the facilities copied from the proprietary software). The information supplied by the modem on the line status (dialling, waiting and no reply) is supplemented with local help messages as in the figure 4.2.

The intelligent terminal controls the mediation between the local terminal and the distant computer using the Telecom Gold protocols. In this way it synchronises the local terminal state machine to that of the host. Interim messages are produced at various points of the logon procedure to keep the user informed of the progress. These are particularly important to the deaf user who requires the audio signals (host modem acceptance tones, dialling tones etc.) to be translated to visual signals. These are also important for naive hearing subjects who may become confused by blank screens. The interim messages are,

1) figure 4.2	with dialling replaced by waiting. Here the intelligent interface is waiting for the host modem to respond.
2) figure 4.5	the host modem has responded and the next stage is in progress.
3) figure 4.6	the host has responded to the call sequence,
4) figure 4.7	the host has responded with the correct prompt and the ID and password. are being sent.
5) figure 4.8	this message instructs the user to wait until the 'dumb terminal' state has been entered.
6) figure 4.9	this message instructs the user to wait until the "intelligent interface' state has been entered.

If the call fails then the format and content of the message reassures the user that the call has not been connected and so no charge has been incurred (figure 4.10).

4.6.3 Software Amendments

The only offline function that was evaluated was the logging on task. Other facilities such as the compose, auto timer and system characteristics models were developed but not evaluated and are described in appendix 5. Extra screens were included to enable the researcher to setup certain parameters such as the subject name, interface to test and the task order. Other screens (figures 4.3 and 4.11) are required to inform the user of the procedure to follow and when to start the assessment. These last screens were made of a completely different format to any of the others so the user realised when the experiment started and finished. Figure 4.12 shows the algorithm of the assessment overview and figure 4.13 and 4.14 show the implementation of the logon task for both interfaces. The user logged on manually using the dumb terminal interface and automatically using the IFE interface.

4.6.4 Intelligent Interface Online Functions

Figure 4.1 shows the online hierarchy of functions. On entry to Telecom Gold the welcome banner is received and displayed followed by the first level of the menu structure (figure 4.4). The banner

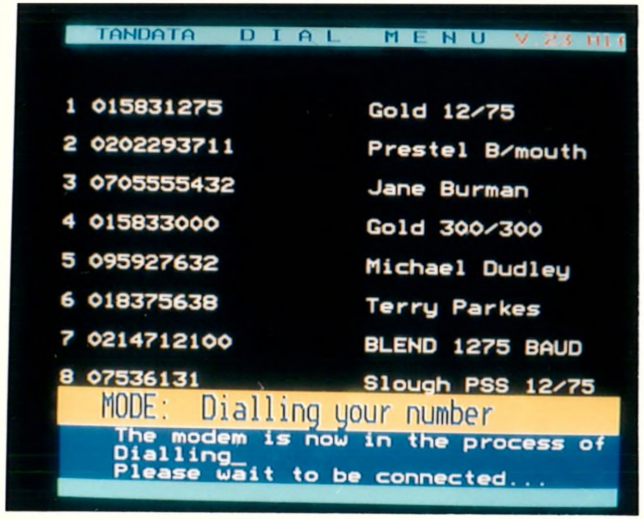


Figure 4.2 Directory screen

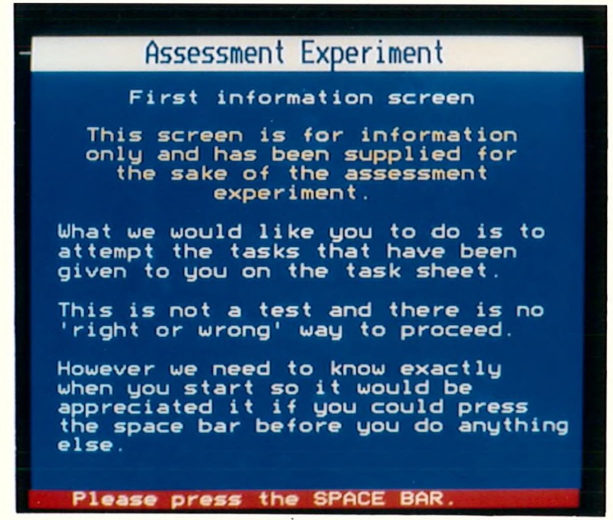


Figure 4.3 Assessment screen

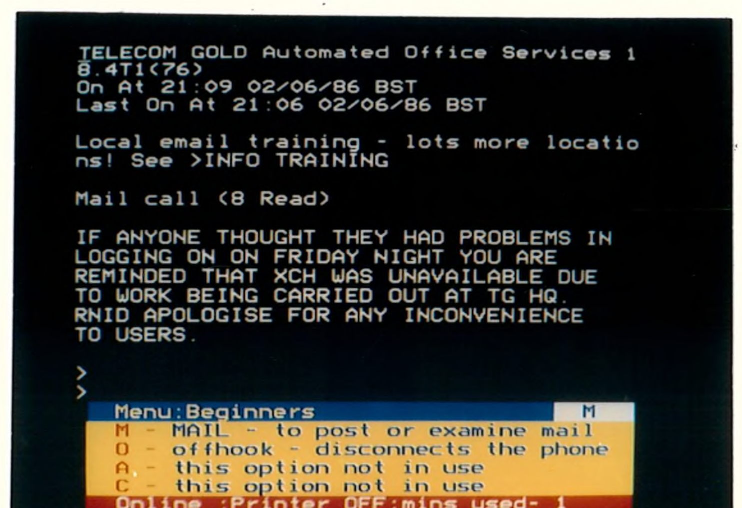


Figure 4.4

Beginners Menu

Figure 4.6

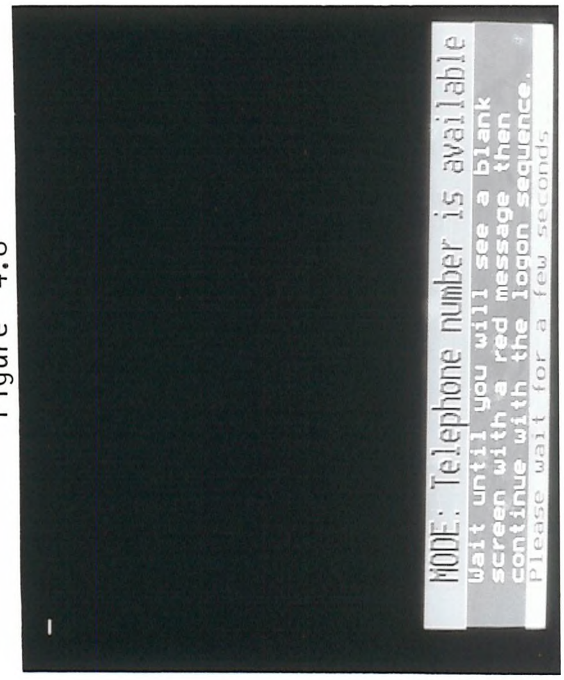


Figure 4.8

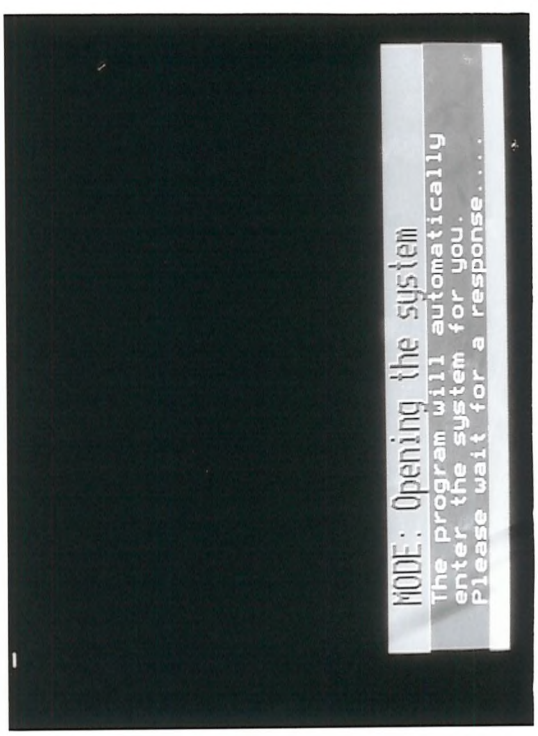


Figure 4.5

Figure 4.7

'Logon' task messages

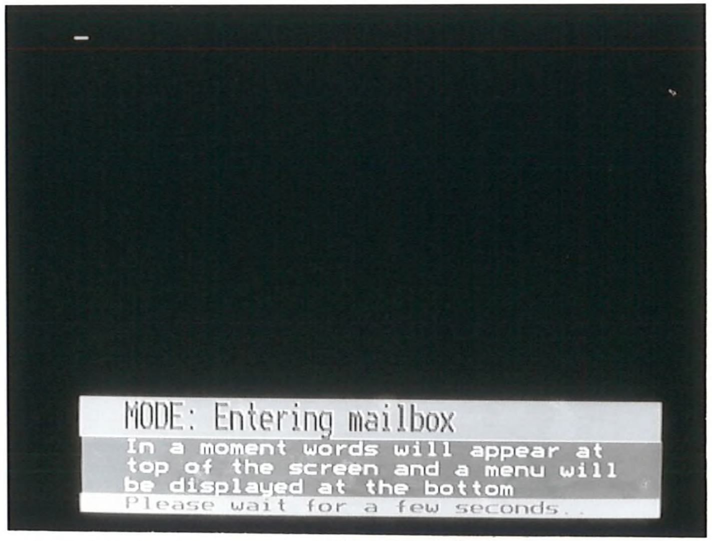


Figure 4.9

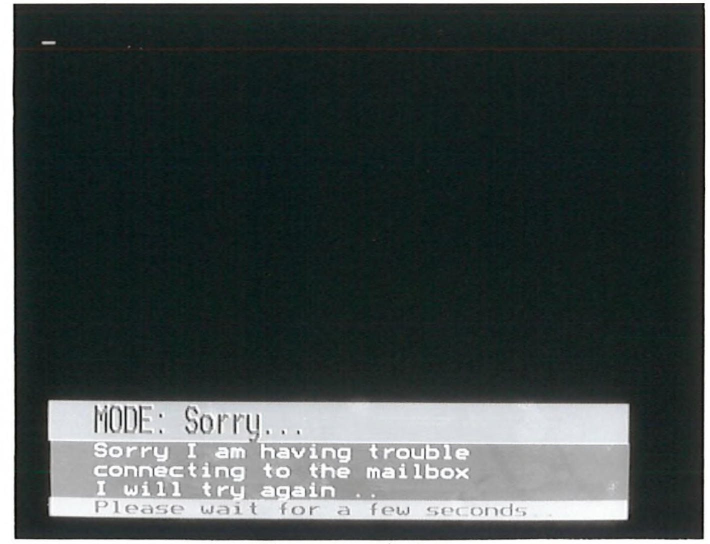


Figure 4.10

More 'Logon' task messages

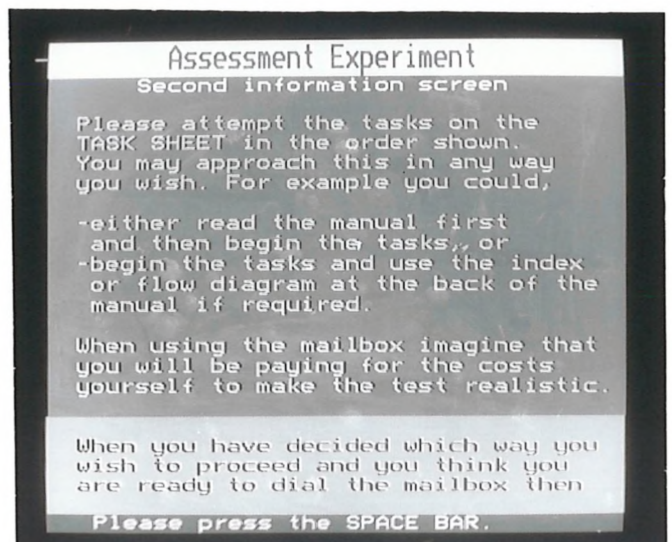


Figure 4. 11
Second assessment screen

```
BEGIN {Overview of assessment experiment}
   start of experiment = TRUE
   REPEAT
     BEGIN
      IF NOT start of experiment
       THEN
         BEGIN {display re-entry information}
           display redial information screen (figure 4.18)
           when user presses SPACE BAR save interaction on disc
         END
      ELSE IF start of experiment is TRUE
       THEN
         BEGIN {initial screens}
           researcher will enter subject name, interface and task sequence
           open data file and store name and interface on disc
           IF logon task is first
             THEN
               BEGIN {introductory comments}
                 display first information screen (figure 4.3)
                 when user presses SPACE BAR save interaction on disc
                 display second information screen (figure 4.11)
                 when user presses SPACE BAR save interaction on disc
                 start of experiment is FALSE
               END {introductory comments}
           ELSE IF logon task is NOT first
             THEN
               BEGIN {screen for researcher}
                 display second information screen (figure 4.11)
                 when researcher presses SPACE BAR save interaction on disc
               END
         END {initial screens}
     END
     BEGIN {start the assessment}
       connect user to TELECOM GOLD modem and mailbox
       IF start of experiment is TRUE THEN introductory comments as above
       REPEAT
         Use {dumb terminal interface} or {intelligent interface overview}
       UNTIL either user OR researcher terminates session
       IF researcher terminates session THEN start of experiment is TRUE
     END {start the assessment}
   UNTIL forever
END{Algorithm of evaluation introduction}
```

Figure 4.12 Algorithm showing overview of experimental task procedure

```
BEGIN {connect user to TELECOM GOLD modem and mailbox}
   Display modem phone numbers, from disc file, on screen.
   REPEAT
     Display message (figure 4.2) and save interaction on disc
     Dial TELECOM GOLD number and display cursor on screen
     REPEAT UNTIL 3 seconds has elapsed OR local modem has responded
     IF local modem has responded
       THEN
         BEGIN {PROCtry_to_connect}
           REPEAT
             receive (rx) characters and wait 1/10th sec
             in case transmission has stopped briefly
           UNTIL host stops transmitting
           REPEAT
             REPEAT rx characters with delay UNTIL host stops (as above)
           UNTIL a clear screen has been received AND host has stopped
           IF host responded with a clear screen
             THEN access mailbox using correct interface (figure 4.14)
         END {PROCtry_to_connect}
       ELSE IF host has NOT responded
         THEN
           BEGIN
             Save interaction point on disc, display message (fig. 4.10)
             reset modem and wait for a response
             IF NO response
               THEN display error message, REPEAT UNTIL forever
           END
   UNTIL forever
END {Connect user to TELECOM GOLD}
```

Figure 4.13 Algorithm showing the implementation of dialling procedure

```
BEGIN {Access mailbox using correct interface}
   IF dumb terminal AND (LOGON task first OR NOT start of assessment)
     THEN
       BEGIN {PROCdec_1}
         display wait message and CLOSE all disc files
         display message (figure 4.8), load and run the ONLINE program.
       END {PROCdec_1}
   ELSE (establish link with TELECOM GOLD mailbox)
     BEGIN {PROCtry_connect}
       Display message (figure 4.5) and wait 2 seconds
       REPEAT
         Send wake up characters (c/r's - but dependent on host).
       UNTIL > 20 characters received OR time elapsed
                 since beginning of procedure is > 7.5 seconds
       IF < 20 characters received THEN host NOT responding
       IF host responding
         THEN
           BEGIN {PROCenter_assess}
             Display message (figure 4.6) and wait for 1/2 second
             REPEAT
               Enter characters from host into a text_string variable
             UNTIL no more characters OR length text_string > 159
             IF text_string length < 20 then host NOT responding
             IF host responding and text_string contains "Telecom Gold"
               THEN
                 BEGIN {DEFPROCcon_enter}
                   display message (figure 4.7)
                   Transmit access protocol from file
                   Transmit HALT transmission chars and display figure 4.9
                   load and run ONLINE program to access host.
                 END {PROCcon_enter}
               ELSE host NOT responding
           END {PROCenter_assess}
     END {PROCtry_connect}
END {access mailbox using correct interface}
```

Figure 4.14 Algorithm showing implementation of access to the mailbox.

consists of mostly unused information except the time of entry and the mailcall function which displays the number of letters (seen and unseen) the mailbox has received for the user. The beginners menu allows only four options.

1) **A** - Advanced mode

In addition to providing access to the most commonly required mailing functions via a menu hierarchy, an option was made available to go into an "advanced user" state. Access to the advanced user mode is user-initiated but there is debate as to whether the transition should be user- or computer initiated.

In this mode the software reverts to the "dumb terminal" operation allowing obscure commands not directly implemented via the menu to be accessed. In this mode the user has complete control over the interaction with Telecom Gold. Other facilities were provided including downloading of data from the host to a local file, uploading data which had been preprepared locally to the host, allowing a connection to a printer and the disconnect feature.

In the assessment these extra facilities were not evaluated and the "dumb terminal" mode was reduced to a simple design with the only extra function the disconnect option. The screen was setup to have 24 lines for the Telecom Gold messages with the last line just instructing the user to use the f6 key to disconnect. The experimental subjects were specifically instructed to use this option to disconnect rather than using the OFF c/r command. The same message used in the intelligent interface disconnect (O) option was used to confirm the logging off request. This message takes up 5 extra lines and will overwrite some information on the screen in the 'dumb terminal' mode so if the user selects the f6 option and then decides not to go offline then the screen is restored. The implementation of the dumb terminal in the assessment is shown in figure 4.15.

```
BEGIN {Dumb terminal interface}
   Display "Press f6 to disconnect " in white text on a red background on
   bottom line of screen and assign top 24 lines for user window
   REPEAT
     REPEAT receive characters UNTIL host stops OR a key has been pressed
     IF a key has been pressed
       THEN BEGIN
              Get a key and save interaction on disc
              IF key is NOT a function key THEN transmit key pressed
              ELSE IF the key is f6 THEN user must confirm the logoff
            END
   UNTIL user confirms logoff OR researcher terminates session
END {dumb terminal interface}
```

Figure 4.15 Algorithm of the dumb interface to TELECOM GOLD

2) **O** - Offline (disconnect option)

When this option is selected and the user positively responds to the confirm message (figure 4.16) then the interface deals with the disconnect procedure to close the mailbox by displaying a message (figure 4.17) and dropping the telephone line. In retrospect a better choice of option may have been **L** - **L**ock and close mailbox.

In the assessment once the subject has logged off a further help screen in displayed (figure 4.18) to request the user to begin the logon sequence again because the task order is different for each group of subjects.

3) **C** - CHAT

This option allows the user to communicate directly through Telecom Gold. This was not evaluated and is described briefly in appendix 5.

4) **M** - enter the **M**ailbox.

This option initiates the send or read tasks using the intelligent interface. In the assessment the subjects were only allowed to use either the dumb terminal OR the intelligent interface and were not allowed to move between the two. An overview of the implementation of the intelligent interface is shown the algorithm in figure 4.19. This

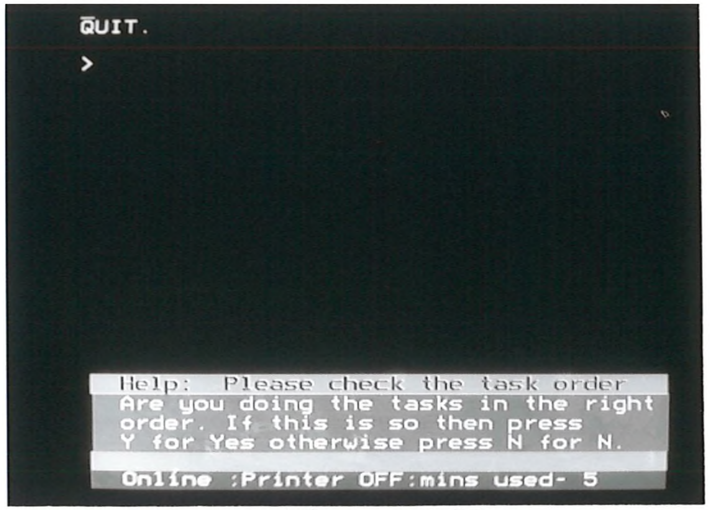


Figure 4.16

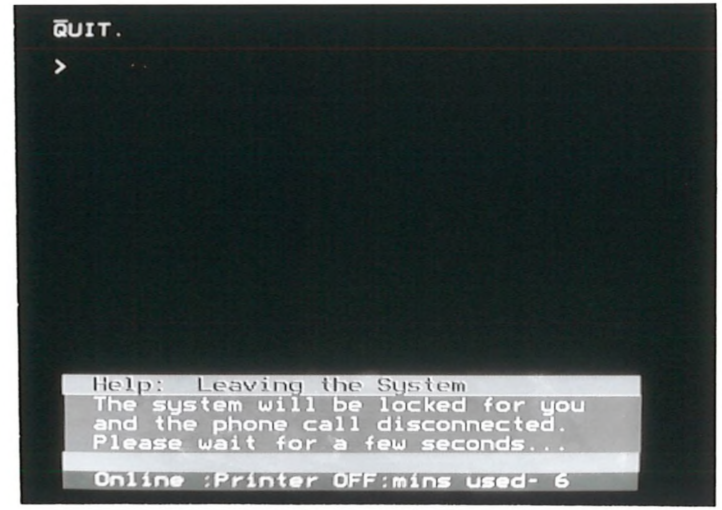


Figure 4.17

'Logging off' messages

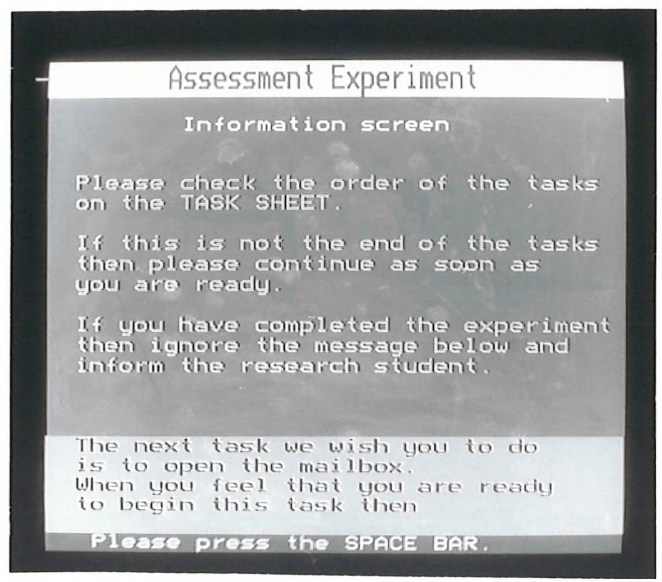


Figure 4.18

Redial screen

```
BEGIN {Intelligent interface overview}
   display command menu (figure 4.4) and initialise variables
   REPEAT
     REPEAT receive characters UNTIL host stops transmitting
     IF a key has been pressed
       THEN
         BEGIN {PROCcommand_select}
           get a key and save interaction on disc
           IF key is "O" THEN user must confirm logoff
           IF key is "M"
             THEN
               BEGIN {PROCmail_mode}
                 set host to command level (figure 4.45)
                 display mail menu (figure 4.21) and initialise variables
                 REPEAT
                   REPEAT receive chars UNTIL host stops transmitting
                   IF a key has been pressed
                     THEN BEGIN
                            get a key and save interaction on disc
                            IF key is "R" THEN read letters
                            IF key is "S" THEN post letters
                            IF key is "P" THEN scan letters
                          END
                 UNTIL key is "Q"
                 display command menu (figure 4.4)
               END {PROCmail_mode}
         END {PROCcommand_select}
   UNTIL user confirms logoff OR researcher terminates session
END {Intelligent interface}
```

Figure 4.19 Algorithm of the intelligent interface to TELECOM GOLD illustrates that when the M key is selected the system is forced to the command level in case the system has entered some spurious state. This takes about four seconds and is discussed in detail later. The mail menu (figure 4.20) is then displayed where each option has more explanation than the Telecom Gold interface at this point. It was hoped that the analogies in each section of the Send, Read and Scan menus would promote the idea of a letterbox and postbox but in retrospect it may have been better to make this more explicit and subdivide the first mail menu into 'letterbox' for the read and scan commands and 'postbox' for the send command.

4.6.5 Letterbox Model

It became apparent from informal trials and direct observation of some subjects using the existing interface that there could be confusion with the operation of the read task. After entering send c/r, the user is prompted for qualifiers whereas with the read c/r command there is no prompt which would enable the user to select a particular letter. This qualifier must be given in the same command line as the read command. For example 'READ FROM FRED' c/r, will begin to read the letters from fred, whereas 'READ' c/r causes all the letters to be read in the mailbox. To obviate this the intelligent interface prompts the user for a qualifier after selecting P(SCAN) or R(READ) (figures 4.21 and 4.23 respectively). Only after the qualifier has been selected were the appropriate Gold command is transmitted and the reply interpreted (see later). (A benefit of the users seeing the transmitted command is that they are taught how to use the command language interpreter).

The options given on SCAN were limited to expanded SCAN or QUICK SCAN, a condensed version of SCAN. It may be also prudent to include the command QUICK SCAN UNREAD (which summarises the envelopes of all the unseen letters) at a later stage as this also seems a well used command. Figures 4.24a and 4.24b shows the implementation of the scan function. When scanning a letter the command is transmitted without a terminating c/r. The c/r is then transmitted and after receiving a page full of information the last response line is checked for the prompts. From the algorithm it can be seen that there are two possibilities either, the prompt is recognised as one of the end of

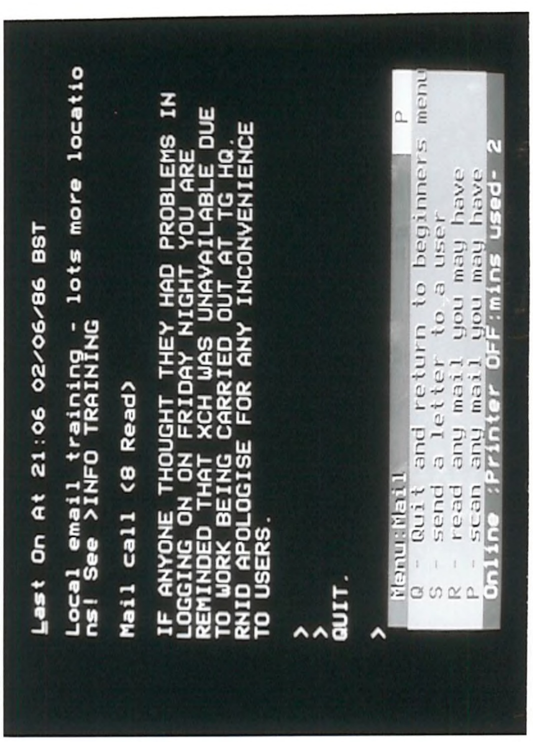


Figure 4.20 Mail Menu

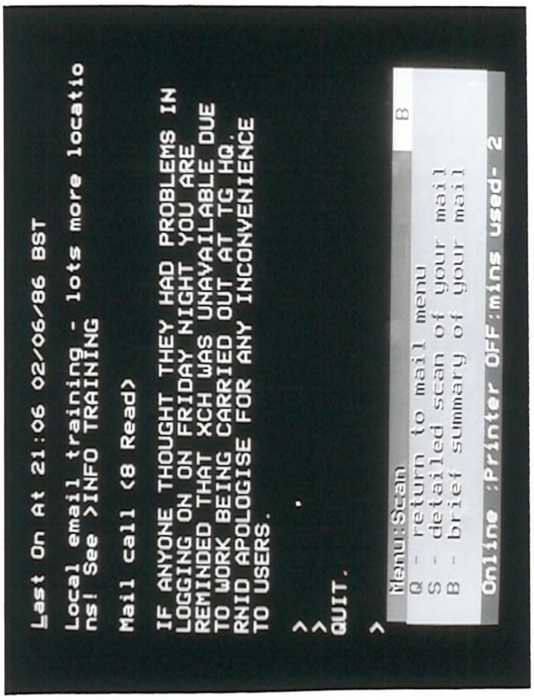


Figure 4.21 Scan Menu

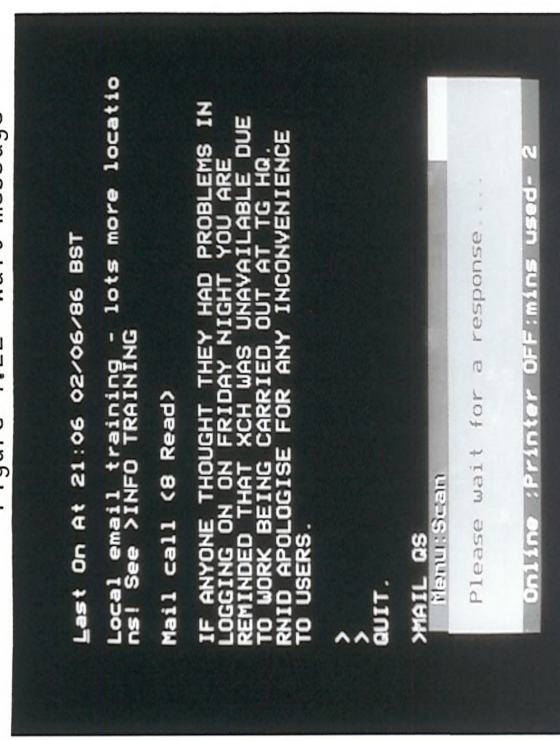


Figure 4.22 Wait message

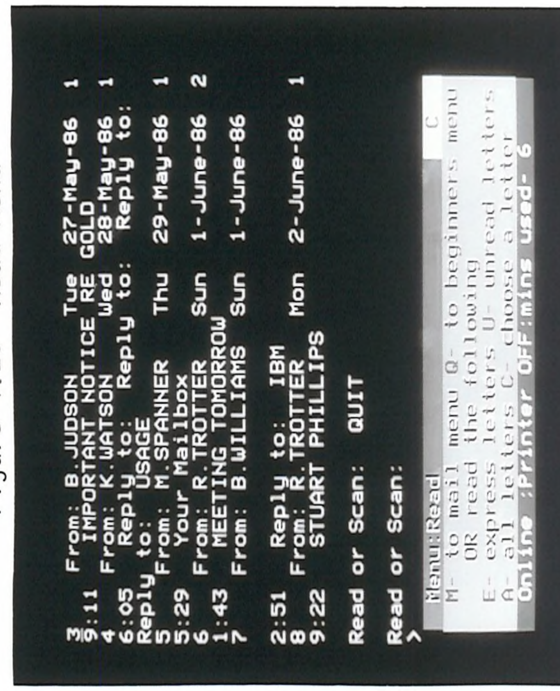


Figure 4.23 Read Menu

```
BEGIN {scan letters} [PROCscan_mode]
    display scan menu (figure 4.21) and initialise variables
    REPEAT
      REPEAT UNTIL a key has been pressed
      Get the key pressed and save interaction point on disc
      IF key pressed is "B" THEN transmit "MAIL SCAN" without a c/r
      IF key pressed is "S" THEN transmit "MAIL QS" without c/r
      IF key pressed is "B" OR "S"
        THEN
          BEGIN [PROCascert][PROCverify]
            transmit a c/r
            display continue menu using SCAN as header (figure 4.22)
            REPEAT UNTIL more than 3 characters have been received
                         OR 7.5 seconds have passed.
            REPEAT
              check last host response (figure 4.46)
              IF last line of the response from GOLD was NOT ">" AND
              NOT "Send, Read or Scan:  " AND NOT "Read or Scan:  "
                THEN unexpected prompt (figure 4.27)
              ELSE IF response$=">"
                THEN set key to "Q"
                ELSE
                  BEGIN
                    set key to "Q" and display message (similar to fig. 4.22)
                    REPEAT
                      transmit "QUIT" c/r and break command to host
                      check last host response
                    UNTIL response is the command line prompt ">"
                  END
            UNTIL key is "Q" OR there is no response from the host
            IF there is no response from the HOST THEN continue or not?
          END [PROCascert][PROCverify]
    UNTIL key is "Q" OR subject wishes to continue
    display mail menu
END [PROCscan_mode]
```

Figure 4.24a) Algorithm to show the implementation of the SCAN function

scan prompts (">", or "Send, Read or Scan: ") in which case the system is forced to command level. If neither of these prompts are transmitted then there are more letters to scan and the unexpected prompt routine is entered.

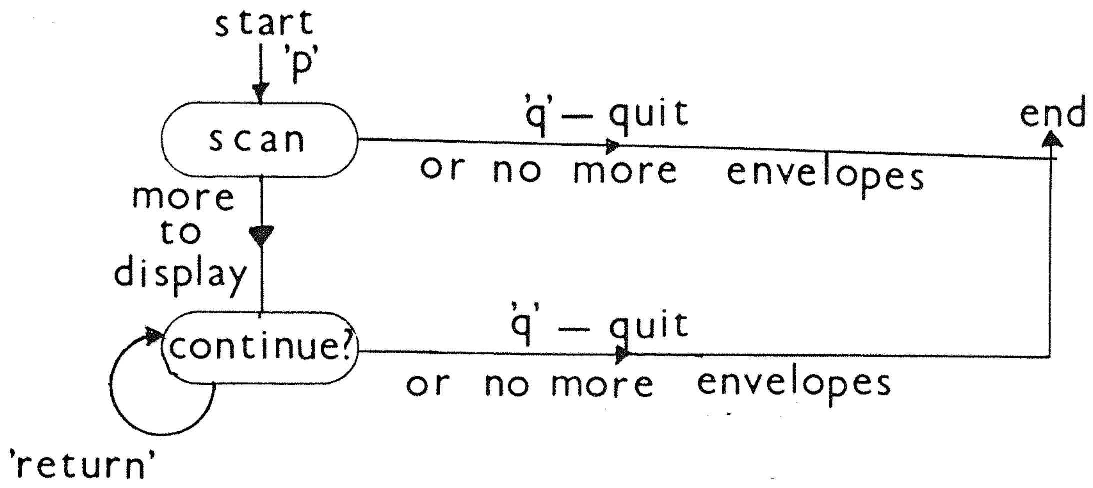


Figure 4.24b) Scan task interaction event state diagram

After **R** (**R**ead) is selected in the mail menu the allowable options are to read important letters (EXPRESS), any unseen letters (UNREAD), to choose a letter by number (CHOOSE) and to read ALL the letters. The interaction event state diagram is shown in figure 4.24a and the implementation is shown in figure 4.24b.

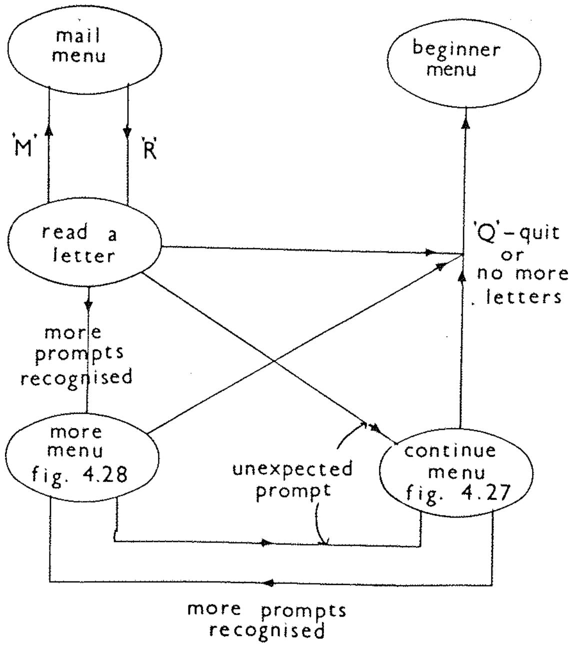


Figure 4.25a) Read task interaction event state diagram

```
BEGIN {read letters} [PROCread_mode]
   display read menu (figure 4.23)
   REPEAT
     REPEAT receive characters UNTIL host stops transmitting
       IF a key has been pressed
         THEN
           BEGIN [PROCread_command_select]
             Set variable 'chars' to zero
             Get a key from keyboard and store interaction point on disc
             IF key is "A"
               THEN BEGIN
                     Read all letters (transmit "MAIL READ" & c/r)
                     Set variable 'chars' to 9
                    END
             IF key is "E"
               THEN BEGIN
                     Read all important letters
                     (transmit "MAIL READ EXPRESS" & c/r)
                     Set variable 'chars' to 17
                    END
             IF key is "U"
               THEN BEGIN
                     Read all unseen letters
                     (transmit "MAIL READ UNREAD" & c/r)
                     Set variable 'chars' to 17
                    END
             IF key is "C"
               THEN
                 BEGIN [PROCchoose_a_number]
                   display message (figure 4.31) and set up window
                   read in number of letter to read then restore screen
                   IF letter to read is NOT "Q"
                     THEN BEGIN
                           Read that letter
                           (transmit "MAIL READ + letter to read" & c/r)
                           Set variable 'chars' to 9
                          END
                     ELSE set key to "Q"
                 END [PROCchoose_a_number]
             IF variable 'chars' is non-zero
               THEN look at transmission from host [PROCread('chars')]
           END [PROCread_command_select]
   UNTIL key is "Q" OR "M"
   IF key is "M" THEN display mail menu and reset key variable
END [PROCread_mode]
```

Figure 4.25b) Algorithm to show the implementation of the read task

After selecting a read option, the relevant command is transmitted, a wait message is displayed and the response from Gold is interpreted to set the menu level (figure 4.26). If there is more information to be displayed one of the menus in figure 4.29 or 4.30 is displayed. The implementation of these lower levels are given in figures 4.27 and 4.28. Otherwise the host is reset to command level and the mail menu is displayed.

```
BEGIN {look at transmission from host} [PROCread(chars)]
   display wait message (similar to figure 4.22) using READ as header.
   REPEAT UNTIL 7.5 seconds have passed OR more than chars + 2
   characters have been received
   REPEAT
     check last host response (see figure 4.46)
     IF the response was the command line prompt ">"
       THEN set key to "Q" and save interaction point on disc
     ELSE IF response is "Send, Read or Scan:  " OR  "Read or Scan:  "
       THEN
         BEGIN
           save interaction point on disc and set key to "Q"
             REPEAT
               transmit "QUIT", c/r and break command to HOST
               check last host response
             UNTIL response is the command line prompt ">"
         END
     ELSE IF response is NOT "--More--" AND NOT "Action Required:  "
       THEN unexpected prompt response (Figure 4.27)
     ELSE IF response$ is the "--More--" OR "Action Required:  "
       THEN more prompt response (Figure 4.28)
   UNTIL key is "Q" (i.e. response is "Send, Read or Scan:  " OR ">")
END [PROCread(number)]
```

Figure 4.26 Implementation of routine to set menu level in read task.

A special case is the choose option (figure 4.31) where the command is not sent immediately but the user must further qualify the request by specifying the number of the letter to read which is obtained from the list of previously scanned mail. The command is then sent to read this

```
BEGIN {unexpected prompt response} [PROCunexpected_prompt]
   display continue menu with "Continue" header (figure 4.29)
   REPEAT get a key and save interaction on disc UNTIL key is c/r OR "Q"
   IF key is c/r OR "Q" THEN display wait message (similar to 4.22)
   IF key is "Q"
     THEN
       BEGIN
         REPEAT
           transmit break command, "QUIT", and to host
           check last host response (figure 4.46)
         UNTIL response is the command line prompt ">"
       END
     ELSE
       BEGIN
         Transmit c/r and display wait continue menu
         REPEAT UNTIL > 1 character received OR 10 seconds have passed
         IF 10 seconds elapsed THEN continue or not?
       END
END [PROCunexpected_prompt]
```

Figure 4.27 Algorithm showing the implementation of one of the READ or SCAN prompts

```
BEGIN {more prompt response} [PROCmore_action]
   Initialise key variable to null and display more menu (fig. 4.30)
   REPEAT
     IF a key has been pressed
       THEN
         BEGIN
           Get key pressed and save interaction point on disc
           IF key is "A" THEN Request letter again (transmit "A" & c/r)
           IF key is "N" THEN Request next letter (transmit "NE" & c/r)
           IF key is "RETURN"
             THEN Request next page of letter (transmit c/r)
           IF key is "D"
             THEN
               BEGIN
                 display message to ask user to confirm delete (fig. 4.32)
                 REPEAT
                   REPEAT UNTIL a key has been pressed
                   Get the key and save interaction point on disc
                 UNTIL key is "Y" OR "N"
                 IF key is "Y"
                   THEN Delete current letter (transmit "DEL" with c/r)
                   ELSE display the more menu (fig. 4.30)
               END
           IF key is "A", "N", "Y" OR "RETURN" THEN validchoice is TRUE
           IF key pressed is "Q"
             THEN
               BEGIN
                 QUIT to command level (transmit "QUIT" with c/r)
                 set host to command level (figure 4.45)
               END
         END
   UNTIL key pressed is "Q" OR a validchoice is TRUE
   display wait message (similar to 4.22)
   IF validchoice is TRUE
     THEN
       BEGIN
         REPEAT until host starts responding OR 7.5 seconds has passed
         IF NO response from HOST THEN continue or not
       END
END [PROCmore_action]
```

Figure 4.28 Algorithm showing the implementation of response interpretation at --More--/Action Required Level

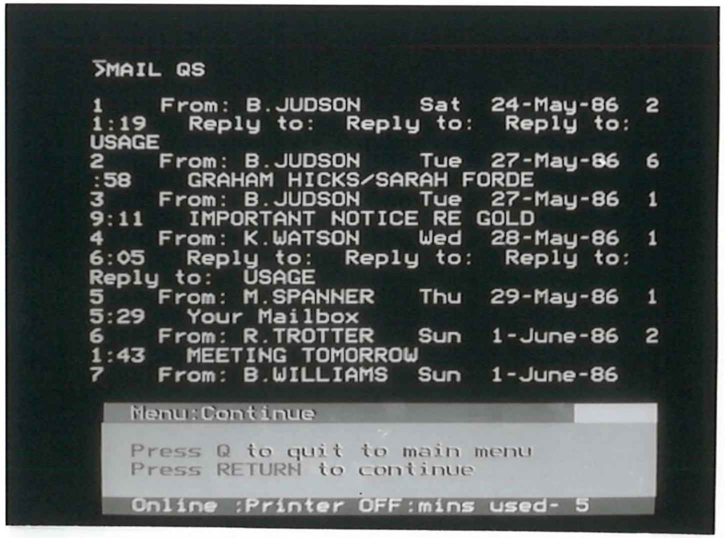


Figure 4.29 Continue Menu

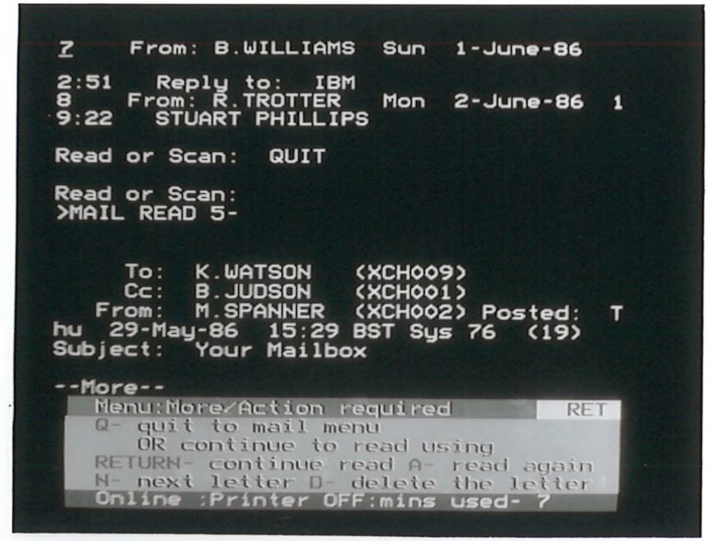


Figure 4.30 More Menu

Figure 4.31 Choose letter message

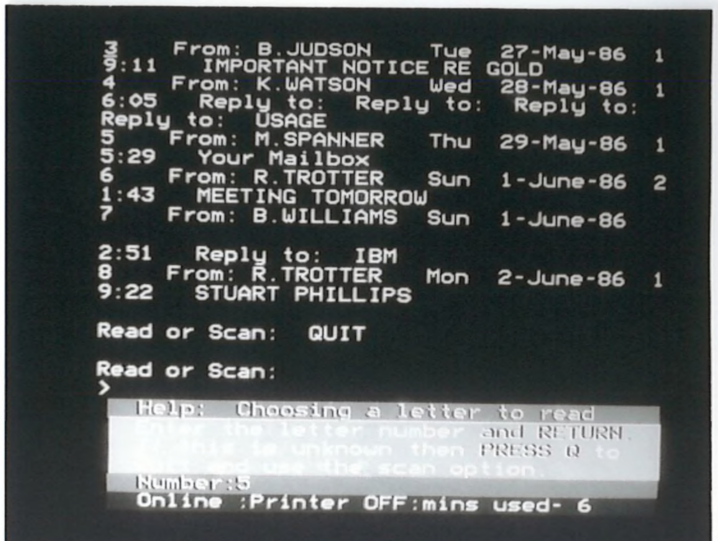
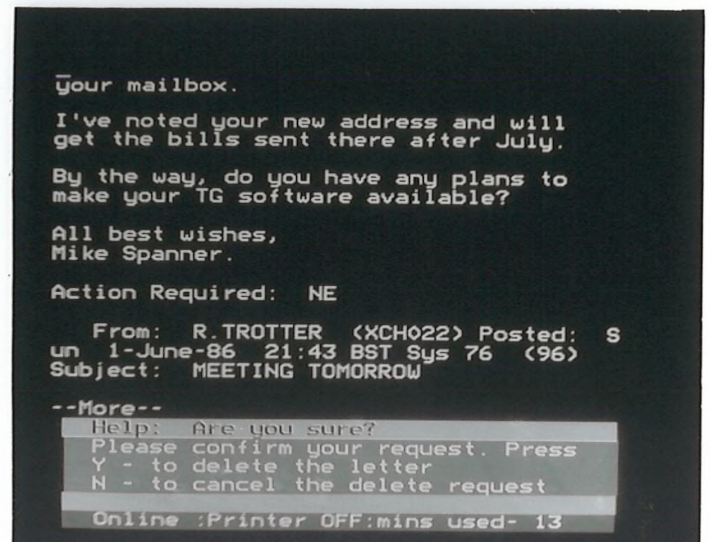


Figure 4.32 Delete confirm message



letter and the following letters ('MAIL READ n-' c/r). In retrospect it may be better to allow the user also to specify a letter in terms of the originator's name and also what it is about. This was not allowed in the prototype design to reduce the complexity of the interaction. The problems of prompt recognition are described later.

4.6.6 Postbox Model

The send task is split into three parts, the envelope composition and the text composition - which is very similar to the offline compose feature for consistency - and the post routine. The overview of this task is given in the interaction event diagram in figure 4.33

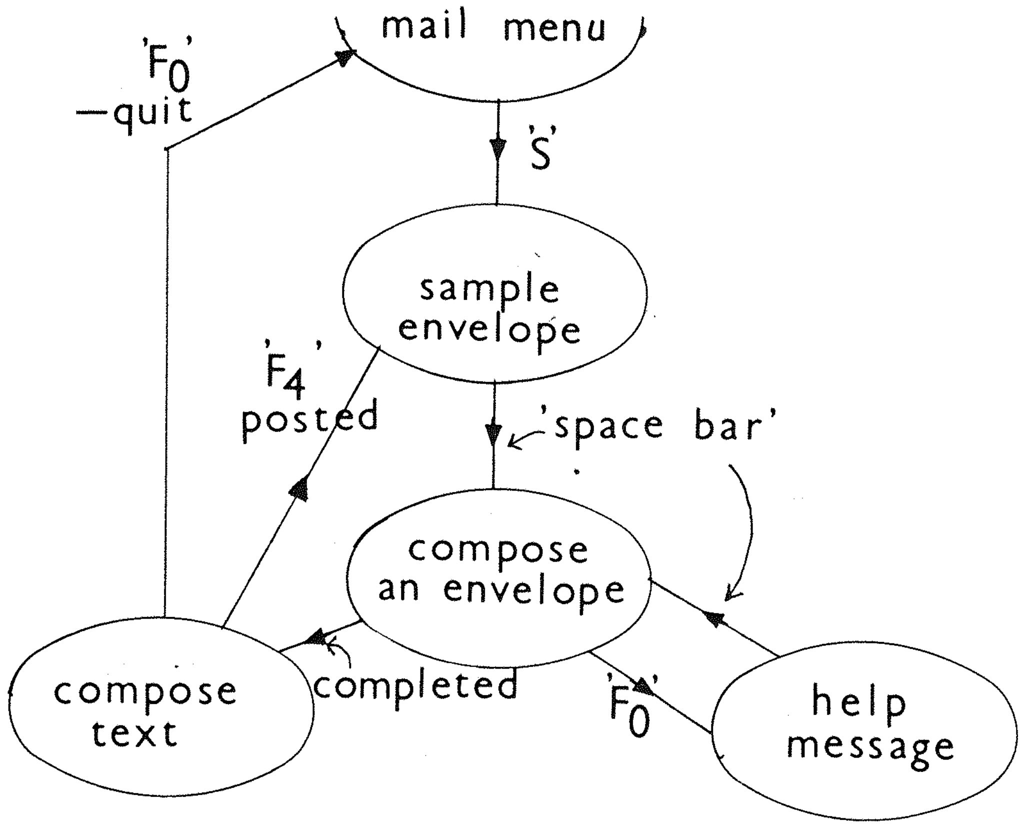


Figure 4.33 Send task interaction event state diagram

1) Envelope composition

This differs from the writing a letter analogy in two ways. Firstly a naive user would normally write an envelope after the content whereas in this case the envelope must be completed prior to the content. Secondly the Gold system allows the user to send a letter to a maximum of 500 people. In the intelligent terminal implementation of the envelope the user is allowed to send a letter to five recipients since the literature has shown that sending a letter to more than one person, with a maximum of 5, is common (Walgreich 1982). This is a good example of where the users model does not agree with the system model. If the interface was designed strictly to the user's model then some of the useful capabilities of the system would remain unused. The form model was chosen for the envelope because it linked easily to the system model and to the users model. For consistency this facility was made almost identical to the compose an envelope feature (see appendix 5) and is described in figures 4.34 and 4.35.

A sample envelope is displayed (figure 4.40) which indicates to the user the style and constraints of the envelope. The user proceeds to the creation of the envelope by pressing SPACE BAR and an empty envelope is displayed figure 4.36. Characters can then be entered, these must either be numeric and/or alphanumeric ('space' is only allowed in the title field). Help and error messages are displayed in the bottom part of the screen figure 4.41 and 4.37.

2) Text Composition

After completing an envelope, a wait message is displayed while the envelope is sent to the mailbox. The user is then asked to update any of the list that may have been incorrect. The software then allows the user to enter the contents (the text mode). The only edit function provided in this mode was delete a character as it was reasoned that it would not be necessary to provide complex editor functions because the naive user would not be familiar with word processing and so would not require a complex editor. No (.) commands such as '.SEND' c/r were allowed only Quit (f0), Display (f2) and Post (f4). This was to ensure that the interface program remained in complete control (figure 4.44). Any dot's <.> were preprocessed and sent as (<space><.>). A completed edit session is shown in figure 4.43 and the implementation is shown in figures 4.44.

3) Posting the letter

In the text mode the user posts the letter by selecting f4. The user is then informed by interim messages as to the progress of the letter (figures 4.39). The user is returned to the sample envelope after the letter has been posted (see figure 4.34). In retrospect, it may have been better to inform the user that the letter has been posted and then give the option to the user to leave or continue to compose.

```
BEGIN {post letters} [PROCletter]
   REPEAT
     display sample envelope (figure 4.40)
     REPEAT UNTIL space bar is pressed and record interaction on disc
     Initialise certain parameters and display letter envelope (fig. 4.36)
     REPEAT
       Set valid_character to FALSE and set up cursor in correct field
       REPEAT
         Get a key when one is pressed and save interaction on disc
         If key is 'f0'
           THEN
             BEGIN
               display help message (fig. 4.41)
               REPEAT UNTIL space bar, save interaction and restore screen
             END
       UNTIL key is not f0
       test key for validity
       IF valid_character THEN clear the error message
         ELSE
           BEGIN {PROCerror_message}
             Use 1/4th sec delay to avoid screen flash
             display "Error Message"
             IF key is c/r and there is no subject name
               THEN Print "You need a topic AND at least one name"
             IF key is left or right arrow
               THEN Print "Left/right movement not allowed" (fig. 4.37)
             IF key is delete and cursor at star of field
               THEN Print "No character to delete"
             IF key is up arrow and cursor in first field
               THEN Print "You must move down from this field"
             IF character is <=" " AND > "~"
               THEN Print "This character is not allowed"
           END
     Save interaction point on disc
     UNTIL END_ENVELOPE is TRUE OR START_SEND is TRUE
     Reset some variables
     IF START_SEND is TRUE
       THEN
         BEGIN
           display message (similar to 4.22) and set host to command level
           send envelope [PROCenvelope]
         END
   UNTIL END_ENVELOPE is TRUE OR key is "Q"
   display wait message and set host to command level
END {post letters}
```

Figure 4.34 Algorithm showing implementation of envelope messages

```
BEGIN {test key for validity}
   IF key is "f1" THEN valid_character is TRUE and key is "Q"
   IF key$ is "c/r" OR "down arrow"
     THEN
       BEGIN [PROCcheck_entry]
         IF there are no characters in field
           THEN BEGIN
                  IF NOT in title field
                    THEN valid_character is TRUE and go to title field
                  ELSE IF key is down arrow and in title field
                    THEN valid character is TRUE and go to first field
                END
         ELSE
           THEN
             BEGIN [PROCcheck_and_move]
               IF NOT in title field
                 THEN valid_character is TRUE and move to next field
               ELSE IF cursor is in title field AND key is down arrow
                 THEN go to first field
               ELSE IF in title field AND key is c/r AND at least a
               subject and title THEN START_SEND is TRUE
             END
       END [PROCcheck_entry]
   IF key is "delete" AND NOT start of field
     THEN delete character and valid_character is TRUE
   IF character is between " " AND "~" AND in title field
     THEN add character to title field and valid_character is TRUE
   IF character is between "!" AND "~" AND NOT in title field
     THEN add character to current field and valid_character is TRUE
   IF key is the up arrow key AND NOT at top subject field
     THEN  BEGIN
             valid_character% is TRUE
             REPEAT go up fields UNTIL a NON blank field OR top field.
           END
   IF valid_character is TRUE and field is full AND NOT title field
     THEN move to next field
END
```

Figure 4.35 Algorithm to test validity of envelope key selection.

Figure 4.36

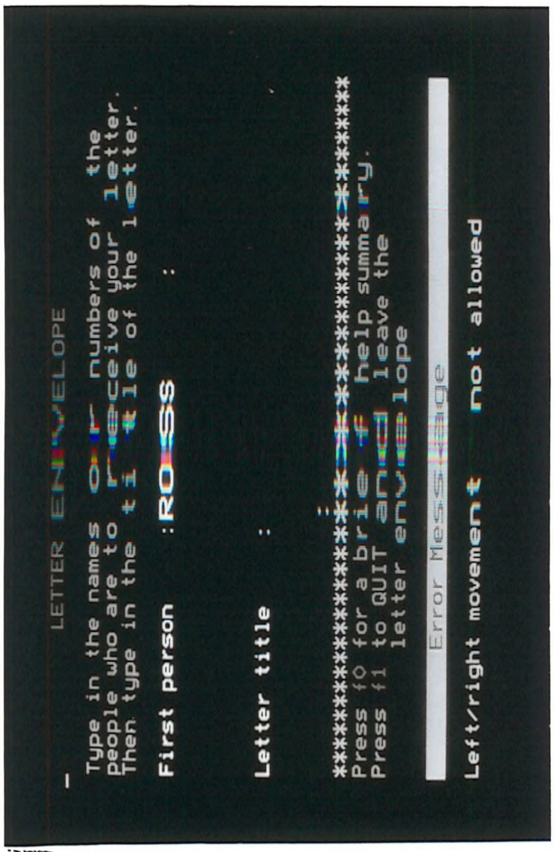


Figure 4.37

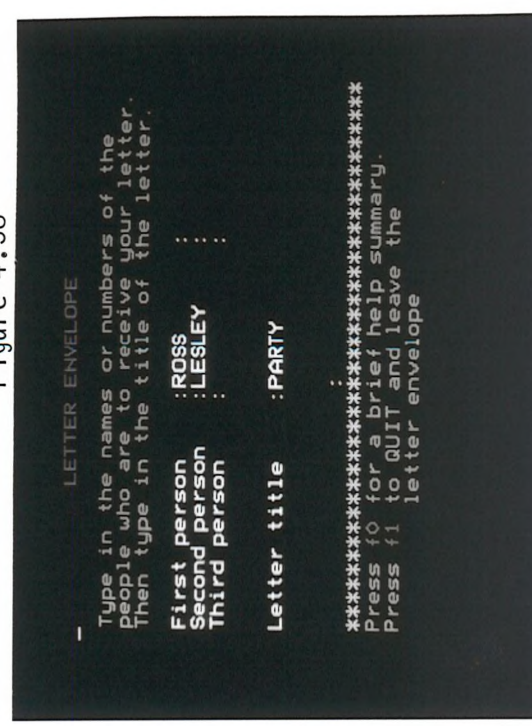


Figure 4.38

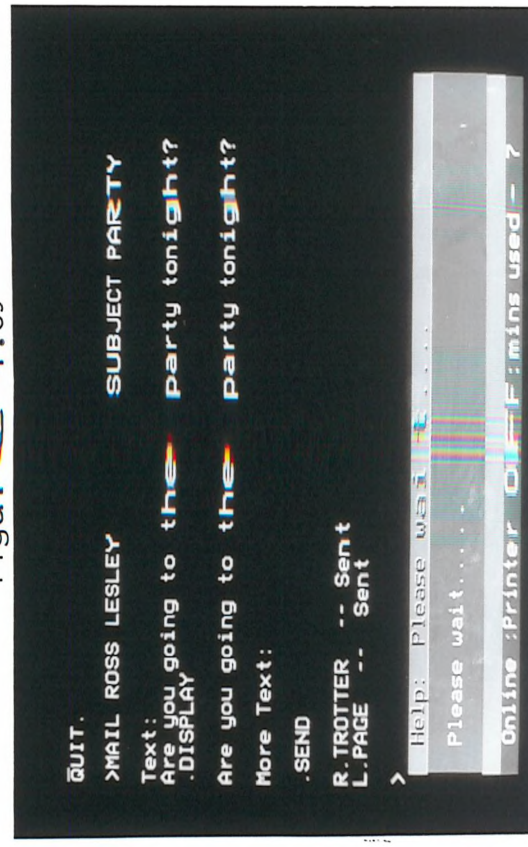


Figure 4.39

'Send' task messages

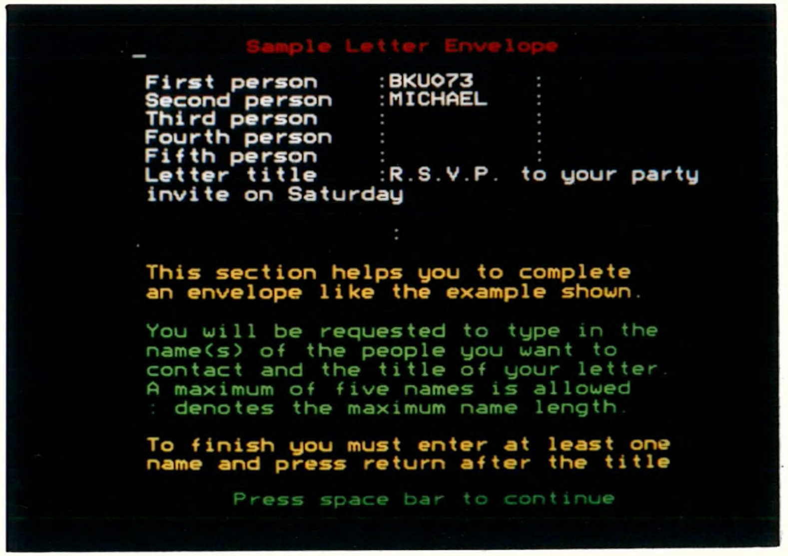


Figure 4.40 Sample letter envelope

Figure 4.41 Envelope with help message

Figure 4.42 Send menu

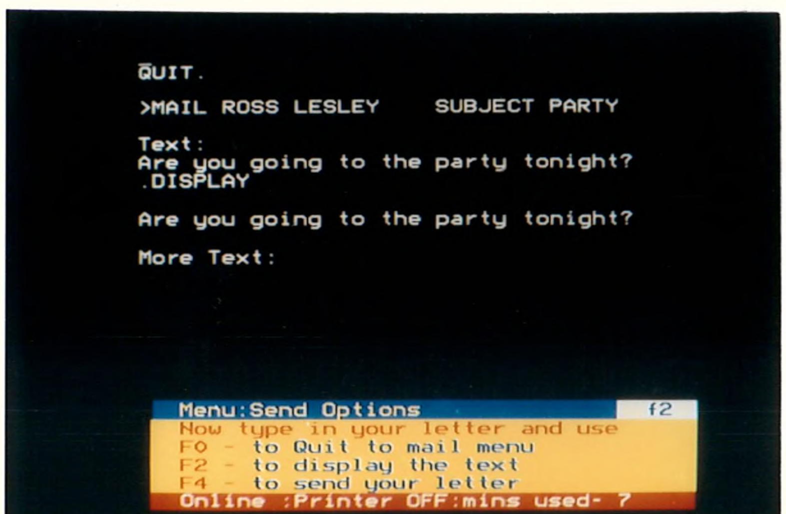
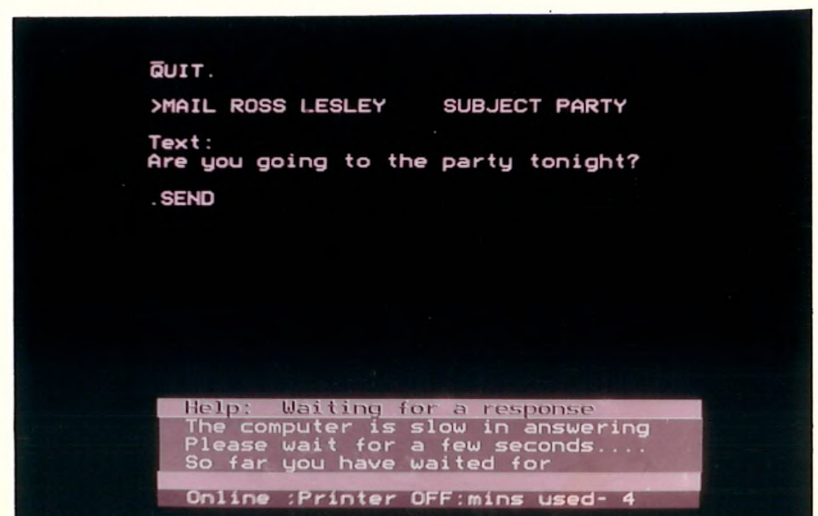


Figure 4.43 Wait message



```
BEGIN {send envelope} [PROCenvelope]
   Save interaction point on disc and transmit the envelope
   REPEAT prompt host with c/r and wait UNTIL host has responded
   BEGIN [PROCcheck_input]
   REPEAT
     display wait message (similar to 4.22) and initialise variables
     REPEAT UNTIL > 3 chars have been received or 7.5 seconds passed
     REPEAT enter characters from host into text_string UNTIL no more
     IF length of text_string >12 AND text_string does not contain
                                    the text "Send" OR "Read" OR "Scan"
      THEN
       BEGIN [PROCrespecify]
         display respecify message and save interaction on disc
         REPEAT get key and save interaction UNTIL key is "Y" OR "N"
         IF key is "N" THEN transmit ignore name code (i.e. c/r)
           ELSE enter and transmit new name or ignore name
       END [PROCrespecify]
   UNTIL length of text_string < 13 characters OR text_string does NOT
                      contain the text_string "Send" OR "Read" OR "Scan"
   IF length of text_string > 7 characters AND "ext" is in text_string
     THEN
       BEGIN [PROCwrite]
         display send menu (figure 4.42)
         REPEAT
           REPEAT receive characters UNTIL HOST stops transmitting
           IF key has been pressed THEN
             BEGIN
               Get a key and save interaction on disc
               IF key is "f0", "f2" OR "f4" then display key on screen
               IF key is "f0" THEN transmit quit command (".QUIT" c/r)
               IF key is "f2" THEN transmit display command (".D" c/r)
               IF key is "f0" THEN transmit post command ("SEND" c/r)
               IF key is "." THEN transmit " "
               IF key is c/r OR between " " and "~" inclusive
                 THEN transmit key
             END
         UNTIL key is "f0" OR "f4"
         IF key is "f0" THEN key is "Q" ELSE set host to command level
       END [PROCwrite]
     ELSE BEGIN
           save interaction on disc, transmit quit command (".quit")
           set host to command level (figure 4.45)
          END
END [PROCenvelope]
```

Figure 4.44 Algorithm of the implementation of send task after the envelope has been composed.

4.6.7 Other Considerations

Due to the memory limitations of the BBC a simple facility was included in the program to allow various sections of the program, held on disc, to be overlayed onto the top of the main program. Overlays were also used to display certain screens and a further prototype should make more extensive use of this facility. Effectively this added virtual memory to the BBC on top of the resident memory. The time to access each disc overlay was about 2 seconds so an interim response (..Please wait .. flashing message) was used.

4.7 Changes in the Software for the Evaluation

To produce an incisive experiment a small set of the available facilities were chosen. The only parts of the interface that were tested, were the logon, send and read tasks.

To support the evaluation experiment, addition software was added to the intelligent interface and the 'dumb terminal' interface to monitor the users activity in accessing the mailbox system. This software was transparent to the user and simply logged each event during the message transaction on a disk file together with a time code derived from a real time clock on the BBC. This disc log could subsequently be printed out so that accurate measurements of the time to carry out individual tasks could be made. The logging consisted of saving a key press, the time code and, in some cases, the lower five lines of the screen containing some of the information sent from the host. The BBC is relatively slow in saving data on disc but an informal test showed that the data logging did not substantially affect the results (see Chapter 5).

4.8 Problems in Implementation

The menu displays were produced from the interpreted command responses and so there had to be close coupling between the intelligent terminal and the mailbox system. The ensuing problems are typical of the text communications in general.

Firstly the slow speed of operation of the BBC computer meant that there were problems receiving information at the fast data rates so the XON/XOFF protocol was used to control the information flow into the local buffer. Secondly there is no error correction provided on the host to microcomputer link and so the line is susceptible to line noise and connection noise (from Strowger exchanges). This problem will decrease in the future with the installation of electronic exchanges and the use of local PSS nodes to access mailbox systems.

Thirdly apart from the command prompts, no extra information is transmitted to enable the intelligent terminal to detect the state of the system. If there were some dialogue markers, invisible to the user, this would enable the local front end to detect the position of the system in the dialogue hierarchy and set the menu level. It is unlikely that such markers will be developed as a standard as this would require standardisation of dialogue structure in the mailbox systems, something which the CCITT are not considering. (Schicker 1982).

Synchronising the menu levels within the intelligent interface therefore had to be done by using contextual information received from the mailbox computer prompts. When recognised, each prompt is used to set the software to the correct level in the software hierarchy and the appropriate menu is displayed. For example, when reading mail, the input is checked for the contextual 'end of mail' messages ("Send, Read and Scan: " and the command prompt ">"). If these are detected then the software forces the host to command level (the ">" prompt) and the relevent menu is displayed.

Difficulties in detection were compounded by,

1) the variation in response times of the host. This is dependent on many factors including the number of people using the system.
2) the fact that the contextual information has a finite possibility of occuring in the text of a letter, (since the contextual information also consists of ASCII characters), and
3) the potential problem of the mailbox computer forcing disconnection or logging the user off when there has been no activity (timing out). This latter problem occurs because, ultimately the user is in control and may elect to do nothing.

To overcome similar problems Kaye and Mcdowell (Chapter 3) used long timeouts in their batch implementation of an intelligent interface. This is reasonable in a terminal which is communicating with the mailbox system on the users behalf but is obtrusive when the user is actually communicating with the system directly.

The implementation of the set host to command level is shown in figure 4.45. The reason for the use of the two types of quit command is that it appears as if the host could be in two possible conditions outside the command level. Either the system could be scanning or composing letters in which case the break command (transmitting ASCII codes 16 and 66) will be sufficient to force the system to the command level or the system could be reading letters in which case the break command would only set the system to the "Action Required: " prompt so a further "QUIT c/r" is required.

After one of the break commands is transmitted text is read in from the host into a string and displayed on the screen and the last character used to detect the system prompt. In retrospect it may be better to investigate the whole string for the command prompt since extra spurious characters can be sent. This routine usually takes between 4 to 10 seconds to reset the host to command level.

The routine check last response line (figure 4.46) is used to determine what text is contained in the last line of the response transmitted by the host. When scanning and reading letters the Telecom Gold host provides other messages to prompt the user. However these prompts have some inconsistencies between tasks.

Within Telecom Gold there exists a parameter file, initialised by the system manager and under the control of the user, from which the host makes decisions as to how to format the display. For example, this file determines how many lines are transmitted before an extra prompt is displayed. For the purpose of the prototype system, since we were using the bottom six lines of the 25 line by 40 character screen for menus and messages, the parameter file was initialised to 19 lines of no more than 39 characters per line. These parameters apply specifically to the format of messages sent from the user mailbox but not to the messages received from other users which presents some

```
BEGIN {set host to command level} [PROCset_host_to_command_level]
   display message (similar to 4.43) and initialise loop
   REPEAT
     IF the loop variable is ODD THEN transmit break command
       ELSE transmit "QUIT c/r" and break command
     wait for two seconds and increment loop variable
     BEGIN {PROCwaitnsee}
       REPEAT UNTIL 7.5 seconds have elapsed OR host sent a character*
       Initialise text string variable to a null string
       BEGIN [PROCstar]
         REPEAT
           read non-null character from host into text_string
         UNTIL host stops
           delay for 9/100th seconds
         UNTIL HOST has stopped transmitting
       END [PROCstar]
       Strip characters from text_string except the final character
   UNTIL text string is ">" (i.e. command level prompt has been received)
END [PROCset_host_to_command_level]
```

Figure 4.45 Algorithm showing the implementation of the routine to reset the HOST computer to the command level

```
BEGIN {check last host response}
   REPEAT
     REPEAT receive characters from host UNTIL HOST stops transmitting
     wait for 1 second in case transmission stopped briefly
   UNTIL HOST stops transmitting
   Make the text string variable to be used in the REPEAT loop null
   Remember cursor position and set cursor at first character of line.
   REPEAT
     Read a character at current cursor position into a text string
     move cursor one position to right
   UNTIL cursor is at original position
   REPEAT
     Check response is one of the expected responses
     These are
     ">" this indicates the command line prompt has been detected
     "Send, Read or Scan:  " OR "Read or Scan:  " this indicates that the
     Read and Scan tasks have been finished
     "--More--" OR "Action Required:  " this indicates that the Read task
     is still in progress
   UNTIL response is recognised OR response is NOT recognised
END
```

Figure 4.46 Algorithm to illustrate the implementation of the routine to read in the last line of the response from TELECOM GOLD.

problems.

This is best illustrated by an example. Say subjects A and B have parameter files set up to produce a format of 25 lines by less than 39 characters. If A sends a letter to B then when B reads this letter a "--More--" prompt will be displayed every 25 lines of text. If, however, subject B receives a letter from subject C who has a format of say 25 lines by 79 characters then when subject B reads a letter the host will stop transmitting after 25 lines but will not display the more prompt. Then after the user presses c/r the next 25 lines will be displayed and the "--More--" prompt is displayed.

In the read task there are two main prompts used and these are the "--More--" prompt and the "Action Required: " prompt. These allow the user to request similar options but the "--More--" prompt is given a) after the envelope has been displayed so the user can choose to move onto the next letter and b) after each successive page of 25 lines as mentioned above. The "Action Required: " prompt always appears at the end of the letter.

In the scan task, however, no "--More--" prompt is given. A page of envelopes is displayed and if there are more to come then transmission just stops. The user continues by pressing return and in fact can quit by transmitting the break command but this is not made obvious in the manual. If no more envelopes exist the "Read or Scan: " prompt is displayed.

This is inconsistent and confusing to the user. Hence, in the IFE interface, separate routines were developed for the "--More--" and "Action Required: " prompt and for the case when no prompt is given but should have been. These are called the unexpected prompt algorithm (figure 4.27) and the more action prompt algorithm (figure 4.28). The latter routine is used in the read and scan task whereas the other is only used in the read task. In retrospect it may have been better to use the unexpected prompt only in the scan task and the more action prompt in the read task. This is because, although Telecom Gold does not accept any of the "--More--" commands, if this prompt not has been sent then the system can be forced to the "Action Required: " level by transmitting a break command before sending the

"--More--" command (for example <break> AGAIN to read the letter again). The interface would thus provide a consistent menu throughout the read task.

Further work is required to determine the optimum conditions and commands to control more effectively the change of level in the dialogue hierarchy and to increase the speed of transition between levels to reduce possible user frustration.

4.9 Conclusion

The features and characteristics of the prototype interactive intelligent interface described in this chapter have been designed to present a unified computer initiated dialogue style to current mailbox systems. Although the prototype system did not exhibit optimum performance it was satisfactory for the trials.

The algorithms used to describe the implementation and the finite state machines used to match the dialogue state of the local interface to the central host processor were general purpose and can be modified to drive another mailbox system. It would seem better, however to modify the design slightly to develop a table driven finite state machine optimised for each system.

CHAPTER 5

DESIGN OF THE EVALUATION EXPERIMENT

5.1 Introduction - Experimental Population Requirements

Practical trials were conducted, using selected groups of subjects, to establish whether the intelligent interface was a significant improvement on the existing mailbox interface and hence whether further project development would be justified. This was investigated in terms of quantitative parameters and qualitative parameters. This type of design strategy is called prototyping.

Two groups of experimental subjects were used in the final assessment. The first group consisted of twelve hearing science under-graduates who were comfortable using computers but had no experience of commercial mailbox systems and so were naive mailbox users. The second group consisted of twelve deaf people who were users of Telecom Gold and had used the mailbox system over a substantial time period. These people were naive computers users but experienced users of Telecom Gold. Each group would test a different hypothesis.

Group A tested the hypothesis,

> Whether the interactive intelligent interface makes the use of mailbox systems quicker, easier and more satisfying than the existing interface for naive users.

Group B tested the hypothesis,

> Whether the use of the intelligent interface causes a decrease in the level of performance for experienced users.

The user groups were chosen because it was easier to classify users as experienced and naive rather than 'learners'. These two user groups were used to simulate the learning effects which would be obtained when using the intelligent interface for a prolonged period of time. Hearing people would have been suitable candidates as expert users but deaf subjects were used for the following reasons.

i) For deaf people text telecommunications provides one of the few methods of long distant contact by telephone and so this user group is a captive audience and many were keen to make themselves available for this trial.

ii) The deaf user is more representative of future consumer users than a business user.

iii) The author has had much contact with the deaf user group on Telecom Gold and so it was more convenient and easier to make arrangements with these subjects rather than unknown people.

iv) Many of the deaf people on the Gold network had experienced other text communications systems such as VISTEL and PRESTEL. These systems had never fulfilled their expectation (see chapter 2) and so these people would tend not to enter the trial with any illusions about the interface performance. These subjects were also more liable to provide critical information about possible improvements.

Some hearing impaired people have a low level of literacy as a result of their disability and this limits their suitability for the trial. From Eldridge et al. (1985), it was decided that the deaf group should contain young and middle aged people with a level of literacy sufficiently high to read text from a screen. Older people, in general, are less likely to use any computer system or keyboard input system, however two older subjects were found to be suitable.

Although a larger more diverse group of users could have been used the trial would have been more costly and time consuming. The selection of a certain type of user allows the statistical analysis to be more precise (see later). Therefore even though the experimental design is complicated by the use of the deaf user group (see below) it makes the trial more efficient in the use of resources (time and cost) and in gathering the most relevant information.

5.2 Performance Measurements

Any human factors study in user interfaces is looking for a statistical relation between the use of a new interface and improved performance over existing interfaces. The four standard variables normally used (reference) are the number of errors, training time, task time and job satisfaction. In this evaluation the quantitative parameters used were the time to do certain tasks. The time is composed of the training time and the task time since the user is learning while doing the tasks. The errors contributed to the total time to do the task and so were not treated separately.

The qualitative parameters of ease of use and user satisfaction were assessed by a post-test questionnaire. The two variables of time and user satisfaction are interrelated and this relationship would be more apparent if the trials were conducted over a longer period. Intuitively we would expect the user who found the interface harder and more frustrating to use to take longer to do the experiment. In a short test such as this it may not be possible to detect this from the quantitative parameters but the qualitative parameters may give a rough indication as to the possible trend.

A critical analysis of the use of mailbox systems reveals that, for most users, there are four main tasks which are performed regularly and these are,

1) entering (or logging onto) the mailbox,
2) sending a letter,
3) reading a letter, and
4) leaving (or logging off) the mailbox.

The last is simple and consists of a single action so only the first three tasks were used in the comparison tests.

a) In the logon test the subject initiated the mailbox session by requesting the software to dial the telephone number using the autodial modem. When the mailbox computer responded the subject then had to complete the access procedure to open the mailbox. When the subject was using the existing interface he entered the commands directly whereas in the intelligent interface the logon sequence was completed by the computer program. The total times to complete the logon task were calculated from the time to start the task until logon had been achieved. Any additional time caused by redialling or the mailbox automatically timing out were removed from the task times.

b) The send test involved the subject composing an envelope with the names of the person (or people) who were to receive the letter and a qualifier (called the subject) which identified the topic of the letter. After completion of the envelope the subject then entered the text of the letter and ended with the relevant termination code. The total time to complete the send task was calculated from the entry of the initial character required to initiate the send task to the end of the termination code used to post the letter. For both sets of subjects the time taken by a subject to enter any extra text, not specified in the task sheet, was removed. For the experienced subject group any time to read the online or offline manuals was removed.

c) The read test consisted of the subject finding and reading the text of a letter. The two letters used in both tests were placed in similar positions within the block of letters in the mailbox so that the letter could not be found easily. There were various methods of accessing this letter, a few of which are,

1) reading all the letters until the relevant one was found,
2) scanning through all the envelopes until the letter position was discovered, then specifically reading this letter,
3) as 1) except only reading the envelopes until the relevant letter was found and then reading the text of that letter.

The total times to complete the read task were calculated from the initial character required to initiate the read task until the text has been read. Since the context of the text in the send and read test were not of interest, the two tests used different names and text messages but both letters were of similar length. The exact procedure to complete all these tests with both interfaces is given in the manuals which are in appendix 8.

5.3 Experimental Design

It was not straightforward to design an experiment which would allow the effects of the different interfaces to be compared and the simple elimination of nuisance effects such as variation between subjects and learning effects. Hence assistance was sought in constructing a suitable design (see acknowledgements and Chatfield 1983, Shneiderman 1980, Hills and Armitage 1979).

The experimental design can best be understood by considering the sequence of design steps that were followed to reach the final design. This begins with a simple design based upon a test using two treatments (the two interfaces). Henceforth treatment a (the existing dumb interface of Telecom Gold) will be known as system a and treatment b (the intelligent interface) as system b.

5.3.1 Stage 1 Parallel Group versus Balanced Crossover Design

There are two main methods of conducting trials to evaluate two systems. One method is a parallel group comparison where each subject tests one system and the analysis is concerned with between subject comparisons. This method tends to be insensitive because of the typically large variation of response between subjects and so a large number of subjects may be required to have a high probability of detecting a substantial difference between the two systems.

The objective of the experiment is to obtain statistically valid results with the minimum number of subjects as quickly as possible and so an alternative method is required: this is the crossover design. Here each subject experiences both interfaces over separate but roughly equal time periods. For either method it is important to ensure that sufficient time is given for the effect of each treatment to become apparent but not long enough for the state of the subject on the learning curve to change considerably.

If each subject experienced the systems in the same order their experience of the first period might affect their performance in the second period and there would be no way of disentangling this effect from the system differences in the statistical analysis. Other factors such as fatigue and level of concentration may also change over time and hence affect the results. Therefore a design is required which takes into account these potential factors. Such a design is a balanced crossover design. Here two groups of people experience different sequences of interfaces: group I experiences interface a first and group II experiences interface b first. The advantages of this type of design are,

1) both groups experience both interfaces,
2) a comparison within subjects has greater precision than a comparison between subjects. Each subject is his own control and so fewer subjects are required to have a high probability of detecting a particular result, which in this case was desirable.

The disadvantage is that there may be a problem in isolating the system effects especially if there is an interaction between the system and period. However if this interaction is significant the design can revert to the parallel group design using results in the first period only, but with a consequent loss of precision.

It would not be sensible to compare the second period results since this would only give an estimate of the difference between subjects who had experienced the first period. The difference in observed times in the second period may be due to the systems or the potentially different state in which the subjects entered the second period. This interaction may be due to either a psychological reaction to the system experienced in period 1 which may persist to period 2 or the rate of change in response to the system by the subjects may vary according to the level of response.

5.3.2 Stage 2 Initial Design and Model

The subject and order of presentation were blocking factors in the initial two period crossover design where the two systems, a and b, under investigation were tested once by each subject. The design was based on a 2 X 2 Latin square (figure 5.1 below). There is insufficient precision if only two subjects are used so the basic design was replicated for further groups of two subjects. The subjects were randomly allocated to each group so that each group contained equal numbers of subjects and it is simple to randomise the rows.

The problem of the crossover trial is to compare systems whose effects are superimposed on large underlying variations between subjects. To disentangle the system effects from the period effects it is necessary to make the following assumptions about the way the two combine to make up the response in the pilot experiment,

1) The system X period interaction is assumed to be negligible.
2) The contribution of the system and the period to the response of a subject in a specific period would be a fixed amount dependent on the system and is the same, apart for random error, for all subjects.
3) This fixed amount will be the same in both periods.

The results of the pilot experiment would then give an indication as to the validity of these assumptions. Using the simplified assumption of a single test being composed of three tasks, the order of which does not affect the times in each period or system, a suitable experimental layout would be,

		Period	
		1	2
Subject	1	a	b
	2	b	a

Figure 5.1
Treatment Design

where a is the test (logon, read or send) using GOLD by itself,
and b is the test with the intelligent interface.

Therefore the mathematical model of the test situation is;

$$y_{ijm} = \mu + s_i + p_j + b_m + e_{ijm}$$

where

y_{ijm} is the observation on system m in presentation j on subject i.
μ is the overall average of the response variable.
s_i is the effect of the ith subject.
p_j is the effect of the jth period in the sequence.
b_m is the effect of the mth system.
e_{ijm} is the random error.

The random error term is assumed to be uncorrelated and normally distributed with constant variance. Using this model it will be possible to determine whether the intelligent interface is better than the other by using a two tailed significance test.

$$H_0 : \mu_a = \mu_b$$
$$H_1 : \mu_a \neq \mu_b$$

μ_a is the mean of the response variable (i.e. time to complete a task) from interface a, and μ_b is the mean for interface b.

5.3.3 Pilot Experiment

When working in a new area of statistical evaluation it is sound experimental procedure to conduct a pilot experiment for the following reasons. Firstly an estimate of the potential variability in the assessment can be obtained which can be used to identify previously unrecognised factors that could substantially affect the results. These can then be included as blocking factors in the design to improve the sensitivity of the statistical analysis. Thirdly the practical feasibility of the evaluation can be investigated. Finally it can be seen whether the design achieves the required objectives.

The pilot experiment was conducted on four colleagues who were computer users but had limited or no experience of mailbox systems. The method used was similar for all four subjects and the questionnaires were not used. Each experiment was divided into two periods carried out sequentially and the total time did not exceed one hour. There was no training apart from a brief verbal introduction to a mailbox system and its use. For each treatment the subject was provided with a manual (see later) and a task sheet giving details of what was required. Direct observation was used by the researcher to identify any potential sources of confusion and error.

In both trials the interaction of the user with the interface was monitored by the program storing the user's interaction with the system on disc with time codes. The results obtained were then analysed and compared with the estimated times obtained by direct observation. Each time was treated as a separate variable and was analysed separately by a three way analysis of variance (ANOVA) table. This analysis enables any differences associated with the system, subject and presentation factors, which may be greater than those expected entirely as a result of random error, to be detected.

It is important that the monitoring is unobtrusive since otherwise it could have substantially affected the times to do the tasks. A simple experiment showed that the response time of Telecom Gold was increased by on average 0.37 seconds per keypress over all the tasks with a worse case average during the read task using the intelligent interface of 0.93 seconds per keypress. This was deemed insufficient to affect the results because of the variation in GOLD response time. In fact this gave a slight advantage to the times for the existing interface because more data was recorded from the interaction with the intelligent interface, as both the keyboard and the input from Telecom Gold were monitored and recorded.

5.3.4 Stage 3 Redesign of System Stratum

The results of the pilot study are discussed below where they led to a rethink of the assumptions and the design of the full-scale experiment. (For complete analysis see appendix 6.)

The initial design is only valid assuming there is no interaction between the systems and other factors. The results of the pilot study, however, indicated that there was significant evidence of a system X period interaction and so it would not be sensible to compare systems using data from both periods. The comparison of data in the first period was too insensitive to detect a significant difference between the two systems so a modification was required to provide protection against this condition occuring in the experiment.

In the pilot experiment a high mean response is associated with high difference in response to the two systems see figure 5.2. This non-linear variation of the observed system difference with the level of response could account for the interaction.

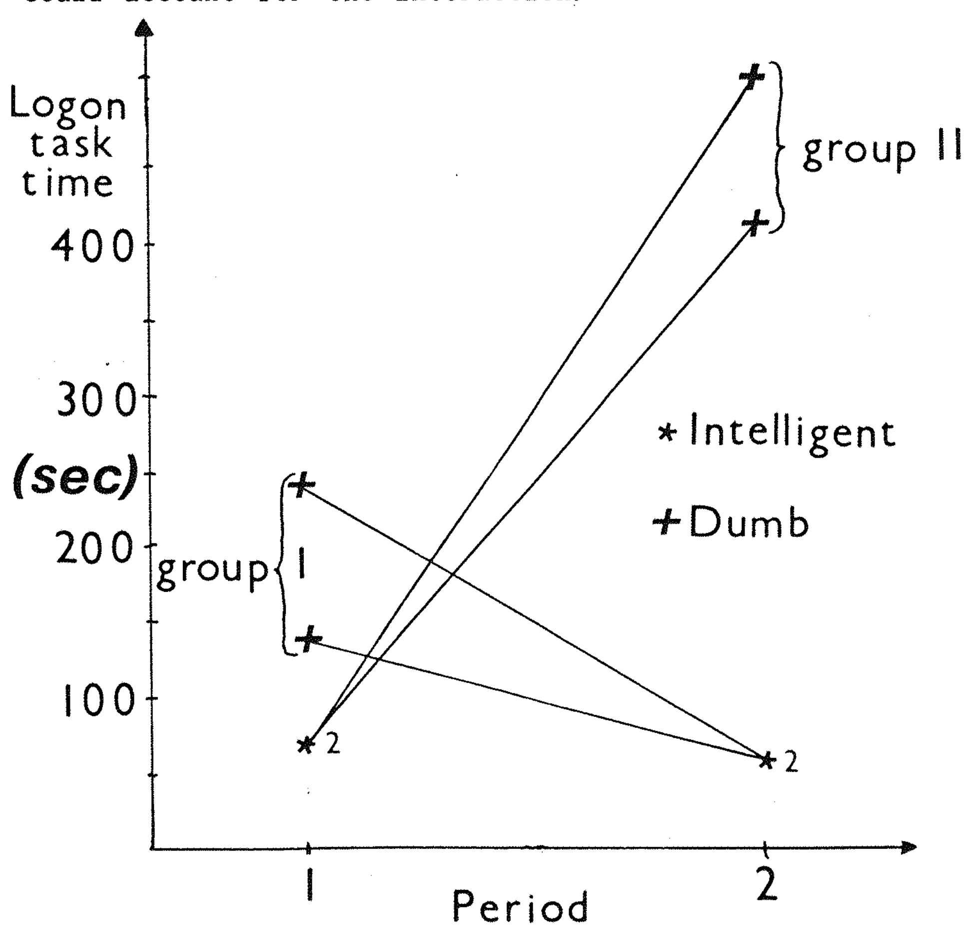


Figure 5.2 Diagrammatic representation of Pilot results.

It might be possible to remove this tendency of the system effect to vary with response level by using a suitable transformation such as the natural logarithm or square root of the raw data. However results of the pilot study showed that it was not possible to remove this interaction so it may be advisable to make one of the following modifications,

1) Design 1 - involving 4 groups of subjects in three periods

Group	Period		
	1	2	3
I	a	b	b
II	b	a	a
III	a	b	a
IV	b	a	b

This gives more information on system versus period interaction. The disadvantages are that it involves a more complicated analysis and increases the length of the experiment. This was undesirable because it was considered prudent to keep the length of the experiment below one hour. It also is not balanced for carryover effects since the sequence ba occurs 3 times but aa occurs only once. However it may be possible to include an adjustment for this effect.

2) Design 2 - involving 4 groups of subjects in 2 periods.

Group	Period	
	1	2
I	a	b
II	b	a
III	a	a
IV	b	b

This design would enable the period effect to be estimated as the system effect should be the same in both periods of group III and IV. The advantage of this method is that groups III and IV need only be run if the results for groups I and II show an interaction between system and period. This design was chosen because it involves the least change to the original design and uses only two periods.

The initial design assumed that the period was an independent quantity which was the same for each subject. From the results it is clear that this assumption may be untrue. Therefore the mathematical model of the test situation becomes;

$$y_{ijm} = \mu + s_i + p_{j(i)} + b_m + (pb)_{jm} + e_{ijm}$$

where

$p_{j(i)}$ = the effect of the jth period within subject i.

$(pb)_{jm}$ = the interaction effect of period j and system m

If the period X system interaction is so high that only the results in period 1 can be used then the intersubject variability could be reduced by grouping subjects into user types. This information can be provided from the pre-test questionnaire which can then be used as a covariate to improve the precision in a period 1 "between subjects" analysis.

5.3.5 The Sensitivity of the Trial

A system difference may be statistically significant but not of practical importance. The power (or sensitivity) of a test is the probability of correctly detecting a difference D, say, between the times for the two systems. For a value of D of PRACTICAL MEANING and importance, we need to ensure that the tests used in the experiment have a sufficiently high power. The power of a test depends on the number of observations in the sample and the variability of the observations. Thus we can control the power of the tests by choice of the population size, provided an estimate of the observation variance can be obtained. Such an estimate can be obtained from the pilot study. If it is assumed that 60 seconds is a meaningfully large true difference and that a trial should have an 80% power then it can be shown that the minimum number of subjects for the simple design is 22. It can also be shown that a similar number is required in the case where there is a period X system interaction. (see appendix 6). The results of the pilot experiment were used to identify other factors which may be included in the experimental design to increase the sensitivity and reduce the population size.

5.3.6 Stage 4 Factorial Approach

In the pilot experiment it was assumed that the three tasks would each give similar comparisons of the systems. This may not be the case as a comparison of one task between systems may be different from a comparison of another task. This is a task X system interaction. Similarly the task order may affect the total time in the period, the task X period interaction. For this reason it is best to adopt a complete factorial approach where the two factors are,

1) systems levels a = Telecom Gold, b = Intelligent interface
   This design is as used in the pilot experiment in section 5.3.2.
2) task levels 1 = logon, 2 = send, 3 = read,
   The design structure of the task stratum is based on 3 X 3 Latin squares which are arranged to produce a balanced design as in figure 5.3 below.

Subject	Period 1			Period 2		
1	1	2	3	1	3	2
2	3	1	2	1	2	3
3	2	3	1	2	1	3
4	3	2	1	3	1	2
5	2	1	3	2	3	1
6	1	3	2	3	2	1

Figure 5.3
Task Stratum design

In the total design every task is preceded by every other task 5 times in a block of six subjects and within each period, every task is preceded by every other twice. Hence the design is balanced for the 1st residual carryover effects when both periods are used and also when only the first period is used. Designs 1 and 2 are combined to give the final 2-factor design as shown in figure 5.4 below. In this design each group must contain multiples of 12 subjects. In order to justify the statistical tests the rows of the design were permuted using a random number table.

	Period 1			Period 2		
	Session			Session		
Subject	1	2	3	1	2	3
1	1a	2a	3a	1b	3b	2b
2	3a	1a	2a	1b	2b	3b
3	2a	3a	1a	2b	1b	3b
4	3a	2a	1a	3b	1b	2b
5	2a	1a	3a	2b	3b	1b
6	1a	3a	2a	3b	2b	1b
7	1b	2b	3b	1a	3a	2a
8	3b	1b	2b	1a	2a	3a
9	2b	3b	1b	2a	1a	3a
10	3b	2b	1b	3a	1a	2a
11	2b	1b	3b	2a	3a	1a
12	1b	3b	2b	3a	2a	1a

Figure 5.4
Complete Factorial
Design

The complete model to this factorial experiment is therefore,

$$y_{ijklm} = \mu + s_i + p_{j(i)} + a_{k(j(i))} + t_l + b_m + (tb)_{lm(j(i))}$$

$$+ (tp)_{lj(i)} + (pb)_{jm} + (tbp)_{lm(j(i))} + e_{ijklm}$$

where μ, s_i, and $p_{j(i)}$ are as before and,

$a_{k(j(i))}$ is the effect of session k within period j and subject i,

t_l is the effect of task l (= logon, read or send),.

b_m is the effect of system m (= GOLD or NEW),

$(tb)_{lm(j(i))}$ is the effect of task l and system m, within period j and subject i,

$(tp)_{lj(i)}$ is the effect of task l and period j, within subject i,

$(pb)_{jm}$ is the effect of period j and system m,

$(tbp)_{lm(j(i)))}$ is the effect of task l and system m and period j within subject i,

e_{ijklm} is the random error.

5.3.7 Method of Analysis

An analysis of variance can be used to investigate the differences in the mean times taken to perform all tasks together using each system. It also enables a comparison between systems of the mean times taken to perform the individual tasks. The analysis of variance consists of three strata : between subjects, between periods within subjects and between sessions. Each effect belongs to a stratum and must be tested using the residual of that stratum.

Stratum 1 - between subjects

Firstly the total times for systems a and b are compared for the two different periods. This comparison is between subjects, and would indicate a period X system interaction. In this case the results from both periods cannot be used unless the interaction effect can be removed by a suitable transformation of the data such as taking the logarithm or square root. If the interaction effect still remains after transforming the data then only a between subjects comparison can be made using data from the first period only. The lower strata of the analysis would be conducted on this first period data.

Stratum 2 - between periods within subjects

If there is no evidence of a period X system interaction, or the interaction can be removed by transforming the data, then both periods of data can be used. The between periods within subjects analysis will indicate whether there is a significant difference between systems and between periods for the overall set of three tasks taken together.

Stratum 3 - between sessions

The tasks can then be compared within each session and within each system. If there is a task X system interaction between sessions this will indicate that the differences in times to complete the tasks for the two systems differ.

5.4 Additional Design Points

5.4.1 Questionnaire and Manual Design

A questionnaire was used to complement the objective performance data outlined above by the subjective measurement of user satisfaction and ease of use. The questionnaire consisted of two parts, pre- and post-test. The pre-test questionnaire was used to screen subjects to ensure that homogeneous groups were chosen; the post test questionnaire was used to corroborate the quantitative data.

The design of questionnaires which would obtain the relevant information was not a simple task and so help was sought from a psychologist in constructing a suitable design (see acknowledgements). The major requirements for a good questionnaire are given below as they relate to the design under study (Bailey 1982, Sinclair 1975, Brigham 1975) and the questionnaires are given in full in Appendix 7.

The questionnaire needs to be applied to, and written for, a representative sample of the population for which the system has been designed. Therefore the assessment was conducted on potential naive users and existing experienced users of a mailbox system. The pre-test questionnaire must contain questions which make it easy to eliminate unrepresentative candidates (see questions 18-28).

The questions were made unambiguous and where any confusion might arise the question was repeated in a different form. For example, hard of hearing people may consider themselves deaf but questions nos. 4 - 9 should enable this mistake to be detected. Open and closed questions were used so that subjects could qualify their answers.

The pre-test questionnaire consisted of three main sections,

1) degree of deafness, nos. 4 - 9
2) level of intellectual and textual ability, nos 10 - 17
3) level of technological experience. nos 18 - 28

A few questions on cost were included as a rough guide to the target price of future systems.

In the post-test questionnaire use was made of the filter question and branching instructions so some subjects could miss out questions irrelevant to them. Also the layout of the instructions was made distinct from the questions. The main sections were,

1) rating questions on ease of use and satisfaction, nos 4- 11,
   (A rating scale of 1 - 5 was deemed sufficient as this was considered the finest resolution people could distinguish between). Each rating level was ordered in increasing level of response for each set of questions and the alternatives were mutually exclusive. A rating test was used instead of a

comparison test because a measure of absolute satisfaction was required. One system may be better than another but may still be considered to be unsatisfactory by the user.

2) open questions about whether subjects were satisfied, frustrated or confused with the interface and if so why. Also they had a chance to write down any comments, positive or negative, about the interfaces (12 - 25)

3) Questions on the manual (26 - 30)

Each questionnaire took about 10 - 15 minutes to complete, depending on the level of detail given by the subjects in the open questions. All subjects found the questionnaire easy to follow and the answers showed that they understood the questions.

Manuals were required for the naive users because they were not familiar with mailbox systems however it was not expected that the subjects would use the intelligent interface manual very much. No manuals were provided for the expert users because these subjects would be familiar with Telecom Gold.

Only a limited set of mailbox commands were to be used so the GOLD manual, in its entirety, was deemed unsuitable for use in the evaluation. Both manuals also need to have a similar format to avoid the possibility of introducing manual design as another variable in the model. Space precludes a detailed description of the manual structure but the reader is referred again to Bailey (1982). The important points included in the manual were a good index, clear demarkation of sections and relevant figures. The basic structure of the GOLD manual was used but improved upon and the intelligent interface manual was made similar in content. The language intended for the business user was reworded and all superfluous information was deleted. Although the resulting manuals were not completely satisfactory they were considered to be a vast improvement on the original. Further experimentation on the design and construction of the manual is necessary but is beyond the scope of this project. The resulting manuals are shown in appendix 8.

5.4.2 Questionnaire Analysis

When analysing the quantitative time parameters it could be assumed that the data is obtained as random samples from normal distributions. Ratings were used for the questionnaire answers so the scale of measurement is non-linear and the normality assumption may not be true in this case. It may be better to use a non-parametric test which does not require the normality assumption. The data obtained from a questionnaire was in the following sections,

a) How easy did you find it to do the following tasks logon, compose an envelope, send a letter and read a letter? (scale 1 for very difficult to 5 for very easy).

b) How satisfying did you find it to do the following tasks logon, compose an envelope, send a letter and read a letter? (scale 1 for not satisfying to 5 for very satisfying).

Grizzle (1965) suggested a suitable model for data from crossover trials which is given below,

$$Y_{ijk} = \mu + S_{ij} + P_k + \beta_i + T_l + E_{ijk}$$

$$j = 1,\ldots n_i,\ i = 1,2\ ,\quad k = 1,2\ ,\ l = 1,2$$

where,

μ = general mean;

S_{ij} = the effect of the jth subject within the ith group;

P_k = the effect of the kth period;

β_i = the direct effect of the ith system;

T_i = the residual effect of the ith system in the first period on the response in the second period. (i = 0 for 1st period measurements).

e_{ijk} = the random fluctuation.

where s_{ij} and e_{ijk} are each normally distributed and mutually independent. For the analysis the non-parametric modified Wilcoxon test (also called the Mann-Whitney test) based on ranks can be applied to 'within subject' functions of the data (sums, differences and crossover differences) (Gibbons 1971). These tests are suitable for the data presented here because of the different patterns of variability at different response levels.

To test the equality of the residual effects a Wilcoxon rank test is performed on the sums of both period data. If the residual effect of the interface b in the first period on the response in the second period is higher than the effect of interface b in the second period then the sums in one half of the test will be higher and so will give a low rank sum in one group. If the residual effects are significant then the data from both periods cannot be used and the Wilcoxon statistic must be applied to the data observed in the first period only. Otherwise if residual effects can be deleted from the model, the test for equality of direct effects can be found by applying the rank test to the differences between the periods.

Finally to check for equality of period effects the crossover differences (i.e. a-b or b-a) are tested. If this is not significant this shows that there are no substantial period differences.

5.5 Experimental Method

Using the modifications obtained from the pilot experiment the following procedure was adopted. The evaluation was conducted on the two subject groups during late 1985 and early 1986. The method used was similar for both groups and for each subject both systems were given sequentially on the same day.

The trial was conducted with the actual Telecom Gold interface rather than a simulated interface. This was to make the experimental situation as near to the real world as possible and because it would be too time consuming to develop a simulated interface. Also allowing the subjects to use the intelligent interface with the existing interface any interface design problems could be discovered. This however may cause problems if the existing interface performs differently, for example in response time, for each user. This problem can be alleviated by asking the subjects to complete the trial at times of the day where the response time is consistent.

For each naive subject the evaluation began with a brief verbal introduction to a mailbox system and an explanation of its use. Any candidates unfamiliar with computers were also given a brief description of the layout and purpose of the keyboard and a brief introduction to the use of menus. No practical demonstration was given to the subject as this could influence the use of the system.

Before beginning the assessment the subjects completed the pre-test questionnaire. From this it was possible to define an 'inclusion criterion which subjects must satisfy to be included in the experiment. This enabled the extreme levels of comfortability, when using a keyboard or computer, of some subjects to be eliminated and allowed the selection of a homogeneous group of people for testing. This improves the precision of the statistical analysis but restricts the conclusions to the population of naive users who are not extremely uncomfortable. A cursory analysis of the pre-test questionnaire showed a correlation between how 'comfortable' a subject felt with computers and the length of time he took to complete the tasks. On the basis of the pre-test two naive subjects were rejected and two new hearing subjects replaced them. In fact, even with the two rejected subjects the intelligent terminal performed better than the dumb terminal but a further evaluation with this subject group is required before any conclusions can be drawn.

Each subject was then given a task sheet (see appendix 9) and manuals with the tasks in a certain order as defined by the structure in section 5.3.6. The subject then carried out the tasks under the observation of the researcher. The observer gave no help to the subjects except for reassurance and occasional queries as to the nature of the task. Videotaping of the experiment had been considered but this was rejected because if the subject had known about it then it might have influenced the results and it would be unethical to videotape without the consent of the subject.

After the first period the subject was given the second task sheet and the manual. At the end of the test the subject was asked to complete the post-test questionnaire. The complete test did not take longer than 60 minutes for the acceptable candidates. None of the naive users were told which system was the intelligent interface.

5.6 Conclusions

The procedure to compare two user interfaces to a mailbox system is not straightforward. Particular care needed to be taken in detecting and removing nuisance factors and potential interaction effects. A simple model which can be used to describe the time (y) to use the system is,

$$y = \mu + (e_{total\_residual})$$

where μ is the mean time and $e_{total\_residual}$ is the total residual variation.

It can be seen that there are several possible sources of variation contributing to the observations and some of these are,

$e_{subject}$ = error due to variation between subjects

e_{period} = error due to variation between periods

$e_{session}$ = error due to variation between sessions

$e_{total\_residual}$ can be said to be composed of these three sources of variation mentioned plus a residual variation $e_{residual}$.

A pilot experiment was conducted using a simple experimental model in order to identify any sources of variation which could be removed from the total residual variation. A model was thus developed which would take into account all the problems exposed by the pilot experiment and so allow a more precise comparison of the mean time differences between the two systems. A complete factorial design was developed which would yield information on the potential experimental problems, enable the nuisance effects to be estimated and provide information as to the practical importance of these factors. Although this model complicated the experimental procedure it fulfilled the requirement to obtain statistically valid results with a small population size.

CHAPTER 6

EXPERIMENTAL RESULTS AND DISCUSSION

6.1 Alternative Hypotheses and Analysis of Variance

A null hypothesis H_0 was tested against an alternative hypothesis H_1. The hypotheses were,

i) There is no difference between the mean times, μ_1 and μ_2, to use the two interfaces, a and b

$$H_0 : \mu_1 = \mu_2$$

ii) There is a difference between mean times to use the interfaces.

$$H_1 : \mu_1 \neq \mu_2$$

In addition to this the qualitative parameters of ease of use and user satisfaction were used to corroborate the quantitative time parameters. In both cases the test was based on an analysis of variance. The pilot experiment analysis had been performed on a calculator. However for the factorial experiment this would have been too time consuming so a general purpose statistical analysis package (GENSTAT) was used to perform the analysis on an ICL 2970 computer. The balanced structure of the design, developed in chapter 5, enabled the ANOVA directive of GENSTAT to be used.

6.2 Naive Subjects

This subject group consisted of 12 naive hearing users who did not register, in the pre-test questionnaire, that they felt uneasy or uncomfortable about using a computer and keyboard. The raw times for the 12 subjects are shown in table 6.1 and the results for the analysis of variance are shown in table 6.2. A diagrammatic representation of the results is shown in figure 6.1.

6.2.1 Analysis between Subjects

a) Interaction between period and system

There was no strong evidence of an interaction between these two factors which means that the difference between systems is the same for both periods. The times in both periods can therefore be used for the remainder of the analysis.

Tasks	1		2		3	
interface	a	b	a	b	a	b
Subject						
1	192	73	322	105	267	305
2	162	69	271	198	357	165*
3	151	70	371	119	265	107
4	225	67	271	172	468	149
5	109	78	187	145	315	108
6	105	69	129	105	366	204
7	175	68	153	137	468	123
8	171	72	200	119	231	101
9	162	71	328	217	360	188
10	88	69	86	120	174	277*
11	105	70	273	195	129	135
12	157	68	207	170	168	150

* time amended because a software fault occurred during test.

Table 6.1 Raw task times for naive users

Strata	Source of variation	d.f.	s.s.	m.s.	F-ratio (significance)
Between Subjects	Period X System	1	10177	10177	1.91 (NS)
	Residual	10	53039	5304	
	Total	11	63216		
Between periods within subjects	Between Systems	1	171112	171112	32.25 (0.1%)
	between periods	1	12272	12272	2.31 (NS)
	Residual	10	53052	5305	1.27
	Total	12	236436		
Between sessions	Between tasks	2	185929	92964	22.34 (0.1%)
	task X system	2	9342	4671	1.12 (NS)
	period X task	2	3475	1738	0.41 (NS)
	period X task X system	2	5328	2664	0.64 (NS)
	residual	40	166443	4161	
	Total	48	370518		
	Grand Total	71	670170		

Table 6.2 Analysis of Variance

d.f. - degrees of freedom.

s.s. - sum of the squares.

m.s. - mean square.

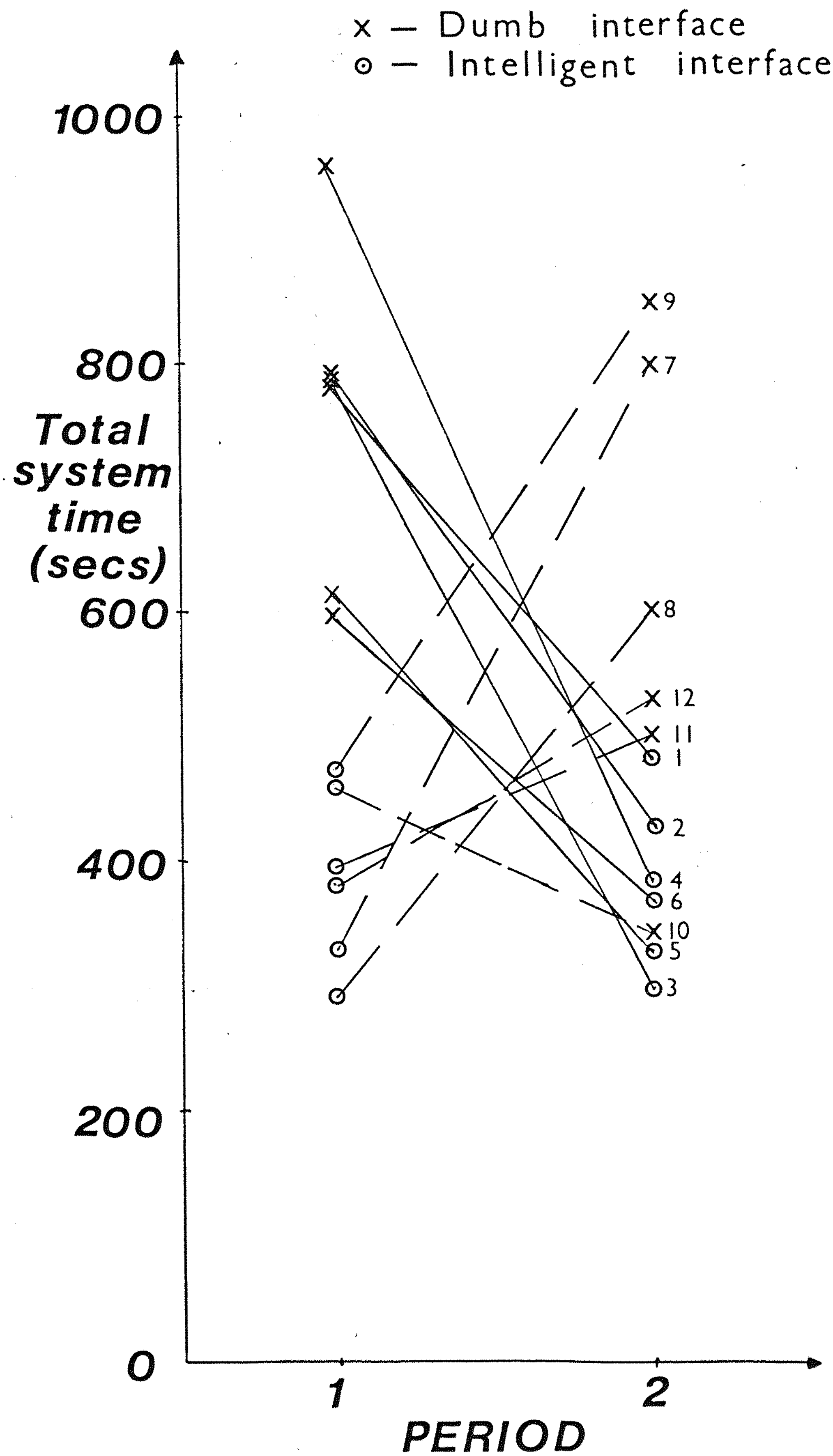


Figure 6.1 Diagrammatic Representation of Naive Users Results

6.2.2 Analysis between Periods within Subjects

a) Differences between systems

There is very strong evidence, significant at the 0.1% level, that there is a difference in the times taken to use the two systems. The total times for all three tasks taken together for the intelligent interface are shorter than for the dumb terminal. The average times taken to complete each task under each system, where the average is taken across period, session and subject, are shown in table 6.3.

	SYSTEM	GOLD=1	NEW=2	AV. FOR TASK
	'LOGON'=1	150.2	70.3	110.3
TASK	'SEND'=2	233.5	150.2	191.7
	'READ'=3	297.3	167.7	232.5
	AV. FOR SYSTEM	226.9	129.4	

Table 6.3 Table of average times to complete each task

The ANOVA indicates that there is a significant difference between the systems. To detect where the differences occur the two systems can be compared for each task as in the table below,

Task	1	2	3	1% LSD
Gold - New	79.9	83.0	129.6	74.4

Table 6.4 Comparison of the two systems for each task

If the difference in the means for each task is greater than the least difference between the two means (significant at, say, the 1% level) then the conclusion is that the means are different. This is effectively a t-test on the two means for each task. From table 6.4 we can conclude that for each task the new interface was faster to use than GOLD, significant at 1% level, using the t-test (see appendix 6 for more explanation of LSD).

b) Differences between periods

There is no evidence that the total times for period 1 differed significantly from the total times for period 2.

6.2.3 Analysis between Sessions

a) Differences between tasks

As might be expected there is strong evidence, significant at the 0.1% level, of a difference between the tasks (there was no attempt to equalise the length of the tasks).

b) Interaction between task and system

The task X system interaction was not significant in the analysis of variance, indicating a similar pattern of differences amongst the times to complete the three tasks for each of the systems. (see following figure 6.2).

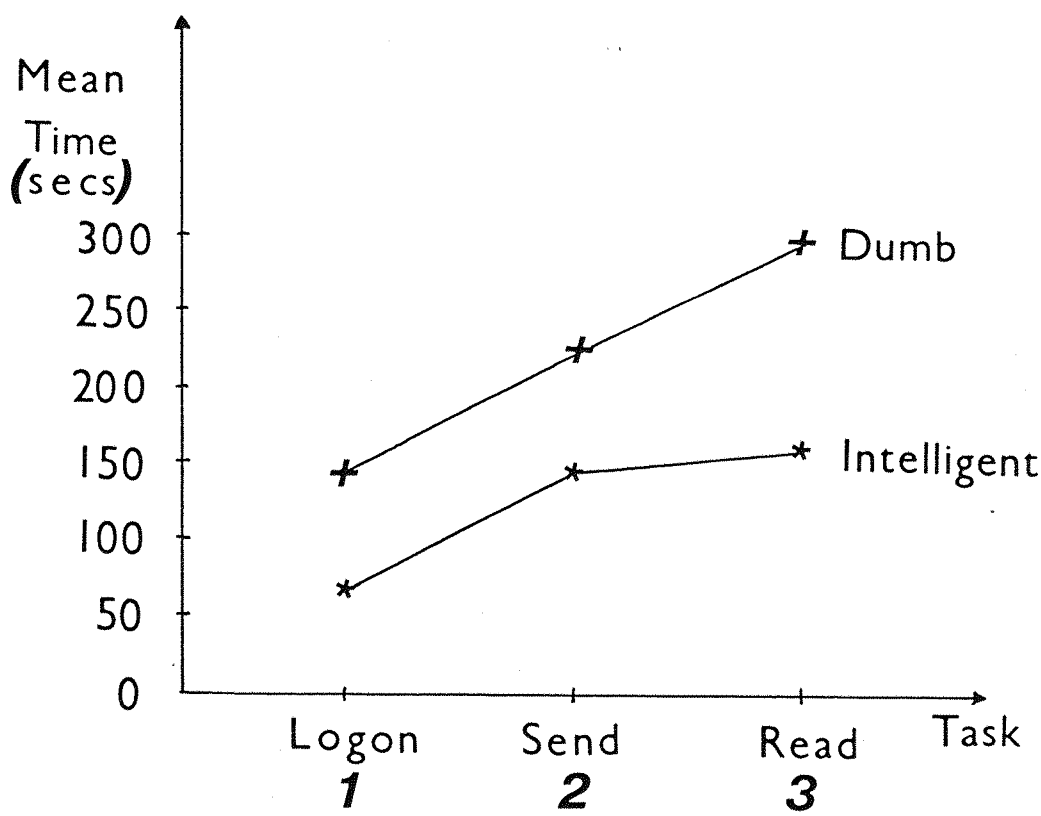


Figure 6.2 The pattern of mean task times for the naive subjects

c) Interaction between period and task

There is no evidence from these data that the period affects the time differences between the three tasks.

d) Interaction between period and task and system

There is no evidence of this type of interaction.

6.2.4 Check on Assumptions

The graph of residuals against fitted values of system times generated by GENSTAT is shown in figure 6.3. There is a noticeable funnelling out as the times increase which indicates a non-constant variance. If this is severe, then this could affect the validity of the t and F tests which are based on the assumption of constant variance for the observations. This funnelling out could be due to an unequal time correlation where the longer the subjects take in the trial the more frustrated they become and so the longer the total time to complete the trial. In an attempt to overcome this the data was transformed using the natural logarithm ($\log_e$) of the raw times. Figure 6.4 shows the resultant graph of residuals against fitted values of $\log_e t$ from which it can be seen that the non-constancy of variance has been removed. However this did not change the conclusions obtained with the raw data. For clarity only the raw results have been included here.

6.2.5 Practical Importance of Results

There is no evidence that doing the tasks in any period or in any order significantly affected the times to complete the tasks. The analysis shows that the intelligent interface makes a mailbox system like Telecom Gold, quicker to use than the existing interface. This improved speed is of practical importance, decreasing the time to use the system as a whole, and each individual task, by a factor of approximately two. Therefore we can conclude for this group of people that the alternative hypothesis is correct, i.e. the mean times to use the two systems differ.

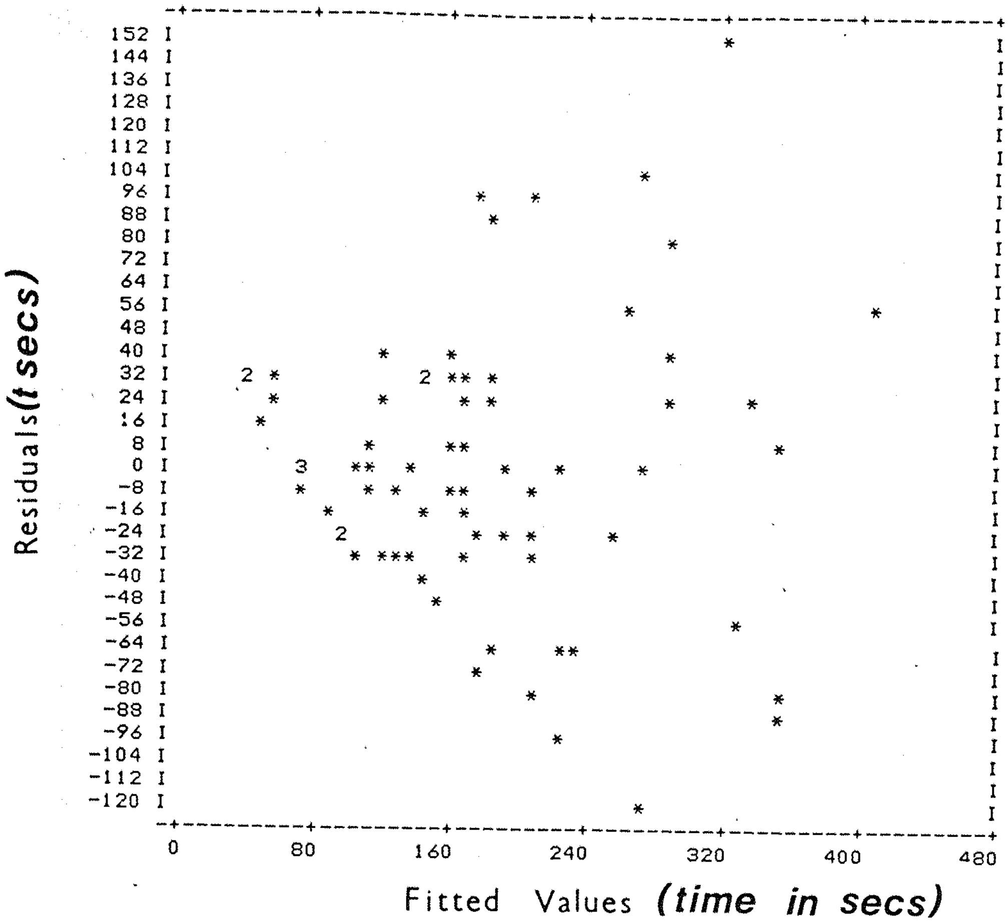


Figure 6.3 Plot of residuals against fitted values (raw data)

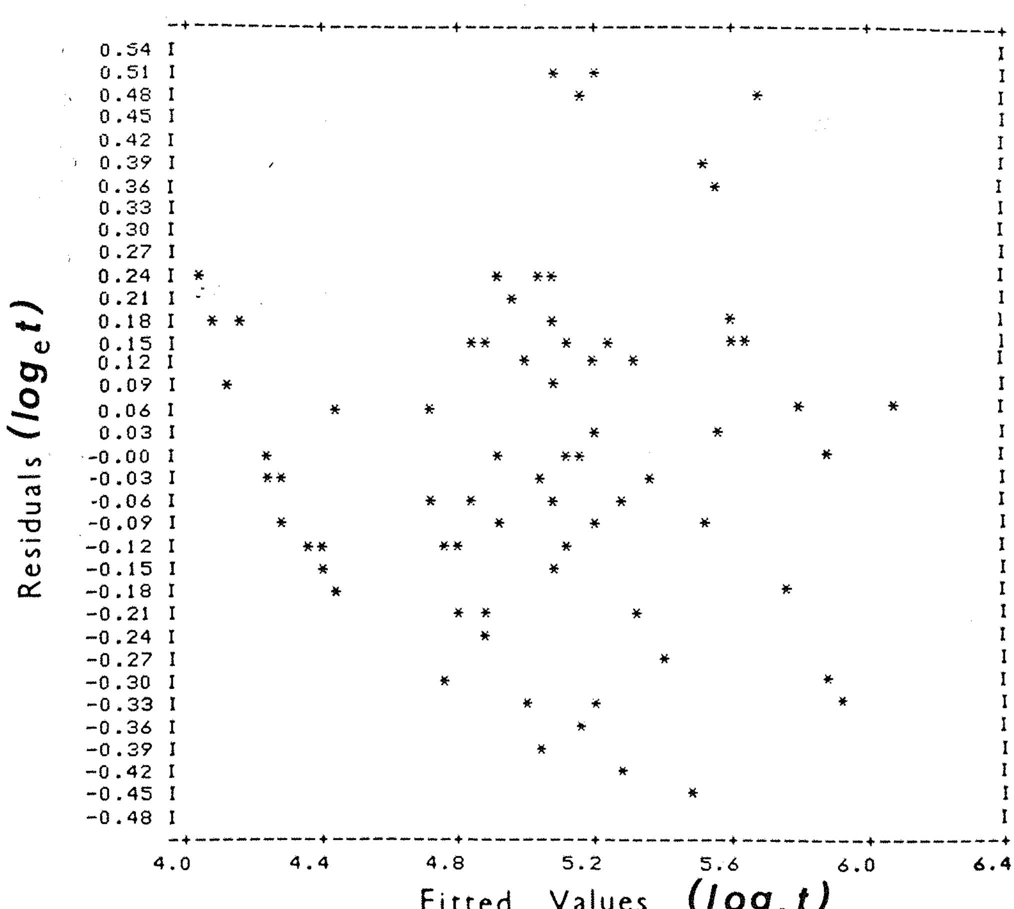


Figure 6.4 Plot of residuals against fitted values ($\log_e$ data)

6.2.6 Non-parametric Analysis

Tables 6.5 and 6.6 give the raw scores that each subject gave for the ease of use and user satisfaction of each interface. The actual calculations will not be discussed in detail and the full analysis is given in appendix 6.

Application of the Mann-Whitney test to the results of the total rating for all the tasks showed that there was no evidence of a period X system interaction so results from both periods could be used. It also showed that overall the intelligent interface was found subjectively easier to use than the dumb terminal at the 1% significance level. If a similar test is conducted on the rating scores for each task individually there is evidence that the intelligent interface is preferred more than the dumb terminal (in terms of ease of use) for 'logon', 'compose and envelope' and 'read' (1%), and 'send' (2%).

Analysis of the 'user satisfaction' results showed that there was period X system interaction. Analysis of the ratings from the first period alone show evidence that the intelligent interface is preferred to the dumb terminal for 'read' (5%), 'compose an envelope' (1%) and send (10%) but there is no evidence that 'logon' is more satisfying in either interface.

6.2.7 Detailed Analysis of Individual Results

The pre- and post-test questionnaire analysis produced the following information. All the subjects were 20-29 year old undergraduates or post-graduates of whom only 1 was female. All were comfortable using computers before the test and none felt uncomfortable during the test. 9 out of 12 thought they might find mailbox systems useful in the future.

In the post-test open questions the following comments were made about interface a (the existing interface). All 12 people found the interface either confusing or frustrating in some aspects. The main factors mentioned were, Question: How easy did you find it to do the following task (on a rating scale or 1 for very difficult to 5 for very easy).

	'logon'		'comp'		'send'		'read'		Total	
Subject	a	b	a	b	a	b	a	b	a	b
1	4	5	4	5	4	5	5	5	17	20
2	2	5	2	5	3	5	1	5	8	20
3	3	4	2	5	2	5	2	4	9	18
4	3	5	3	4	3	4	2	5	11	18
5	3	5	3	4	3	3	2	4	11	16
6	3	5	3	4	4	4	4	4	14	17
7	4	5	5	5	4	5	2	5	15	20
8	3	5	4	3	5	4	4	4	16	16
9	3	4	2	3	2	3	2	4	9	14
10	4	5	3	5	4	5	3	3	14	18
11	4	3	4	4	3	5	2	4	13	16
12	2	5	3	4	3	4	3	4	11	17

Table 6.5 Raw 'ease of use' scores for naive users

Question: How satisfying did you find it to do the following tasks, on a scale or 1 for dissatisfied to 5 for very satisfied.

	'logon'		'comp'		'send'		'read'		Total	
Subject	a	b	a	b	a	b	a	b	a	b
1	4	3	3	4	3	4	3	3	13	14
2	2	4	1	5	3	4	2	4	8	17
3	2	3	1	4	2	4	1	3	6	14
4	2	4	2	4	2	4	1	4	7	16
5	2	3	2	2	3	3	3	3	10	11
6	3	5	3	5	3	3	3	3	12	16
7	4	3	3	3	5	3	2	3	14	12
8	3	3	4	4	3	3	4	4	14	14
9	5	3	3	3	3	3	2	3	13	12
10	3	2	3	4	4	4	3	3	13	13
11	4	3	3	4	4	4	3	4	14	15
12	2	3	3	4	3	4	4	4	12	15

Table 6.6 Raw 'user satisfaction' scores for naive users

a) the unfriendly appearance of options and the lack of on-screen information leading to confusion as to the expected response. Subjects had particular difficulty in discovering how to read a specific letter.

b) the unpredictable command syntax and confusing commands such as,
i) the critical use of full stop (e.g. .send)
ii) a qualifier was expected by the subject after specifying read <RETURN> but the system did not provide one.
iii) different exit methods from similar levels, (.q, .s, quit)
iv) confusion as to the difference between SCAN and READ. During the experiment the experimenter observed that many subjects SCANNED mail with the intention of READING the letter.

c) the slow response of interface. This was due mainly to the slow response of Telecom Gold. The disc access time did not adversely affect the total task times.

Suggested improvements were to make the existing interface a more like the intelligent interface b, to increase the number of characters per line and lines per page and to have more onscreen information.

Only 6 out of 12 people found some aspects of interface b confusing or frustrating. The main factors were,

a) the speed of response of the interface. The intelligent software increased the time to execute the commands. This would not be the case in the final version because the software would reside in memory as firmware ensuring faster execution.

b) confusion as to whether to press TAB or RETURN to move to another field in the 'compose a letter' envelope procedure.

c) one person thought that there was too much information to understand within the menu window and also that the mailbox text window changed too quickly, whereas one subject wanted them to change quicker!

Suggested improvements were to include the facility to specify a letter by name of person or subject rather than just by the number of the letter obtained from the SCAN command, more immediate feedback and the facility to enter SCAN from READ.

The manuals were used by all subjects for interface a and only one candidate found it unhelpful. The manuals were used by 7 out of the 12 candidates for interface b and only 2 found them unhelpful so the manuals did not seem to contribute a negative factor to the test.

6.3 Experienced Subjects

This subject group consisted of 12 hearing impaired users who were mailbox users. The raw times for the 12 subjects are shown in table 6.7 and the results for the analysis of variance are shown in table 6.8. A diagrammatic representation of the raw results is shown in figure 6.5.

6.3.1 Analysis Between Subjects

a) There is no evidence of a period X system interaction.

6.3.2 Analysis between Periods within Subjects

a) Differences between systems

There is no evidence that there is a different between systems. The average times taken to complete each task under each system, where the average is taken across period, session and subject, are shown in table 6.9.

From table 6.10 below we can conclude that the dumb terminal is preferred to the intelligent interface for 'read', the reverse is true for 'logon', and there is no evidence of a difference in mean time to use for 'send'.

Tasks	1		2		3	
interface	a	b	a	b	a	b
Subject						
1	81	65	60	90	51	362
2	137	62	178	151	120	143
3	88	66	75	110	42	76 (*)
4	125	59	333	171	336	488
5	297	66	157	143	289	243
6	110	70	121	157	235	164
7	76	64	34	114	54(**)	166
8	91	66	37	106	31	151 (+)
9	141	60	43	75	63	352
10	90	62	78	118	102	351
11	133	67	139	94	88	342
12	131 (++)	59	78	91	105	393

* time amended due to subject using previous knowledge to find letter

** time amended due to software fault.

\+ time adjusted because subject had misunderstood test.

Table 6.7 Raw times for experienced users

Strata	Source of variation	d.f.	s.s.	m.s.	F-ratio (significance-%)
Between Subjects	Period X System	1	22614	22614	1.59 (NS)
	Residual	10	142036	14204	
	Total	11	164649		
Between periods within subjects	Between Systems	1	15842	15842	4.37 (10%)
	between periods	1	25916	25916	7.15 (5%)
	Residual	10	36253	3625	1.27
	Total	12	78011		
Between sessions	Between tasks	2	144008	72004	16.27 (0.1%)
	task X system	2	129473	64736	14.62 (0.1%)
	period X task	2	13189	6594	1.49 (NS)
	period X task X system	2	6716	3358	0.76 (NS)
	residual	40	177077	4427	
	Total	48	470462		
Grand Total		71			

Table 6.8 Analysis of Variance

	SYSTEM	GOLD=1	NEW=2	AV. FOR TASK
	'LOGON'=1	125.0	63.8	94.4
TASK	'SEND'=2	111.1	118.3	114.7
	'READ'=3	126.3	269.2	197.8
AV. FOR SYSTEM		120.8	150.5	

Table 6.9 Table of average times to complete each task

Task	1	2	3	1% LSD	5% LSD
Gold - New	61.2	-7.2	-142.9	71.2	53.2

Table 6.10 Comparison of both systems for each task

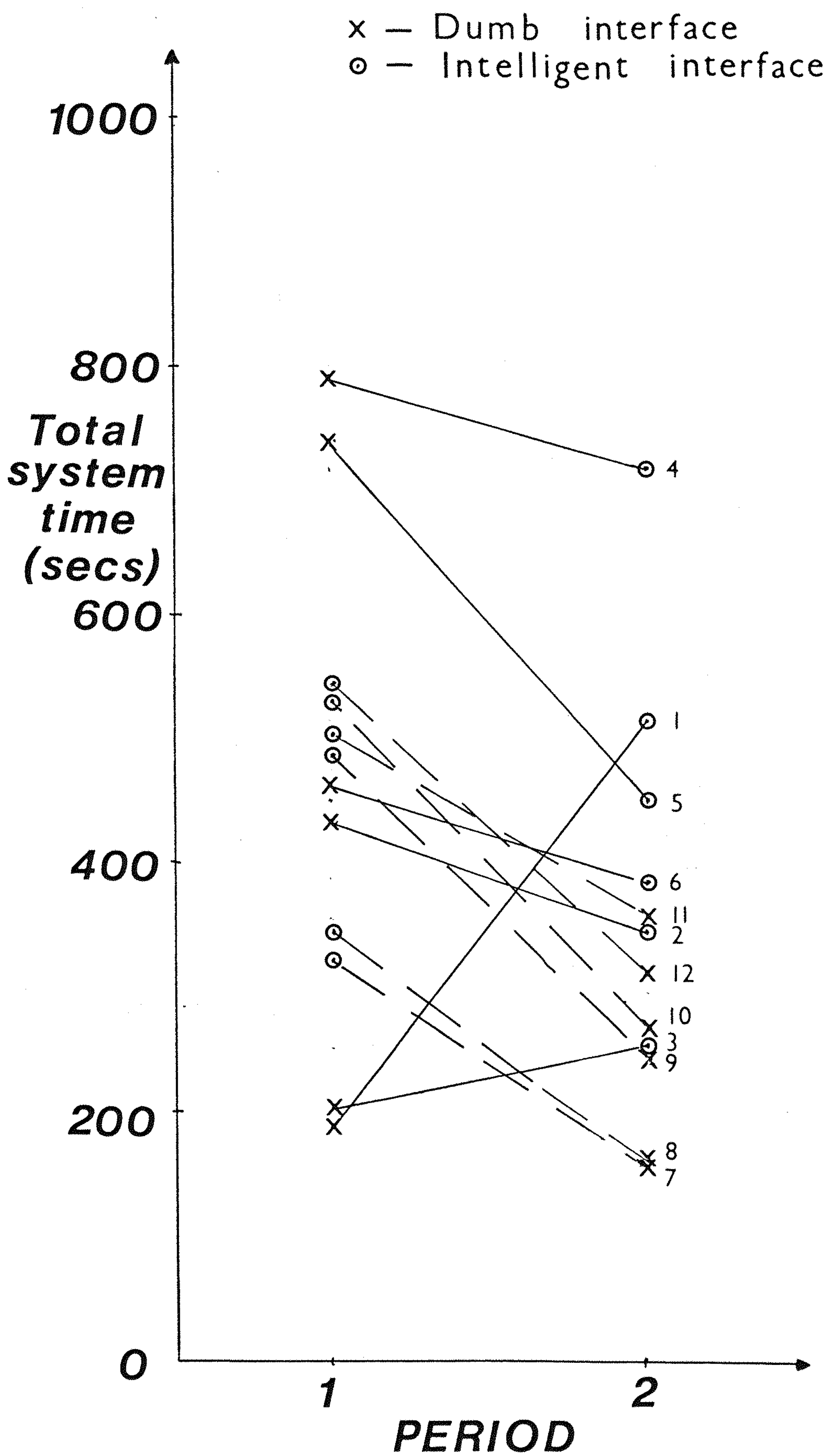


Figure 6.5 Diagrammatic Representation of Experienced Users Results

b) Differences between periods

There is some evidence, significant at the 5% level, of a period effect. A possible explanation is that when the subjects experienced the first period with system a there was only one unfamiliar factor, the assessment itself, whereas when system b was used then there were two unfamiliar factors in the first period.

6.3.3 Analysis between Sessions

a) Differences between tasks

Again there is strong evidence, significant at the 0.1% level, of a difference between the tasks.

b) Interaction between task and system

There is very strong evidence of a task X system interaction, significant at 0.1% indicating that there was not a similar pattern of differences amongst the times to complete the three tasks for each of the systems, see figure 6.6. Therefore it is misleading to average the system time over all three tasks in this case.

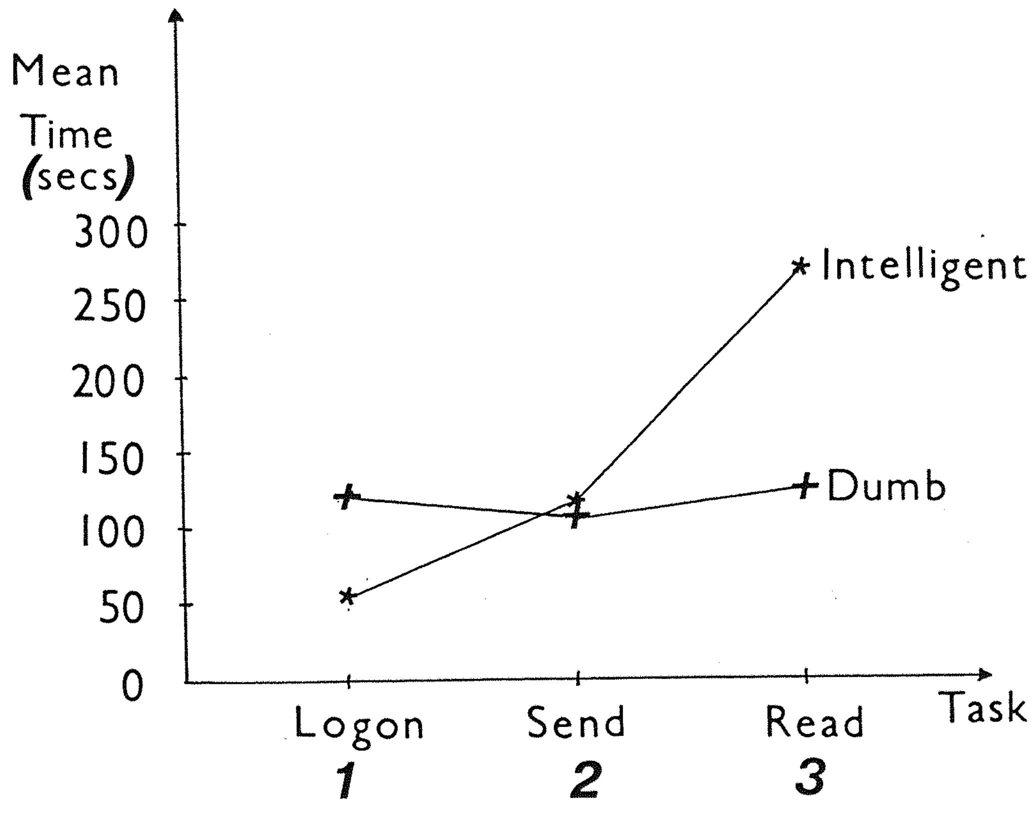


Figure 6.6 The pattern of mean task times for experienced users

c) There is no evidence of interaction between period and task.

d) There is no evidence of an interaction between period and task and system.

6.3.4 Check on Assumptions

The residual plots again show a non-constancy of variance and the results were processed as before. There were no discrepancies between the conclusions of the raw results and the $\log_e$ results.

6.3.5 Practical Importance of Results

There is no evidence that doing the tasks in any period affected the times to complete the tasks. The analysis shows that there is very strong evidence of a task X system interaction so it is not sensible to average over all the tasks. The only task in which interface b (the intelligent interface) slows the user is in the 'read' task.

6.3.6 Non-parametric Analysis

Tables 6.11 and 6.12 give the raw scores that each subject gave for the ease of use and user satisfaction of each interface. The actual calculations will not be discussed in detail (see appendix 6).

Application of the Mann-Whitney test to the results of the total rating for all the tasks for the ease of use and the user satisfaction scores showed that there was no evidence of a period X system interaction so results from both periods could be used. It also showed that overall there was no evidence to suggest that the intelligent interface was found subjectively easier or more difficult to use than the dumb terminal. If a similar test is conducted on the rating scores for each task individually it is found that the dumb terminal was preferred to the intelligent interface for only the 'read' task (ease of use 5% and user satisfaction 10%).

Question: How easy did you find it to do the following task, (scale 1 for very difficult to 5 for very easy).

	'logon'		'comp'		'send'		'read'		Total	
Subject	a	b	a	b	a	b	a	b	a	b
1	5	5	5	2	5	2	5	1	20	10
2	5	5	5	5	5	5	5	5	20	20
3	4	5	5	5	5	5	5	4	19	19
4	3	3	3	3	4	3	3	3	13	12
5	3	4	2	3	2	3	3	3	10	13
6	5	4	5	4	5	4	5	4	20	16
7	4	5	4	3	4	3	4	3	16	14
8	3	5	5	2	5	3	5	2	18	12
9	5	5	5	5	5	5	5	5	20	20
10	4	5	3	5	4	3	4	3	15	16
11	2	2	4	4	4	2	2	2	12	10
12	5	3	2	3	5	5	5	2	17	13

Table 6.11 Raw 'ease of use' scores for experienced users

Question: How satisfying did you find it to do the following tasks, on a scale or 1 for dissatisfied to 5 for very satisfied.

	'logon'		'comp'		'send'		'read'		Total	
Subject	a	b	a	b	a	b	a	b	a	b
1	4	3	4	3	4	3	4	3	16	12
2	3	3	3	3	3	3	3	3	12	12
3	4	4	5	5	5	5	4	5	18	19
4	4	3	4	3	4	3	4	3	16	12
5	3	2	3	2	3	2	3	2	12	8
6	4	3	3	4	4	5	4	5	15	17
7	3	4	4	2	4	2	4	2	15	10
8	2	5	4	2	4	3	5	2	15	12
9	5	3	5	3	5	3	5	3	20	12
10	3	4	3	4	3	5	4	4	13	17
11	2	2	3	4	2	4	4	2	11	12
12	4	5	3	4	3	4	4	5	14	18

Table 6.12 Raw 'user satisfaction' scores for experienced users

6.3.7 Detailed Analysis of Individual Results

The pre- and post-test questionnaire analysis found the following information. The subjects varied in age from mid-twenties to old age pensioners of which only 3 were male. All except one were comfortable using computers before the test and none felt uncomfortable during the test. Also none of the subjects thought they would find mailbox systems of little use in the future.

In the post-test open questions the following comments were made about interface a. As expected 11 of the 12 people were not frustrated by interface a. 7 out of the 12 said some aspect of interface a confused them but this was mainly due to slow response of the system and the unexpected format (e.g. unfamiliarity of BBC computer). One person was confused when a software bug meant she had to start again and one person misread the task sheet and found the wrong letter. None of these problems were serious and most people were comfortable using the interface familiar to them. None of the factors mentioned by the naive users were noticed by the experienced users.

All 12 people found some aspects of interface b confusing or frustrating. The main factors were,

a) speed of response of interface.

b) two people were confused as to whether to press RETURN to move to another field in the 'compose a letter' envelope procedure. Observation showed this to be true of some of the other subjects.

c) 10 out of the 12 people noted unfamiliar layout and menus. Most of the people thought that this would be limited to the first occasion and that it would be easy to use the next time.

d) Two people found that there was too much information to understand in one of the read menu windows and that some help information was unclear.

Suggested improvements were,

1) the inclusion of a facility to specify a letter by name of person or subject rather than just by the number of the letter obtained from the SCAN command, more immediate feedback and the facility

to enter SCAN from READ.

2) improved wording of help information and the inclusion of more information on screen (more lines per page).
3) to include type ahead menu suppression and less options for experienced users.
4) to improve the speed of response.

Some general comments were that the interface b was 'fun' to use and although unfamiliar thought they could adjust. It was thought to be better for new users than the existing interface.

6.4 Direct Observations and other Considerations

Although no formalised direct observation measurements were made, many of the above points were noted by the experimenter and some other interesting points were perceived as important during the experiment and will be mentioned briefly here.

When using the intelligent interface to send a letter after the letter is posted by the f4 - '.send' c/r command the subject is then taken back to the envelope option. This was confusing for some subjects and so a better solution would be to go to the mail menu instead. The user could then reselect the send task if required with very little overhead in key selections. The form envelope was coped with very well by most subjects although the options could be made clearer to aid understanding.

It became obvious during the experiment that many experienced subjects were unfamiliar with the use of menus which meant that some added a terminating c/r to each menu selection or tried to enter the relevant Telecom Gold command at the menu level. In some cases this caused confusion and also caused them to 'miss' the letter and have to find the letter again. If the use of menus had been made clearer this may have meant the read task times would have been different and that the performance of the intelligent interface would have been better.

The speed limit in using the interfaces is obviously someone who is familiar with both styles. So an informal experiment was carried out with the author completing the tasks as quickly as possible. The results were very similar on each attempt and are shown below.

	SYSTEM	GOLD=1	NEW=2
	'LOGON'=1	61	68
TASK	'SEND'=2	25	43
	'READ'=3	40	61

Table 6.13 Table of average times to complete each task

The logon tasks times were similar whereas the intelligent interface performed worse in both send and read tasks. In the send task the time difference was that to send the envelope because obviously in interface a the envelope is sent during composition whereas in interface b the envelope is composed in the form and then sent. However this time overhead is not too important considering the editing flexibility available. In the read task the added time is due to the interface trapping and translating the commands and so some overhead is unavoidable. The translation algorithm, however, could be finely tuned to reduce the actual time taken to match the local menu level to the host state.

6.5 Conclusions

For the naive subjects there is no evidence that the period affects,

1) the average time taken to complete all the tasks (period effect),
2) the time differences between the two systems, (period X system), i.e. for the total times for each system a and b;
3) the time differences between the three tasks (period X tasks)

This enabled both periods to be used in the analysis.

There is very strong evidence, significant at the 0.1% level, of a difference in the times taken to use the two systems (i.e. all three tasks) and also in times taken to complete the three tasks. Initial results show that in all cases using the intelligent interface b was quicker, easier and more satisfying than the existing interface a, except for 'logon' where it was quicker and easier but not significantly more satisfying.

In the comparison of the two systems for experienced users there was no evidence of a period X system interaction. There was evidence of a task X system interaction and interface b was only found to perform worse than interface a on the read task in terms of time to do the task, ease of use and user satisfaction.

A substantial proportion of the work reported in Chapters 5 and 6 is to be submitted for publication (Watson et al. 1986).

CHAPTER 7

GENERAL DISCUSSION AND FUTURE RECOMMENDATIONS

7.1 Introduction

The following discussion develops some of the implications of the evaluation results and identifies possible areas of future work. The chapter is divided into four main sections, implications for experimental design and user interface design, standardisation issues, technical considerations and communications for the hearing impaired.

7.2 Evaluation Implications

7.2.1 Evaluation Shortcomings

Although every attempt was made to make the experimental situation as near to the real world as possible, some elements of the trial were artificial. It is difficult to infer general conclusions from these trials because of the time limit and the population sample which may not be typical of the deaf or hearing community. The deaf sample were literate and familiar with computers. They were more socially advantaged than other deaf people and so they would have a wider geographical selection of friends. They would then have more use for long distance text telephone communication.

Another possible problem is the Hawthorne effect where 'people tend to work harder when experiencing a sense of participation in something new and special' (Bailey 1982). The acceptance by the subjects of the invitation to take part in the trial may mean that they were more tolerant of some of the interface shortcomings.

In their evaluation of the telephone bureau Grossfield et al. (1982) found that the users enthusiasm for the trial caused the subjects to organise their daily routine around the bureau opening hours. The problem in obtaining objective results is increased if the user has no experience of text communications before.

"There is a general problem in attempting to evaluate novel services for handicapped people because almost any service, providing it works for a good proportion of the time will be an improvement on the previous state." (Grossfield et al.)

This should be borne in mind in the following discussion.

7.2.2 Implications for Future Experimental Design

One of the most important points to emerge from these results is the agreement between quantitative results and qualitative results. Whichever interface was quicker it was also found to be the easiest and most satisfying to use. The implication of this is that in cases where it is too complicated or time consuming to conduct an objective analysis a subjective evaluation will yield valuable and reasonably accurate results. Generally the results are encouraging enough to suggest further development of the intelligent interface and more trials on other subject groups based on qualitative parameters only.

The results proved that the experimental design and prototype strategy based on the categories mentioned by Ramsey and Attwood were justified. The benefits of a pilot experiment outweighed any loss of time involved and helped identify many potential problems. The results also imply that the models used to describe the use of the system by the subjects were more than adequate for the two user groups tested and suggests that such a model could be used for the evaluation of other user groups. The model was also sufficiently robust to produce valid results even with the non-constancy of variance. The model could be refined further particularly if other important factors can be isolated and included as blocking factors in the design. Factors such as past experience and hearing impairment have already been used but other factors that may be important are degree of hearing impairment (e.g. whether a person is post- or pre-lingually deaf), level of literacy, etc.

Although it is difficult to extend the conclusions to include other user groups there is some evidence from the results which would suggest that this is not unreasonable. Firstly, although the deaf user group was homogeneous in terms of their experience of mailbox systems, the subjects differed in age and mailbox usage varied from more than once per day on average to once per week. These factors did not seem to affect the performance. Secondly, the two hearing subjects, whose results were not used because they did not fit the 'inclusion' criteria, still performed better with the intelligent interface. Further investigation of this group (uncomfortable computer naive university graduates) and other user groups seems advisable with future prototypes.

There is a problem here in that these subjects take longer to complete the task. This means that either each subject will only be able to do one task or that each subject may have to complete different tasks in separate sessions. This latter design would complicate the model.

There are also problems evaluating other user groups because of the natural fear of keyboards and computers which some subjects may have. It is apparent, from conversations the author has had with some hearing impaired people, that there is a misconception that to use a keyboard one must be able to touch type. This is especially true of the older generation and of some of the younger generation. The latter group, however, were keener to attempt to use equipment when it has been demonstrated to them and when they have an incentive to use it. This was particularly apparent when the author was involved in demonstrations of equipment at local deaf clubs during the project reported by Eldridge et al. (1985). This suggests that some sort of practical training may be important in the evaluation of mailbox systems, see also Walgreich (1982), to bring subjects up to a common standard and reduce variability between subjects and so increase the sensitivity of the experiment.

When the experiment starts there is a question as to whether the researcher should be there or whether a complete self-contained experiment should be produced. Although it was considered important not to pressurise the subjects so they are free to experiment with the interface, e.g. by not using videotaping, it was also necessary for the experimenter to note the subjects reactions to the experimental procedure by direct observation. It is interesting to note that even if the experimenter said nothing it seemed comforting to the subjects that someone was there in case anything went wrong.

The experiment in its present form cannot evaluate the learning curve of the interface and indicate how long it would take for someone to become a proficient user of the dumb terminal or intelligent terminal. Further evaluation is therefore recommended to investigate the long term performance of the interfaces by the subjects using the system in home trials.

It is possible that further analysis of the monitoring software results and questionnaire answers may reveal information about the task times and the causes of errors. From informal conversations with the subjects after the evaluation, it appeared as if most of them enjoyed doing the questionnaire and found it easy to understand. The format of the manual could be improved particularly in the area of better graphics and layout but was considered more than adequate by most subjects who used it. In retrospect the questionnaire should have endeavoured to discover what are the most common commands used by each subject.

7.2.3 Implications for Interface Design

It is apparent from the evaluation that the intelligent terminal should be pursued further as it seems to be the best solution to the problem of public to private domain transition. The power and flexibility of the interface in being able to mediate the communication between the user and a variety of unmodified computer text communications systems, such as mailbox systems, suggests that this is a potentially viable and commercial solution to the problem of terse user-initiated interfaces. There are, however, many improvements that will be required before the prototype interface is suitable as a finished product and there are also some suggestions that could be used to improve existing mailbox user interfaces.

The existing Telecom Gold interface obviously performed worse than the intelligent interface for the naive users by a factor of two. The general conclusion was that the existing interface was unfriendly, unpredictable and confusing. The main improvements required were a more consistent interface, more help facilities and the intelligent interface was recommended for naive users by both user groups. There was a particular need for a further prompt after the scan and read commands had been selected. There is the facility on Telecom Gold to personalise the mailbox with special prompts and for the user to create abbreviations for the combinations of well-used commands. It is possible that further investigation could reveal whether these resources will improve the interface characteristics.

Generally there is a need where there are limited facilities in the intelligent interface to have 'expert' facilities, such as type ahead menu-suppression and an improved advanced user interface. At a later stage it may be better to include a hierarchical help menu structure locally for the advanced user where function key menus are available to enable the user to learn about the system in a progressive fashion and to provide short term memory support. Further work should be done on the transition between the two interfaces to determine the optimum changeover conditions.

The intelligent interface performed well with both groups but the following improvements seem appropriate from the comments made. The 'compose an envelope' procedure should show a better indication of how to move fields. In fact, an investigation of the interface by the author after the evaluation identified some other errors in the interface that did not become apparent during the evaluation.

In the 'send' task, there was some confusion when the <.> character was preprocessed to produce a <. > as some subjects thought they had accidentally pressed <space bar>. So another method of preprocesing the 'full stop' to prevent use of the 'dot' commands may be required.

The 'read' task could be improved by,

a) allowing the user to specify a requested letter by name or subject as well as by letter number,

b) clearer layout of read menu. Direct observation showed that subjects had difficulty identifying the options available.

c) facility to enter SCAN from READ. Direct observation showed that the facility to enter READ from SCAN and SEND from READ would also be useful.

d) change the algorithm so the more action prompt always appear in the 'read' task and the unexpected prompt only appears in the 'scan' task.

The logon task design, with the interim responses, proved most satisfactory. Improvements that could be included may be to use the whole screen for the interim messages. Although the directory function was not evaluated (only one number could be dialled), in the future it would seem logical to separate the local directory service into sections based on the different conceptual models of the database, mailbox system, conferencing system and telephone.

The evaluation shows that the conceptual models used in the interface were valid except for minor changes to the letterbox/postbox metaphor being appropriate. Further work is required on the other models mentioned in chapter 4 and to specifically identify what sort of role a batch interface should play in the intelligent terminal design and the form of this interface.

The interactive communications and CHAT model require investigation in a similar manner to the online/offline and letterbox/postbox model. This model more accurately reflects the telephone model. Particular reference should be made to what style of screen interface is required. For example this could be.

a) a split screen (vertical split or horizontal split). There is some case to support this method as it a) allows the recipient to answer before the question is received and b) allows interruptions to be made. This is considered to be rude by some members of the deaf population who have used this method, or

b) a continuous screen where interruptions are allowed but would be interspersed in the text of the other person. Identification of speakers may be difficult here (especially if an online conference is set up).

In the case of direct point-to-point communications between two people it is a simple matter to separate the communications. In the indirect case (CHAT) this is more difficult because the host computer echoes both people conversations to both terminals.

Following on from this further work could be carried out on a comparison of the effectiveness of direct point-to-point communications such as text telecommunications, handwriting, visual teleconferencing (Robinson and Pearson 1984) and the use of shortforms (Damper et al. 1986) to increase the information throughput.

7.3 Standardisation Issues

7.3.1 CCITT standards

Although the intelligent interface showed much potential in terms of ease of use, satisfaction and time to use, the most pressing problem that must be investigated is how such a terminal would fit into the CCITT standards. The obvious next stage is to implement an intelligent terminal which can communicate with a mailbox system which implements a CCITT X400 standard.

Further it seems as it there may still be non-CCITT mailbox systems around for some time so intelligent terminals will have to communicate with both types of systems. The problems experienced by Kaye and McDowell (1984) and ourselves must be followed up. An obvious solution to the problem associated with the proliferation of multiple mailbox systems is the implementation of CCITT X400 standards on existing systems which would enable the interconnection of different messaging services.

To enable intelligent terminals to impose alternative dialogue styles in a more elegant way, it would seem appropriate to include in the standard dealing with the mediation with the local terminal some sort of dialogue markers, "invisible' to the user, which could be used to keep track of the state of the host. Better error correction facilities would have to be employed as the transmission of such markers would be subject to the same interference as transmission of the textual content of mail. If this does not happen, more complex algorithm may be required to ensure the state of the local machine matches that of the centralised host.

7.3.2 Telephone communications standards

Most telephone lines which connect home subscribers to text communications systems are 2-wire on the local loop and hence to the PSTN or PSS network. Connection of the computer to the telephone network is via the modem. This changes the digital signals from the computer into audio signals that can be transmitted down the voice grade 2-wire link and converted back at the other end. If two simultaneous channels on the line at the same time, (i.e. full duplex transmission), frequency division multiplexing is needed. Due to the bandwidth limitation and the need for guard bands, the upper rate for FSK transmission is 300/300 baud full duplex and 1200/1200 baud for half duplex operation. Currently 1200/1200 full duplex is the highest speed used by Telecom Gold which is a significant limitation on the information flow.

The new System X digital telephone network (which operates at 64kbits/sec) is called the ISDN (Integrated Service Digital Network) and is being installed in the U.K. at present. In theory once the ISDN is established there should be no barrier to high speed subscriber to subscriber communications. However by 1990 only 50% of the trunk network will have been changed to an optical network. The greatest limitation of the service will be the analogue link in the ISDN because even by the year 2000 there will still be local subscriber 2 wire links so the local intelligent interface will still be required.

7.3.3 Intelligent Interface Model

It is clear that the conceptual view of the intelligent terminal could have been expanded prior to the actual interface design. This would have enabled an interface to be designed which was more general in its application and was linked to various knowledge bases which could easily be modified or extended. The sort of conceptual view of the intelligent terminal which this produces is shown in figure 7.1.

Here we see that the local environment consists of an interface with associated user interface profile store, a structure store and the active mailbox which accesses the passive mailboxes, conferencing systems or point to point systems using information from the text communications profile store.

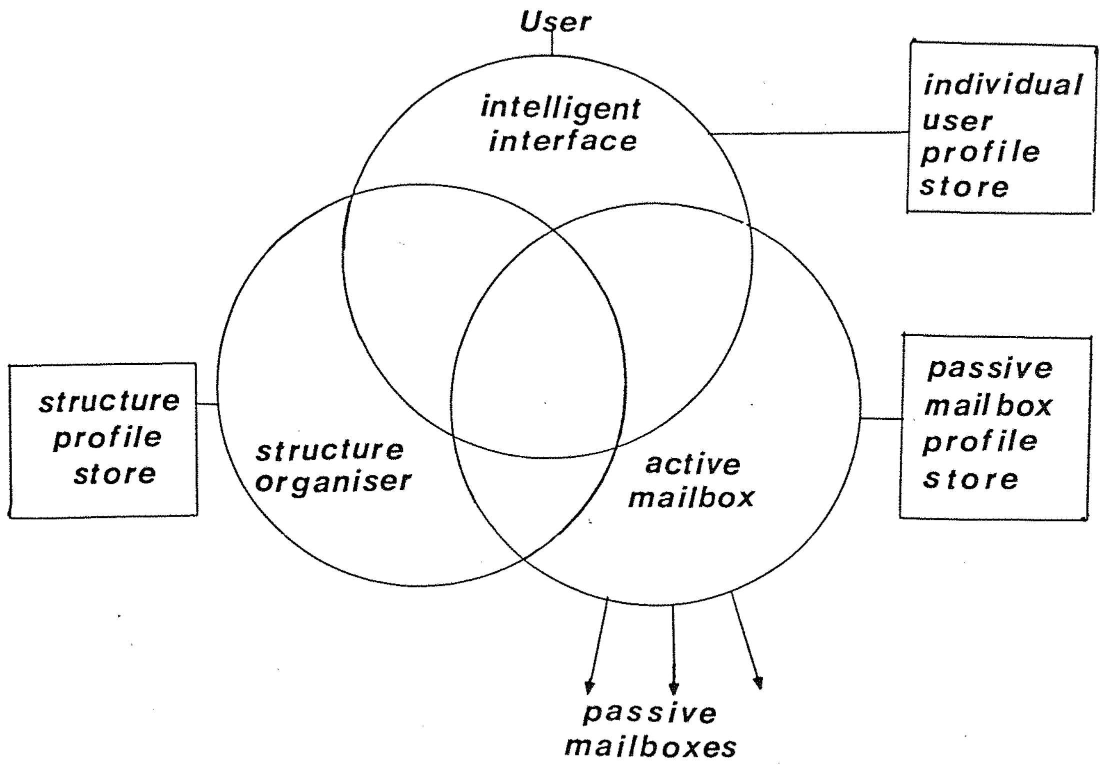


Figure 7.1 Conceptual Model of a Future Intelligent Terminal

The textcom profile store would contain the characteristics of each text system. The profile representation would take the form of a description set containing certain parameters. For example, the description could take the form of a table as follows:

<textcom name, systems type, phone number, baud rate, response code, PSS code, PSS identity, logon sequence (ID, password), command translation table, send command syntax, read command syntax, ... delete character code, break code, ... offline code ... >

The description could be implemented as a file where each member is a record and the identifier as the record key. The relevant characteristics can be read from a local knowledge base prior to access to the system. The advantages of this method are that the program can use the standard file access mechanisms existing in the operating system and also editors can be used to modify the records. Another approach is to use in memory tables but this uses more memory space. An early example of use of such a knowledge store is the proprietary SAGESOFT software for use with the AMSTRAD PCW8512 computer.

A potential problem is the difficulty in determining the characteristics of a new system. One alternative is for the characteristics to be preset by an engineer in a similar way to preset television channels.

In many applications it is important to have some sort of structure to the communication. For instance, a conference will be conducted quicker if it is known who is the chairman, the secretary, where sub-conferences are, etc. This implies some sort of structure parameters and these could also be stored locally and used by the intelligent terminal to conduct communication between the user and the conference i.e. a structure store (Wilson 1984b). The structure store could maintain information about the user's previous converations and help the user to cross reference them to other conferences or mailbox replies. The structure store could also keep track of who is on what system and how to keep in contact with them. This will be particularly important with the proliferation of multiple mailboxes.

Similarly the interface profile store could contain the characteristics of the user. The program could use these parameters to construct the dialogue with the user. The parameters could contain information which could be used to personalise the interface. The description could take the form of a table as follows.

<user name, average response time, ... menu level, prompt level, help message level, preferred dialogue type, visual/ non-visual cues, VDU character size, ... etc. >

This information could be configured by the individual using some high level interface specification program and again stored as a record at a local knowledge base. By developing and evaluating a prototype interface to a mailbox system, this project has helped to identify some of the user interface design features which can be used in the interface design.

7.4 Technical Considerations

7.4.1 Hardware Implications

Although at the beginning of the project the BBC microcomputer was a reasonable sophisticated home computer, in the relatively short time it has been superceded by other cheaper, faster and more flexible microcomputers. This illustrates the continual problem in that a home computer is either obselete or about to be marketed which makes isolating a potential candidate for further development difficult. In these days, however, more and more people are buying computers for a purpose and so outmoded characteristics are not such a disadvantage. An example of this is the plethora of wordprocessors based on, what is considered by many, outmoded operating systems.

In any future design a complete standalone computer system with monitor, local hardware and multi-standard modem with software controllable status switching should be used. (The modem used in the project did not have software control.) This modem should not require the user to know any of the technical requirements of the equipment and should include the following facilities: auto baud rate select, auto dial, auto call, and auto answer.

7.4.2 Software considerations

The difficulties with trapping and recognising the prompts quickly and synchronising the menu levels needs further investigation to determine the best algorithms to use since it seems unlikely that some sort of state level dialogue markers will be implemented within current standards. Also it may be better to preprocess some of the host responses to strip off unused information. The problem here is how to define 'unused' information as what is unused to some users may be useful to others.

Further evaluation should be done on the other sections of the software that have been developed such as the use of offline composition and storage and the potential secretarial ('batch') role of the intelligent terminal. Further work will be required to test the intelligent terminal with other text communications systems and different protocols, data rates etc.

7.4.3 Software Design

The prototype interface program was developed using standard computer programming principles, i.e. state diagrams, pseudo English algorithms etc., however there are alternatives. In retrospect it would have been better to try to use a high level dialogue description language specifically optimised for interface design. Benbasat and Wand (1984) have developed a method which, with certain modifications, could be suitable for intelligent software design.

Space precludes a detailed description of this method (see Appendix 10) but further development should investigate how this method, which is based on a table driven software, compares with other dialogue generators and whether it is indeed suitable for this purpose. One of the potential advantages of this method of implementation is that it parallels structured programming.

7.5 Improvements for hearing impaired users

7.5.1 The Intelligent Terminal

It is obvious that mailbox systems as they stand have, and still do, perform useful functions for the hearing impaired. However if mailbox systems are to become more acceptable to the wider deaf and hearing population the interface must be made more usable and less costly. The project has demonstrated the importance of intelligent terminals to hearing impaired users and has showed the potential to overcome both of the above problems. Further work is required to develop a terminal designed to provide a consistent unified interface to all types of text systems within the one terminal. The interface provided showed that one could be designed for users who have little technical background of mailbox systems to learn to use in a very short time.

The design showed the importance of including various techniques to improve the bandwidth (i.e. audio to visual feedback transformation) of communications. An example of which is in the logon task where there is a need to hear the modem reply tones before initiating the dumb terminal logon task. Currently the author has observed that this is a rather hit and miss affair. Further work could extend this by using techniques such as reverse video, short paragraphs, spaces in text, segmentation, time lapses (synchronous communications only), etc. (Jones 1978).

Further use of line and tone detection equipment to enable deaf people to be able to detect the engaged, ringing and unavailable tones has been carried out by the RNID and should be incorporated in the design of the intelligent terminal.

7.5.2 Cost Analysis

The financial aspect is a significant factor which will affect the viability of using text systems for the deaf. Even if the intelligent terminal proves to be a better way of using such system it may not be an attractive proposition if the cost is too high.

Appendix 11 gives some rough calculations to show the potential installation and running costs of using two types of text communications systems. If is clear that the hearing impaired are discriminated against because of the extra equipment cost and extra running cost. Although negotiations between representatives of the hearing impaired community and the business groups involved has achieved some success (e.g. the Telecom Gold flat rate for off-peak use), further progress is required to bring the running costs within the reach of the ordinary hearing impaired person. The questionnaire showed that 4 hearing impaired people had a normal quarterly telephone bill of less then £50 and 4 people had a normal bill of £50-£100. The use of an intelligent terminal to reduce usage time has been illustrated from the evaluation and further use of local intelligence to transmit pre-prepared messages and receive mail items automatically in off-peak periods can facilitate further cost reductions.

The present cost of intelligent terminals will be a barrier to their use however microprocessor based equipment costs are dropping all the time and it is feasible that in five years a complete package, i.e. computer, monitor, modem, disc drive, printer and software, will be available for the same price as a home computer costs today. The questionnaire showed that this will be required as 11 out of 12 hearing impaired people wanted a complete system for less then £600.

7.6 Conclusions

The intelligent terminal as a mediator between users and mailbox systems shows a lot of promise and with the improvements suggested could be a useful aid to the hearing and hearing impaired in their use of text communications systems. The results of the evaluation have significant implications for the designers of intelligent terminals for mailbox systems particularly with the introduction of the CCITT X400 standards.

CHAPTER 8

CONCLUSIONS

This thesis has investigated the development and evaluation of an intelligent terminal as a means of improving the existing user interface to mailbox systems for the hearing and hearing impaired. Present work in mailbox systems concentrates mainly on the standardisation of mailbox system internal architecture and protocols, but there is little on human-computer interface specification.

Previously those with hearing impairment have communicated via the telephone network indirectly via a host computer using dumb terminals or directly using custom-built aids. One problem with these approaches is the requirement for users to have some technical understanding of the underlying technology. Also this earlier work on aids has been too specific to the user group concerned and has failed to satisfy the requirement of the hearing impaired to be able to communicate with the wider hearing community.

In the comparison of alternative text communications methods for the hearing impaired it was concluded that a modified standard aid, and particularly mailbox systems, have many advantages over custom-built or customised aids. It was recognised, however, that substantial improvements in the cost of use and a reduction in the complexity of the user interface could be produced by the use of an intelligent front end interface as a means of accessing such systems. Furthermore, these advantages were applicable to the wider hearing community. The intelligent terminal offered the added advantage to the hearing impaired of being able to provide audio/visual transformation which is specifically required by this group.

Existing text communications aids, including mailbox systems, suffer from the same problems of many mainframe computer systems, that of terse dialogues. This results from the requirement to maintain concise communications in order to maximise information flow along low data rate lines and ensure efficient use of the central processor. Consequently this project set out to investigate the potential of using an intelligent terminal to demonstrate the feasibility of imposing an alternative computer-initiated dialogue style and structure on existing systems, and to make possible a comparative evaluation of the efficiency of message transactions using each of the two different interfaces to carry out standard tasks.

Instead of building a microcomputer based user interface in an ad hoc manner, which has created so many problems in the past, a systematic approach was adopted which included analysing the problem from a conceptual, as well as a practical, viewpoint.

This approach meant that it was important to consider potential standardisation issues in mailbox systems. In an analysis of early versions of the CCITT X400 standards it was postulated that the intelligent interface should be designed to fit into one of three places in the currently accepted model: this being an interface acting as a mediator between the user and the central host. This option offered the most flexibility in terms of the access to other text communications systems, authority boundaries and bandwidth limitations. Also it helped reduce the load on the centralised host system and could help improve the user interface characteristics.

An examination of previous work done in this area showed problems with the 'batch' method of accessing mailbox systems and particularly identified some practical problems in dealing with current non-CCITT systems. Also some potential pitfalls in user interface design were uncovered which helped in revealing a potentially useful and systematic approach to developing a prototype interactive interface.

An interactive intelligent front end interface was implemented based on a BBC micro-computer and modem. To implement the required computer-initiated dialogue style, the user interface was designed around a hierarchical menu structure and using question-answer and form-filling techniques. Unfortunately, no existing mailbox system implemented the message submission delivery protocol embodied in the CCITT X400 standards at the time the work was carried out. Therefore the intelligent terminal had to communicate directly with the mailbox system by emulating input from a user at a dumb terminal, and trapping and translating the responses of the central host computer to set local menu levels. Implementing the interface presented various practical problems particularly in the development of reliable finite state machines used to match the dialogue state of the local interface to the central processor.

Before practical trials could be conducted it was necessary to design an evaluation experiment which would allow the effects of the different interfaces to be compared and the simple elimination of nuisance effects such as subject variability and learning effects. Performance was measured quantitatively by the time to complete several typical tasks, logging on to a mailbox, sending a letter and reading a letter.

Initially a two period crossover design was developed which was tested using a pilot experiment on work colleagues. This gave an indication as to the validity of the assumptions made in the model and the practical feasibility of the experiment. It was also used to estimate the potential variability of the assessment and to isolate previously unidentified factors that could have affected the results. This enabled us to develop a better model which would take into account these new factors. Although this model complicated the experimental procedure it fulfilled the requirement to obtain statistically valid results with a small population size.

The evaluation experiment was based on a factorial approach and consisted of a design based on a 3 X 3 Latin square inside one based on a 2 X 2 Latin square. Using this design an analysis of variance could be performed on the results to determine the significance of various factors and their interaction effects.

A questionnaire was used to complement the objective performance measurements by the subjective measurements of user satisfaction and ease of use. The questionnaire consisted of two parts, pre- and post-test. The pre-test questionnaire was used to screen subjects to ensure that homogeneous groups were chosen; the post test questionnaire was used to corroborate the quantitative data.

Practical trials were conducted using two groups of subjects, naive and experienced, to establish whether the intelligent interface was a substantial improvement on the existing dumb terminal interface to a commercial mailbox system (Telecom Gold).

Few of the subjects felt any uneasiness during the experiment, most of them enjoying the exercise. For the naive user group there was significant evidence that the intelligent interface performed better than the dumb terminal in all the tasks in terms of time to use, ease of use and user satisfaction. For the experienced user group, there was no evidence that the intelligent interface performed significantly worse than the dumb terminal interface except in the 'read a letter' task. (This was seen to be due to this part of the interface being two restrictive and having some poor user interface characteristics.) The practical significance of this is that, while it might be expected that a menu-driven interface would be advantageous for naive users, a counter argument could be put that such an interface would be unpopular with experienced user, who would prefer to be in direct control of the dialogue. The results obtained from the experiment do not support this view.

The questionnaire responses of both groups provided useful information on improvements to both interfaces and comments on the evaluation experiment. Perhaps the most surprising conclusion which can be drawn from the experiment is the close relationship between the results obtained from the practical trials and those obtained from the analysis of the post-test questionnaire. This relationship suggests that, in future, a reliable estimate of the comparative performance of such interface could be obtained simply from questionnaires alone.

In the system as a whole the interface design was readily accepted however the users suggested some improvements that could be incorporated into the intelligent interface particularly in the menu layout in the 'read' task and in accessing one task from another. It was apparent that the use of menus is not as obvious as one might suspect which may suggest that type ahead menu suppression is not advisable for some users. Direct observation showed that the interface was improved by the use of interim responses but that the algorithms used to trap the host responses and translate them into menu levels could be improved to reduce execution time. The interface designer should be aware of the usefulness of interim responses particularly in telecommunications systems where response times vary.

The results from the evaluation were promising enough to suggest further development of the intelligent interface and more trials of future prototypes on other user groups, based on qualitative parameters only, seems appropriate. The results seemed to verify the design strategy and the conceptual models used. Various suggestions for future work have been generated from this work particularly as they related to user interface design, CCITT standards and to text communications systems in general. Future designers should adopt a systematic approach to develop a more general conceptual model of an intelligent terminal which is capable of interfacing to other types of text communications systems.

In summary this project demonstrated the feasibility of using an affordable, commercially available home computer as an intelligent front end interface to a mailbox system. The problem was approached in a systematic way, by examining previous text communications systems, mailbox standards, user interface guidelines and the statistical theory involved in evaluation methodology. This helped identify problems concerned with existing interfaces and led to a user interface design based on a set of well-defined principles.

The results of the evaluation showed that,

* the intelligent front end interface achieves a better performance than a dumb terminal for naive users,
* there is no evidence of a difference in the performance of both interfaces for the experienced users,
* there are certain modifications would could improve the performance of each interface,
* a robust evalution design based on a prototyping approach is very important,
* the concept of imposing an alternative dialogue style on existing mailbox systems using an intelligent interface does justify further development.

REFERENCES

The numbers in {} brackets indicate the section where this article is referenced. (e.g. {3.2} refers to Section 3.2)

Anderson, H., (1978) {1.1}
'What is electronic mail and where does it fit into the office?'
Telecommunications, 12 (November 1978 issue), 31-54

Alty, J.L., (1984) {3.7}
'Use of path algebras in an interactive adaptive dialogue system.'
IFIP Conference on Human-Computer Interaction - INTERACT '84, London, 4-7 September 1984, 351-354.

Bailey, R.W. (1982). {5.4, 6.4}
'Human Performance Engineering: A Guide for System Designers',
Prentice-Hall Publishers

Bair, J.H.(1983). {3.2, 3.6}
'Avoiding working non-solutions to office communications system design',
OAC '83 Conference Digest. The 4th Annual Office Automation Conference Philadelphia, 115-119, February 1983.

Benbasat, I., and Wand, Y. (1984). {7.4}
'A structured approach to designing human-computer dialogues',
International Journal of Man-Machine Studies, 21, 105-126.

Besson, R. (1984). {2.3, Appendix 11}
'A communication means for the hearing and speech deficient through the telephone network using MINITEL terminals',
Proceedings of the Symposium on Text communications for the Deaf, The Hague, 25th/26th October 1984, 93-100.

Bolle, G. (1984). {Appendix 11}
'Some financial aspects of text communications.',
Proceedings of the Symposium on Text communications for the Deaf, The Hague, 25th/26th October 1984, 133-140.

Breakthrough Trust and Department of Industry (1984). {2.3, 2.5}
'Visicom Final Report',
Breakthrough Trust, Charles W. Gillett Centre, Birmingham.

Brigham, F.R. (1975). {5.4}
'Some quantitative considerations in questionnaire design and analysis',
Applied Ergonomics 1975, 6.2, 90-96

Carey, T. (1984). {2.6}
'Dialogue handling with user workstations',
Interact '84 First IFIP Conference on 'Human-Computer Interaction', 4-7 September 1984, 2 30-34.

CCITT (1983). {3.2}
'X400 (X.MHSI1,REV-19), Message handling systems: system model service elements',
Draft recommendation of the Special Rapporteur on Message Handling held at Brighton 24th October 1983.

Chatfield, C. (1983). {5.3}
'Statistics for Technology', 3rd Edition.
Book;- Chapman and Hall Publishers ISBN 0-412-25340-2

Cunningham, I. (1984). {3.2}
'Electronic mail standards to get rubber-stamped and go worldwide',
Data Communications (March issue), 159-168.

Damper, R.I., Smith, J.W. and Dabbagh, H.H. (1986). {7.2}
'Shortforms: abbreviated natural language for computer text entry'.
Manuscript in preparation. To be submitted to IEEE transactions on Systems, Man and Cybernetics.

Davis, A..C. (1983). {1.2}
'Hearing disorders in the population: First phase finding of the MRC National Study of Hearing',
In: Hearing Science and Hearing Disorders (1983). Eds. M.E. Lutman and M.P. Haggard. Academic Press, London, 35-60.

Denmark, J.C. et al. (1979). {1.2}
'A Word in Deaf Ears - A Study of Communication and Behaviour in a sample of 75 Deaf Adolescents'. Undertaken jointly by Department of Psychiatry for the Deaf, Whittingham Hospital, Lancashire and the Social Research Branch, Department of Health and Social Security, London.
Published by Silverdale Press, Hayes, Middlesex, 1979. Research team J.C. Denmark, M. Rodda, R.A. Abel, U. Skelton, R.W. Eldridge, F. Warren and A. Gordon.

Dullemond, J. (1984). {2.2}
'International Text Communication',
Proceedings of the Symposium on Text communications for the Deaf, The Hague, 25th/26th October 1984, 83-88

Eason, K.D., (1982). {3.6}
'Human Factors in Information Technology',
Physics in Technology (G.B.) 13 No.5 (September 1982), 196-201.

Ehrich, R.W., (1983). {2.6}
'Research on Human-Computer Interfaces at Virginia Tech',
SIGCHI Bulletin, 14 No.3, 16-20.

Eldridge, S.P., Fieldhouse, Y., Lau, K.S., Riddle, S., Southard, R., Townshend, R.J., and Wood, H. {5.1}
'An investigation into telephones for the deaf',
Fourth Year Course in Engineering, Multidisciplinary Project report, Southampton University, Department of Electronics and Information Engineering.

Gibbons, J.D. (1971). {5.4}
'Non-parametric statistical inference'. McGraw-Hill.

Gilb, T. and Weinberg, G.M., (1977). {3.6}
Book - 'Humanised Input'. Winthrop Publishers.

Glaser, R.E., (1981). {2.2}
'A Telephone Communication Aid for the Deaf',
Proceeding from the John Hopkins First National Search for

Applications of Personal Computing to Aid the Handicapped. IEEE Computer Society, (October 31, 1981), 11-15.

Grizzle, J.E. (1965, corrected 1969). {5.4}
'The two period change-over design and its use in clinical trials',
Biometrics, 21, 467-480

Grossfield K., King R.W., Martin M.C.,and Tolcher D.J.,(1982). {2.3, 6.4}
'A Feasibility Trial of a Bureau-based Telephone Service for the Deaf',
British Journal of Audiology, 16, 27-38.

Haggard, M., Gatehouse, S. and Davis, A. (Institute of Hearing Research, Nottingham University, 1983). {1.2}
'The High Prevelance of Hearing Disorders and its Implications for the Services in the UK',
British Journal of Audiology, 1981,15, 241-151.

Hammer, M., Kunin, J.S., and Schoichet, S. (1983). {3.6, 3.7}
'What makes a Good User Interface',
OAC'83 Conference Digest, (February issue 1983), 121-130.

Hills, H. and Armitage, P. (1979). {5.3}
'The two period cross-over clinical trial',
British Journal of Clinical Pharmacology, 8, 7-20.

Hiltz, S.R. and Turoff, M.(1978). {2.3, 3.6}
Book - 'The Network Nation'. Addison-Wesley Publishing Company.

Housman, E.M. (1980). {2.3}
'Online Communications By computer conferencing and electronic mail',
4th International Online Information Meeting, 129-135.

Institute of Hearing Research, Nottingham University (1981),{1.2}
'Population study of hearing disorders in adults: preliminary communications',
Journal of the Royal Society of Medicine, 74, 819-827

Institute of Hearing Research , Nottingham University (1982),{1.2}
'The National Study of Hearing Prevalence and population parameters of

auditory impairment in the adult population of Great Britain', Paper given to the Epidemiology section of the Royal Society of Medicine, November 1982.

James, E.B. (1981). {2.6, 3.4, 3.7}
'The User Interface: How We May Compute',
In - "Computing Skills and the User Interface" M.J. Coombs and J.L. Alty (Editors), 337-71.

Johnson, A.B. and Hagstad, R.F. (1981). {2.2}
'DTMF Telecommunications for the Deaf and Speech Impaired',
Proceeding from the John Hopkins First National Search for Applications of Personal Computing to Aid the Handicapped. IEEE Computer Society, (October 31, 1981), 29-32.

Jones, P.F. (1978). {3.6}
'Four Principles of Man-computer Dialogue',
Computer Aided Design, 10(3), 197-202.

Judson (1986). {2.5, Appendix 11} Personal Correspondence.

Kaye, A.R. and McDowell, R. (1984). {3.4}
'A User Agent for Multiple Computer-Based Message Services',
In 'Computer Message Services': Proceedings of the IFIP 6.5 Working Conference. Nottingham, England, May 1984 125-134

Kidd A.L., (1982). {3.6, 3.7}
'Man-machine dialogue design', Research study Vol 1/Study 1/ April 1982.
Published by Martlesham Consultancy Services, British Telecom Reseach Laboratories, Ipswich.

King-Beer, M. (1973). {2.2}
'The Silent Communicator - The Phonetype',
Aural News, No. 120 Summer 1973. Middlesex and Surrey League for the Hard of Hearing, p9.

Krause, M. (1984) {2.2}
'Presentation of text-telephones used in Germany',

Proceedings of the Symposium on Text communications for the Deaf, The Hague, 25th/26th October 1984, 59-64

Levitt, H., (1981). {2.2}
'A Pocket Telecommunicator for the Deaf',
Proceeding from the John Hopkins First National Search for Applications of Personal Computing to Aid the Handicapped. IEEE Computer Society, October 31, 1981, 39-41.

Marshall, C.R. (1985). {3.2, 3.5}
'More Bite for Mailbox Users',
In Infotech - State of the Art Report: Man-machine integration edited by N. Bevin and D. Murray, Feb/March 1985.
Published by Pergammon Press.

Martin, J. (1973). {3.7}
'Design of Man-Computer Dialogues', Prentice-Hall.

McCann, P.H. (1983). {3.7}
'Development of the User-Computer Interface',
Computers and Education (G.B.), 7, No.4, 189-196.

McManis, C. (1985). {3.1}
'Local power in a remote link',
BYTE, December 1985, 10, no.12, 251-258.

McQuillan, J.M. (1981). {3.7}
'Computer Communications in the Automated Office',
Electro' 81 Conference Edinburgh 33/2/1-6.

Miller, R.H. and Vallee, J.F. (1980). {1.1, 2.3, 3.7}
'Towards a formal representation of Electronic Message Systems',
Telecommunications Policy, 4, no. 3, 79-95.

Montgomery, I. and Benbasat, I. (1983). {1.1}
'Cost/Benefit Analysis of Computer Based Message Systems',
Management Information Systems Quarterly, March 1983, 1-14.

Myers, W. (1982). {2.7}
'Personal Computers Aid the Handicapped',
IEEE Micro, (February issue, 1982), 26-40.

Palme, J. (1984). {3.2}
'You have 134 Unread Mail! Do you want to read them now?',
In 'Computer Message Services', Proceedings of the IFIP 6.5 Working Conference on Computer Based Message Systems, Nottingham, 1-4th May 1984, North Holland Publications, 1984, 175-184.

Panko, R.R. (1981). {3.7}
'Standards for electronic message systems',
Telecommunications Policy, 5, No.3 181-197.

Ramsey, H.R., Atwood, M.E., and Kirshbaum, P.J. (1978). {3.6}
'A critical annotated bibliography of the literature on human factors in computer systems'. (Technical Report No. SAI-78-070-DEN).
Englewood, Colorado: Science Applications, Inc., May 1978. (NTIS N. AD A058081).

Ramsey, H.R. and Atwood, M.E., (1979). {3.6}
'Human factors in computer systems: A review of the literature', (Technical Report No. SAI-79-111-DEN). Englewood, Colorado: Science Applications, Inc., September 1979. (NTIS N. AD A075679).

Ramsey, H.R. and Atwood, M.E. (1980). {3.6}
'Man-Computer Interface Design Guidance: State of the Art',
Proceeding of the Human Factors Society - 24th Annual Meeting, 85-89.

RNID 1980 (Revised 1980). {2.2}
'Teletext, Prestel, The Deaf Telephone', 'Hearing' Supplement, 289-296.

RNID (1981). {2.2}
Hearing July/August 1981, 168-169.

Robinson, J.A., and Pearson, D.E., (1984). {7.2}
'Visual Teleconferencing at Telephone Data Rates'.
Proceeding of the International Teleconferencing Symposium, 3-5 April 1984, Intelsat, 386-393.

Sandberg, A., (1979). {1.1, 3.7}
'Electronic Mail and Message Systems',
In - Convergence: Computers, Communications and Office Automation - State of the Art Report, 201-207 Infotech International.

Schicker, P. (1982). {3.2, 4.8}
'The IFIP model of a Computer Based Message System',
In 'Office Information Systems', edited by N.Naffah, published by INRIA/North-Holland Publishing Company, 1982, 253-267.

Selander, S.E. (1982). {3.6}
'Several approaches for Improving User cordiality in the design of Online Systems for Novice Users',
Computer Personnel 9, No. 2, 18-21.

Shackel, B. (1982). {2.3}
'The BLEND System: Programme for the study of some 'Electronic Journals',
The Computer Journal, 25, No. 2, 1982, 161-168.

Sharpless, G.T. (1979). {3.4}
'An Intelligent Home Information Terminal',
In - "Electronic Text Communications", Berlin, Germany, 131-138.

Shneiderman, B. (1980). {5.3}
'Software Psychology: Human Factors in Computer and Information Systems', Winthrop Publishers Inc. ISBN 0-87626-816-5

Sinclair, M.A., (1975). {5.4}
'Questionnaire Design',
Applied Ergonomics, 1975, 6.2, 73-80

Smith, H.T., (1982). {2.3}
'User Issues in Network Communications systems',
Man-Machine Systems International Conference, Manchester 107-112.

Sommerville, I. (1981). {3.5}
'Providing the User with a Tailor-made Interface',
Systems Architecture 6th International Computing Symposium, London,
321-329.

Stevens, G.C. (1983). {3.7}
'User-friendly computer systems? A critical examination of the
concept',
Behaviour and Information Technology, 2, No.1, 3-16.

Stoffels, B. (1980). {2.2}
'Telecommunications Devices for the Deaf',
Telephone Engineer and Management, October 1, 1980.

Strom, B.I. (1982). {2.3}
'Computer Conferencing - Past, Present and Future',
In "Office Information Systems", edited by N.Naffah, published by
INRIA/North-Holland Publishing Company, 1982, 287-309.

Telecom Gold (1983). {2.5, 4.2}
'Electronic Mail Reference Manual'.

Trundell, L. (1983). {Appendix 1)
'New Technologies for Electronic Communication'
Online National Meeting, 555-559.

Ulrich, W.E. (1980). {3.7}
'Introduction to Electronic Mail',
NCC'80 Conference Record, (AFIPS: May 1980), 485-488

Vanderheiden, G.C. (1981). {2.5}
'Practical applications of microcomputers to aid the handicapped',
Computer, 14, 54-61

Vanderheiden, G.C. (1982). {2.7}
'Computers can play a dual role for disabled individuals',
BYTE (USA) 7, No.9, Sept. 1982, 136-162.

Walgreich, B.J. (1982). {2.3, 3.7, 4.6, Appendix 11}
'Electronic Mail for the Hearing Impaired and its potential for other disabilities',
IEEE transactions on communications, com-30, No. 1., January issue, 58-65.

Ward, P. (1975). {2.3}
'The Silent Minority helps itself',
New Scientist Jan 1975, 65, No. 930, 24-26.

Watson, K.I., Downton, A.C., and Lewis, S.M., (1986) {6.5}
'Evaluation of an interactive user agent for computer based message systems'
Manuscipt in preparation. To be submitted to Behaviour and Information Technology.

Welsh, J.A. and Wilson, P.A., (1981). {1.1, 3.7}
Book - 'Electronic Mail Systems - A Practical Evaluation Guide',
NCC/Wiley Publications.

Wilhelmsson, A. (1984). {2.2}
'Diatext',
Proceedings of the Symposium on Text communications for the Deaf, The Hague, 25th/26th October 1984, 55-58

Wilson, P.A. (1983). {2.3}
'Commercial Electronic Mailbox Systems', NCC Publications, 1983.

Wilson, P.A. (1984a). {3.2}
'Standards and the Electronic Mailbox', NCC Publications, 1984.

Wilson, P.A. (1984b). {7.3}
'Structures in Mailbox System Applications'
In 'Computer Message Services', Proceedings of the IFIP 6.5 Working Conference on Computer Based Message Systems, Nottingham, 1-4th May 1984, North Holland Publications, 1984, 147-164.

Wilson, P.A., Maude, T.I., Marshall, C.J., and Heaton, N.O.,(1984). {3.3}
'The Active Mailbox - your on-line Secretary',
In 'Computer Message Services', Proceedings of the IFIP 6.5 Working Conference on Computer Based Message Systems, Nottingham, 1-4th May 1984, North Holland Publications, 1984, 135-146.

APPENDIX 1

Comparison of Distributed and Centralised Networks

Distributed network	Centralised network
1. Handle messages locally.	1. Provides a centralised "mailbox" for each user.
2. Each node has the capacity to compose and store text to be transmitted.	2. Gives user access to text editing and message storage of host computer.
3. Data transmitted node to node over regular phone lines so quality is important and the costs are dependent on distance. Best suited to local message traffic.	3. User logs onto his mailbox to read messages and can send messages to other users.
4. Message transmission can be delayed because line is busy although autodialling helps.	4. No interference problems as recipient is not required to be present.
5. Messages sent to one node at a time, not easily adaptable to multiple copies.	5. Can easily send multiple recipient copies.
6. Local equipment must be ready to receive at all times so workstation cannot be used for offline activities unless it is multi-tasking.	6. Equipment only required when calling host computer.
7. Local facilities may include auto dial, auto answer, data storage in non-volatile memory, e.g. microcomputer - note extra software required to perform extra functions.	7. Special capabilities are not required. Any standard dumb ASCII terminal and modem will do. However an intelligent terminal can be used to advantage for text creation prior to logon. Need file storage over long period.
8. Need to set a standard data transfer rate.	8. No problem with different rates.
9. Automatic transmission of messages overnight is possible	9. Must be able to accomodate transmission of long messages

(Trundell 1983)

APPENDIX 2

TABLE A-2 FSK PROTOCOLS

TABLE OF PROTOCOLS USED ON THE TELEPHONE NETWORK

MODEM PROTOCOL CCITT	USED BY	BAUD RATE	DUPLEX	TRANSMIT FREQUENCIES(Hz) SPACE	TRANSMIT FREQUENCIES(Hz) MARK	RECEIVE FREQUENCIES(Hz) SPACE	RECEIVE FREQUENCIES(Hz) MARK	ANSWER TONE (Hz)
V21 originate	DATEL 600	300	FULL	1180	980	1850	1650	----
V21 answer		300	FULL	1850	1650	1180	980	2100
V23 mode 1	DATEL 200	600	HALF	1700	1300	1700	1300	2100
V23 mode 2	ON TELECOM GOLD	1200	HALF	2100	1300	2100	1200	2100 *
V23 back		75	--	450	390	450	390	---- **
Examples:								
V23 back	VISTEL TO VISTEL	75\75	HALF	450	390	450	390	450 ***
V23 mode 2 & back	VISTEL TO PRESTEL	1200\75	FULL	2100	1300	450	390	390
V23 mode 2 & back	PRESTEL TO VISTEL	75\1200	FULL	450	390	2100	1300	2100

* The Prestel terminal protocol has binary 1 idle tone transmitted continuously between characters. A 2100 Hz tone indicates a start bit(0).

** The Prestel terminal uses the V23 mode 2 and back protocols to form a full duplex channel. The Prestel terminal is a full duplex terminal and the Prestel computer acts as a half duplex terminal.

*** A calling Vistel always generates a 1300Hz tone. If it receives a 450Hz tone it creates a half duplex channel with one of its own kind and if it receives a 390Hz tone it sets up a full duplex channel with a Prestel terminal using the V23 protocol (i.e. it simulates a Prestel computer).

APPENDIX 2

A Brief Outline of the CCITT V22 Protocol

This protocol is to be used with modems operating on the PSTN in duplex. The channel separation is by frequency division and the modulation is differential phase shift keying. The DATEL 1200 modem uses alternative B.

Alternative B

1200 bps synchronous

1200 start/stop (8,9,10 or 11 characters)

600 bps synchronous

600 start/stop (8,9,10 or 11 characters)

Line signals

(1)Carrier tone and Guard tone

The carrier frequencies are 1200Hz(±0.5Hz) for the low channel and 2400Hz(±0.5Hz) for the high channel. A guard tone of 1800 Hz(±20Hz) is transmitted when the modem is transmitting in the high channel but not with the low channel.

(2)Encoding of the data bits

The modem can transmit only at 600 baud so to obtain a data rate of 1200 bps the data is divided into groups of 2 consecutive data bits (dibits). Each dibit is encoded as a phase change relative to the phase of the preceding signal element(see table below).

Each bit at 600 bps is encoded as a phase relative to the phase of the preceding signal element.

DIBIT VALUE 1200 bits/s	BIT VALUE 600 bits/s	PHASE CHANGE (DEGREES)
00	0	+90
01	-	0
11	1	+270
10	-	+180

Note: Scrambled binary is used.

A P P E N D I X 3

BBC HARDWARE DETAILS

The BBC computer contains, amongst other things, 32k of user RAM, two parallel ports and an RS423 port (an RS232 variation). The RS423 port is used to communicate with the modem: a TANDATA TM200 model. This model is capable of being manually switched to operate at V21, V23, reverse V23, and 1200/1200 half duplex. In the majority of the development the V23 protocol was used to communicate with the host. The modem is connected to the telephone line via a standard 600 series connector. The automatic dial modem could hold a page of eight numbers and this page could be sent to the BBC via the RS423 link but transmission was so slow it was saved on disc for quick access. Once connected to the host modem the modem arranged the protocol and if the host modem responded a transparent link was provided between the microcomputer and the host.

A standard colour monitor and twin 5 1/4 inch double sided disc drives were used in the development. The firmware consisted of an ACORN OS v1.2, ACORN DOS v1.2, and a BASIC II interpreter. A program from a proprietary magazine was used to enable the word-processor WORDWISE to be used as a screen editor. The 32K memory was a large restraint and so disc overlays were used. This provided a time overhead in some cases but appropriate anticipation of user requirements and use of interim messages alleviates these problems. (An added advantage to this approach was the forced, though natural, structuring of the program.) The present cost of the equipment would be about £1500.

The BBC operates in seven screen modes. MODE 7 was used because it seemed the most flexible in terms of the low memory requirements, window definition, colour display, visual clarity and teletext compatibility (for possible use with the PRESTEL mailbox). The screen size was 40 characters wide by 25 deep. Windowing was possible but only one window could be active at any one time. Informal trials showed that from the range of colours available the most distinct combination were blue on white, blue on yellow, white on red and the reverse combinations. However green, yellow, blue and red on black were also visible.

APPENDIX 4

PSEUDO BASIC VERSION OF OFFLINE AND ONLINE PROGRAMS

1) OFFLINE PROGRAMS

```
START Program
      start_of_experiment is set TRUE:Load and run "OFFLINE"

OFFLINE program -  program to do terminal functions
BEGIN
      Initialise memory to accept overlays
      ON ERROR GOTO ERROR routine
      PROCoverlay("setup"):PROCbegin
      PROCoverlay("evaluat"):PROCevaluation
      PROCoverlay("CALL"):PROCassess_call
END
      :
DEFPROCoverlay(F$)
      Display "Please wait for a few seconds..." on bottom line
      IF F$ is NOT null THEN loads overlay F$ on top of the program
         ELSE deletes overlay if error has occured
      delete 'Please wait' message
      ENDPROC
      :
DEFPROCkeypress(word$): Store time and string word$ on disc file
      ENDPROC
      :
DEFPROCrx
      REPEAT
         IF RS423 input buffer NOT empty THEN PROCrx_char(TRUE)
         UNTIL RS423 input buffer empty or keyboard buffer is NOT empty.
      ENDPROC
      :
DEFPROCrx_char(display%)
      Uses the machine code compiled by PROCcompile to receive a
      character from the RS423 buffer and display it on the screen.
      IF this character is required by a routine (display% variable is
      FALSE then the character is stored in a global variable for
      future use. A delay is included for certain baud rates.
      ENDPROC
      :
DEFPROCupload
      This opens the disc file specified, sends characters byte by
      byte from the file using PROCtx and closes the file.
      ENDPROC
      :
DEFPROCreset
      Sends the reset sequence to the modem using PROCtx without
      receiving any characters while transmitting and checks for a
      response. If there is no response the program stops and a
      message (37) is displayed otherwise the program continues.
      ENDPROC
      :
DEFPROCt_out(adval%,time%)
      Waits until the RS423 input buffer has < adval% characters in or
      the time elasped > time%. IF the latter is true a global
      variable T% is set FALSE.
      ENDPROC
      :
DEFPROCmessage(m%)
      PROCremember:PROCsmall_w
      Set up screen message format e.g. figure 4.5
      Uses the message procedures PROCm% to display the text of the
      message in the formatted window:PROClarge_w:PROCrestore:ENDPROC
```

```
REMark The following procedures and functions are identical to those
in the ONLINE program
DEFPROCtx(Y%)
DEFPROCsetup_RS423_window
DEFPROCsetup_message_window
DEFPROCremember
DEFPROCrestore
DEFPROCdelay(wait%)
DEFFNletters(length%)
ERROR resets some parameters and displays error messages:END

Overlay :2.U.SETUP
DEFPROCbegin:PROCinit:PROCcompile:ENDPROC
      :
DEFPROCinit
      This initialises string variables that would otherwise take up
      extra memory space when used and also parameters such as arrays
      function keys, baud rates, printer status, time, and screen mode.
      ENDPROC
      :
DEFPROCcompile
      Thiscompiles a machine code routine and stores it in computer
      memory to be called from the BASIC program (PROCrx_char). It is
      required because the BBC is to slow to accept characters from
      the RS 423 port at 1200 baud. Many of the ASCII character sent
      from the modem are hardware dependent and so this routine
      decodes the input and uses it to ignore certain characters and
      intercept others to carry out certain other functions.
      ENDPROC

Overlay  :2.U.EVALUAT
DEFPROCevaluation
BEGIN {Algorithm of evaluation introduction}
      IF NOT start of experiment THEN
            BEGIN{display re-entry information}
               display redial information screen (figure 4.18)
               when user presses SPACE BAR save time on disc
            END
         ELSE IF start of experiment is TRUE
          THEN
            BEGIN {initial screens}
               input subject name,interface to test, sequence
               open data file and store name and interface on disc
               start of experiment is FALSE
               IF logon task is first THEN
                BEGIN{introductory comments}
                 display first information screen (figure 4.3)
                 when user presses SPACE BAR save time on disc
                 display second information screen (figure 4.11)
                 when user presses SPACE BAR save time on disc
                END
               ELSE IF logon task is NOT first THEN
                BEGIN {screen for researcher}
                 display second information screen (figure 4.11)
                 when space bar is pressed save interaction on disc
                END
              END {initial screens}
END{Algorithm of evaluation introduction}
ENDPROC
```

Overlay :2.U.CALL Contains procedures to access TELECOM GOLD

```
DEFPROCassess_call
      Loads modem phone numbers off disc and display on screen.
      REPEAT
          PROCmessage(34):Set cursor at line 22 x position 29
          dial TELECOM GOLD number:PROCt_out(0,300)
          IF HOST responded PROCtry_to_connect ELSE PROCsorry_try_again
          UNTIL forever
      ENDPROC
      :
DEFPROCtry_to_connect
      PROCpage
      REPEAT
         Using PROCpage receive characters until either the RS423
         buffer is empty OR a clear screen character (ASCII 12) has
         been received (indicates the host modem has responded).
         UNTIL HOST responded OR RS423 buffer is empty
      IF HOST responded PROCsel_conn ELSE PROCsorry_try_again
      ENDPROC
      :
DEFPROCpage
      REPEAT
         PROCrx
         wait 1/10th sec in case HOST transmission has stopped briefly
         UNTIL RS423 buffer is empty OR keyboard buffer is NOT empty
      ENDPROC
      :
DEFPROCsel_conn
      IF using dumb terminal THEN PROCdec_1 ELSE PROCtry_connect
      ENDPROC
      :
DEFPROCdec_1
      IF the subject is NOT to do the LOGON task first AND the
      assessment screens have NOT been displayed THEN PROCtry_connect
         ELSE display wait message:CLOSE all disc files:PROCwhich
         load and run the ONLINE program.
      ENDPROC
      :
DEFPROCwhich
      IF the subject is to use dumb interface THEN PROCmessage(40)
         ELSE IF the subject is to use the intelligent interface
         PROCmessage(41)
      ENDPROC
      :
DEFPROCtry_connect
      PROCcontact
      IF system has responded THEN PROCmessage(33):PROCenter_assess
         ELSE PROCsorry_try_again
      ENDPROC
      :
DEFPROCcontact
      PROCmessage(31):PROCdelay(200):time_out%=FALSE
      REPEAT
         PROCwakeup_sys
         UNTIL RS423 input buffer has > 20 characters in OR time
         elapsed is > 7.5 seconds
      IF RS423 input buffer has < 20 characters in time_out%=TRUE
      ENDPROC
      :
DEFPROCwakeup_sys
      Send appropriate wake up characters (such as carriage returns).
      Actual characters depend on host being accessed
      ENDPROC
      :
DEFPROCenter_assess
      Using PROCinput(20) check text string (string$) to see IF it
      contains the text string "Telecom Gold".
      IF this is TRUE PROCcon_enter ELSE PROCsorry_try_again
      ENDPROC
      :
DEFPROCinput(lin_len%)
      After a 1/2 second delay use PROCinstream to obtain string$ and
      IF length of string$<lin_len% then PROCsorry_try_again
      ENDPROC
      :
DEFPROCinstream
      REPEAT
         Procedure to input characters from RS423 input buffer and
         produce a GLOBAL string variable (string$)
         UNTIL RS423 buffer is empty or stream$ length is >159
      ENDPROC
      :
DEFPROCcon_enter
      Use PROCupload to open a file and send the contents of a record
      to GOLD. This record will contain all the logon protocol
      required to enter the required system. The GLOBAL variable
      system% is used to select the required record.
      Send the XOFF characters and then load and run the relevant
      online program to allow user to access required HOST.
      ENDPROC
      :
DEFPROCsorry_try_again
      Store time and text string "Sorry" on data file
      PROCmessage(36):PROCreset_tx:PROCdelay(2)
      IF RS423 buffer is empty PROCmessage(37):REPEATforever
      ENDPROC
      :
DEFPROCm30:Display text of message 30 (figure 4.6)
      PROCkeypress("Link established check for GOLD response")
      ENDPROC
      :
DEFPROCm31:Display text of message 31 (figure 4.5)
      PROCkeypress("Try to establish link")
      ENDPROC
      :
DEFPROCm33:Display text of message 33 (figure 4.7)
      PROCkeypress("Sending user number etc.")
      ENDPROC
      :
DEFPROCm34:Display text of message 34 (figure 4.2)
      PROCkeypress(mode$)
      ENDPROC
      :
DEFPROCm36:Display text of message 36 (figure 4.10):ENDPROC
DEFPROCm37:Display text of message 37 (similar to 4.10):ENDPROC
DEFPROCm40:Display text of message 40 (figure 4.8):ENDPROC
DEFPROCm41:Display text of message 41 (figure 4.9):ENDPROC
```

2) ONLINE PROGRAMS

```
BEGIN{Program}
      Initialise memory of BBC to allow space for overlays
      ON ERROR facility set to GOTO Error Routine if an error occurs
      PROCoverlay("INIT"):PROCbegin
      IF NOT initial info screens have been displayed
          THEN PROCoverlay("eval"): PROCevalustart
      PROCstart_rx:PROCsetup_RS423_screen_window:PROCrx
      IF user is to use dumb terminal
         THEN PROCoverlay("ADASS"):PROCa_mode
         ELSE PROCoverlay("MAIL") :PROCcommand
      Reset function keys
END{Program}

DEFPROCa_mode:PROCa_cont:PROCreset_line:PROCoffline
      ENDPROC
      :
DEFPROCcommand
      PROCdisplay_command_menu:Set key variable to null
      REPEAT
         PROCrx
         IF keyboard buffer is NOT empty THEN PROCcommand_select
         UNTIL key pressed is "O" OR CONTROL f8
      PROCreset_line:PROCoffline
      ENDPROC
      :
DEFPROCcommand_select
      PROCgetkey(TRUE)
      IF key$ is "O" PROCconfirm ELSE IF key$ is "M" PROCmail_mode
      ENDPROC
      :
DEFPROCmail_mode
      PROCset_host_to_command_level:Set key variable:PROCdisplay_mail_menu
      REPEAT
         PROCrx
         IF keyboard buffer is NOT empty PROCmail_command_select
         UNTIL key$ was "Q"
      PROCdisplay_command_menu
      ENDPROC
      :
DEFPROCmail_command_select
      PROCgetkey(FALSE)
      IF key$ is "R" THEN PROCread_mode
      IF key$ is "P" THEN PROCscan_mode
      IF key$ is "S" THEN PROCletter
      ENDPROC
      :
DEFPROCread_mode
      PROCdisplay_read_menu
      REPEAT
         PROCrx
         IF keyboard buffer is NOT empty
            THEN PROCread_command_select
         UNTIL key$ was "Q" OR "M"
      IF key$ is "M" PROCdisplay_mail_menu:Reset key$ variable
      ENDPROC
      :
DEFPROCrx
      REPEAT
         IF RS423 input buffer has > 100 characters THEN PROCstop_rx
            ELSE
            IF RS423 input buffer has < 15 characters and the
            XOFF flag has been set THEN PROCstart_rx
         IF RS423 input buffer NOT empty THEN PROCrx_char(TRUE)
         IF the dumb terminal is being used
            THEN use UNTIL either the RS423 input buffer is empty or
            the keyboard buffer in NOT empty
            ELSE use UNTIL RS423 input buffer is empty
      IF RS423 buffer is empty AND time elasped has increased by a
      whole minute AND the intelligent interface is being used
         THEN PROCupdate_status_line
      ENDPROC
      :
DEFPROCrx_char(display%)
      This routine uses the machine code compiled from PROCcompile to
      receive and display a character from the RS423 buffer.
      IF display% is FALSE
         THEN store character as global variable for use in program.
      A delay is included for certain baud rates.
      ENDPROC
      :
DEFPROCtx(Y%)
      Uses operating system call to send the character Y% to RS423
      output stream then delay for a time dependent on the baud rate
      before another character can be sent. A global variable is used
      to decide whether to receive characters during the wait time.
      ENDPROC
      :
DEFPROCwordtx(word$,return%)
      Uses the PROCtx routine to send the text string word$ character
      by character without receiving any characters.
      IF return% is TRUE THEN PROCtx(c/r):PROCwait_for_response(0,750)
      ENDPROC
      :
DEFPROCstop_rx:Transmit (without receiving) 3 XOFF characters
      ENDPROC
      :
DEFPROCstart_rx:Transmit (without receiving) 3 XON characters
      ENDPROC
      :
DEFPROCdelay(wait%):waits for wait%/100 seconds:ENDPROC
      :
DEFPROCsave_last_five_lines
      IF dumb terminal save lines 19-24 on disc
      IF intelligent interface save lines 13-18 on disc
      ENDPROC
      :
DEFPROCkeypress(word$,savescr%)
      IF savescr% is TRUE THEN PROCsave_on_disc("screen")
      IF savescr% is TRUE THEN PROCsave_last_five_lines
      IF key$ was c/r THEN PROCsave_on_disc("Return")
         ELSE IF key$ was ASC(word$)=134
            THEN PROCsave_on_disc("disconnect")
         ELSE IF the key$ was delete THEN PROCsave_on_disc("delete")
         ELSE any other key$ THEN PROCsave_on_disc(word$)
      ENDPROC
      :
DEFPROCsave_on_disc(text$)
      Save on disc text string text$ and present time
```

```
ENDPROC
:
DEFPROCupdate_status_line
      PROCswitch_cursor_off:PROCremember:PROCstatus_line
      PROCsetup_RS423_screen_window:PROCrestore:PROCswitch_cursor_on
      ENDPROC
      :
DEFPROCletter
      Set up initial parameters
      REPEAT
         PROCoverlay("ENV"):PROCletter_env
         IF key$ is NOT f1 (i.e. exit% key)
            THEN PROCmessage(32):PROCset_host_to_command_level
            PROCoverlay("SEND"):PROCenvelope
         UNTIL key$ is "Q" OR "f1" OR exit% is set TRUE
      PROCmessage(34):PROCset_host_to_command_level:PROCoverlay("MAIL")
      ENDPROC
      :
DEFPROCgetkey(savescr%)
      Get a key$ from the keyboard buffer
      IF key$ is > "%" key THEN make key UPPER CASE
      PROCdisplay_key(key$)
      IF savescr% is TRUE THEN PROCkeypress(key$,TRUE)
         ELSE PROCkeypress(key$,FALSE)
      IF key$ is "O" OR (CONTROL f8)
         THEN PROCkeypress("disconnect",TRUE)
      ENDPROC
      :
DEFPROCdisplay_key(key$)
      PROCremember:PROCsetup_message_window
      Display key$ in upper R.H. corner of message window.
      Use RET for the c/r key:PROCrestore
      ENDPROC
      :
DEFFNdigits
      This procedure allows the user to input a number of up to 3
      digits (numbers are processed to remove effect of shift lock
      key) or use "Q" to quit. Delete is allowed.
      IF the user inputs a number of more than 3 digits an error
      message PROCmessage(22) is displayed and the user is asked to
      input a smaller number.
      ENDFN

DEFFNletters(length%)
      Accepts characters from the keyboard to form a string variable
      of maximum length (length%). It allows characters between ! and
      ~ inclusive and delete and uses PROCkeypress to store the
      characters on disc. " "(ASCII 32) is the quit character. It
      returns the string to the main program.
      ENDFN
      :
DEFPROCwait_for_response(char%,timer%)
      REPEAT
         IF system has timed out THEN PROCtx(13)
         PROCt_out(char%,timer%)
         IF system has timed out PROCdelay(200)
         UNTIL RS423 buffer is NOT empty
      ENDPROC
      :

DEFPROCt_out(adval%,time%)
      PROCswitch_cursor_off:PROCremember:timeout=presentTIME
      IF no. of chars in RS423 input buffer < adval% THEN PROCmessage(37)
      REPEAT
         IF number of characters in RS423 input buffer <adval% AND
         more than two seconds has passed AND the time has increase
         by a whole second THEN display no. of seconds in message(37)
         UNTIL the number of characres in the RS423 buffer > adval%
         OR presentTIME - timeout > time%/100 seconds.
      IF the latter condition is TRUE THEN timeout is TRUE
      PROCrestore:PROCswitch_cursor_on
      ENDPROC
      :
DEFPROCcheck_last_host_response
      PROCread_until_host_stops
      String variable response$=FNget_last_response_line
      REPEAT
          Check response$ is not one of the expected responses
          UNTIL response$ is recognised OR response$ is NOT recognised
      IF response$ is NOT recognised THEN A$="" ???
      ENDPROC
      :
DEF FNnot_fn_keys
      IF key$ pressed was a function key THEN return a value of TRUE
      ELSE return a value of FALSE.
      ENDFN

DEF FNget_last_response_line
      Set up initial parameters
      REPEAT
         Read the characters on the screen from position 0 into a
         string variable response$ using the machine code routine in
         PROCcompile
         UNTIL the last character at the cursor position has been read
      Restore cursor position
      Return value of response$ to procedure that called this function
      ENDFN
      :
DEFPROCtxbreakquit: Transmit break characters and the word "QUIT"
      without receiving.
      ENDPROC
      :
DEFPROCtxbreak:Transmit break characters without receiving
      ENDPROC
      :
DEFPROCset_host_to_command_level
      PROCmessage(34):initialise loop
      REPEAT
          IF the loop variable is EVEN THEN PROCtxbreakquit:PROCdelay(200)
            ELSE PROCtxbreak:PROCdelay(200)
          PROCwaitnsee:Increment the loop variable
          UNTIL the loop string variable star$=">"(i.e. the command
          prompt from GOLD has been received).
      ENDPROC
      :
DEFPROCwaitnsee
      PROCwait_for_response(0,750):Initialise text string star$=""
      PROCstar:Make star$=final character is text string star$.
      ENDPROC
```

```
    :
DEFPROCstar
    REPEAT
      REPEAT
        PROCrx_char(FALSE)
        IF not a null character THEN star$=star$+CHR$(chr%)
        UNTIL RS423 buffer is empty:PROCdelay(9)
      UNTIL RS423 buffer is empty
    ENDPROC
    :
DEFPROCstatus_line
    Displays time elasped, printer, online status in status window
    ENDPROC
    :
DEFPROCreset_line:Disconnect the modem from HOST:PROCmessage(29)
    ENDPROC
    :
DEFPROCmessage(m%)
    PROCremember:PROCsetup_message_window:Clear small screen
    Format screen as shown in figure 4.32
    IF m%=x   THEN PROCmx
    IF m%<>38 THEN PROCstatus_line:PROCsetup_RS423_screen_window
      ELSE PROCsetup_status_window:Openup whole screen
    PROCrestore
    ENDPROC
    :
DEFPROCsetup_status_window:Bottom line of screen (white text on red)
    ENDPROC
    :
DEFPROCsetup_RS423_screen_window: The top 19 lines of the screen.
    ENDPROC
    :
DEFPROCsetup_message_window: The bottom 6 lines of screen.
    ENDPROC
    :
DEFPROCremember:Stores current cursor position in large window
    ENDPROC
    :
DEFPROCrestore:Resets cursor to old position in large window
    ENDPROC
    :
DEFPROCm22:Displays text used in message 22 (similar to fig.4.32)
    PROCkeypress("Choose too large",FALSE)
    ENDPROC
    :
DEFPROCm24 and m25:Displays message 24 and 25 (figure 4.32)
    Confirm delete message:PROCkeypress(mode$,TRUE)
    ENDPROC
    :
DEFPROCm27
    Displays text of message 27. Wait while the mailbox is tested to
    see if it is open (not shown).
    PROCkeypress("Mailbox open?",TRUE)
    ENDPROC
    :
DEFPROCm28:Displays text of message 28 (not shown)
    Do you wish to continue even though the mailbox is slow.
    ENDPROC
    :
```

```
DEFPROCm29:This displays the text of message 29 (figure 4.17)
    mode$="Leaving the System"
    ENDPROC
    :
DEFPROCm32 and DEFPROCm34:(similar to figure 4.22) wait messages
    ENDPROC
    :
DEFPROCm37:Displays text of message 37 (not shown)
    PROCkeypress(mode$+" because response if slow",TRUE):ENDPROC
    :
DEFPROCm38:Displays text of message 38 (figure 4.16)
    PROCkeypress("Please check the task order",FALSE)
    ENDPROC
    :
DEFPROCdisplay_command_menu (figure 4.4)
    PROCkeypress("Beginners menu",TRUE)
    ENDPROC
    :
Error Routine:Resets some parameters and displays error message.
    END
    :
DEFPROCoffline
    IF key$ is CONTROL f8 THEN PROCfinish
      ELSE close data file and load and run OFFLINE program.
    :
DEFPROCfinish
    PROCkeypress("Tasks finished",TRUE)
    CLOSEall data files and load and run the "BEGIN" program
    ENDPROC
    :
DEFPROCconfirm
    IF the key$ is 'f6'
      THEN PROCremember:Save the lines 19-25 of screen in a
      temporary disc file
    PROCmessage(38):Switch cursor off
    REPEAT
      Get a character from the keyboard and convert to UPPER CASE
      PROCkeypress(UPPER CASE character, FALSE)
      UNTIL UPPER CASE character is "Y" OR "N"
    IF UPPER CASE CHARACTER is "N" THEN PROCredisp
    ENDPROC
    :
DEFPROCredisp
    IF key$ was "O" THEN PROCdisplay_command_menu ELSE PROCint_1
    Switch cursor on
    ENDPROC
    :
DEFPROCint_1
    PROCsetup_message_window:Clear message window
    Redisplay last 5 lines of RS423 screen using temporary disc file
    PROCa_line:Restore cursor position:PROCa_window
    delete temporary file
    ENDPROC
    :
DEFPROCoverlay(F$)
    Load overlay :2.0.F$ above main program:ENDPROC
```

```
OVERLAY :2.0.MAIL

DEFPROCread_command_select
      PROCgetkey(FALSE)
      IF key$ is "A" THEN PROCwordtx("MAIL READ",TRUE):PROCread(9)
      IF key$ is "E" THEN PROCwordtx("MAIL READ EXPRESS",TRUE):PROCread(17)
      IF key$ is "U" THEN PROCwordtx("MAIL READ UNREAD",TRUE):PROCread(16)
      IF key$ is "C" THEN PROCchoose_a_number
      ENDPROC
      :
DEFPROCchoose_a_number
      PROCmessage(21):PROCset_window_for_choose
      letter_to_read$=FNdigits:PROCrestore
      IF letter_to_read$ is NOT "Q" THEN PROCsend_choice:PROCread(9)
         ELSE set key$ to "M"
      ENDPROC
      :
DEFPROCsend_choice:PROCwordtx("MAIL READ "+ letter_to_read$, TRUE)
      ENDPROC
      :
DEFPROCread(reply%)
      PROCdisplay_continue_menu(0,"Read")
      PROCwait_for_response(reply%+2,1500):PROCread_chars_from_host
      ENDPROC
      :
DEFPROCread_chars_from_host
      REPEAT
         PROCcheck_last_host_response
         IFresponse$ is the command line prompt ">"
            THEN Set key$ variable to "M":PROCkeypress(response$,FALSE)
            ELSE IF response$ is the "Send, Read or Scan:  " prompt OR
               the "Read or Scan:  " prompt
               THEN PROCkeypress(response$,FALSE):PROCquit_read(TRUE)
               set key$ variable to "M"
            ELSE IF response$ is NOT the "--More--" prompt AND
               response$ is NOT the "Action Required:  " prompt
               THEN PROCunexpected_prompt
            ELSE IF response$ is the "--More--" OR "Action Required:  "
               THEN PROCmore_action
         UNTIL response$ is either "Send, Read or Scan:  " OR ">"
      ENDPROC
      :
DEFPROCunexpected_prompt
      PROCdisplay_continue_menu(1,"Continue")
      REPEAT
         Get a character key$ from the keyboard
         PROCkeypress(key$),FALSE)
         UNTIL key$ is c/r OR "Q"
      PROCmessage(32)
      IF key$ is "Q" THEN PROCquit_read(TRUE)
         ELSE PROCsend_c/r_to_continue
      ENDPROC
      :
DEFPROCmore_action
      Initialise key$:PROCdisplay_more_menu
      REPEAT
         IF keyboard buffer is NOT empty PROCmore_select
         UNTIL key$ is "Q" OR a valid "more option" has been chosen
      PROCmessage(32)
      IF a valid "more option" has been chosen
         THEN PROCt_out(0,750):IF timeout is TRUE THEN PROCno_resp
      ENDPROC
      :
DEFPROCmore_select
      PROCgetkey(TRUE)
      IF key$ is "A" THEN PROCwordtx("A",TRUE):validchoice=TRUE
      IF key$ is "N" THEN PROCwordtx("NE",TRUE):validchoice=TRUE
      IF key$ is "D" THEN validchoice = FNdelete_sure
      IF key$ is c/r PROCtx(c/r):validchoice=TRUE
      IF key$ is "Q" PROCwordtx("QUIT",TRUE):PROCset_host_to_command_level
      ENDPROC
      :
DEFFNdelete_sure
      PROCmessage(25)
      REPEAT
         Wait UNTIL the keyboard buffer is NOT empty
         Get a key from the buffer:PROCkeypress(key$,FALSE)
         UNTIL key$ is "Y" OR "N"
      IF key$ is "Y" PROCwordtx("DEL",TRUE) ELSE PROCdisplay_more_menu
      IF key$ is "Y" return TRUE to PROC that called FUNCTION
      ENDFN
      :
DEFPROCsend_c/r_to_continue
      PROCtx(13):PROCdisplay_continue_menu(0,"Wait")PROCt_out(1,1000)
      IF timeout is TRUE THEN PROCno_resp
      ENDPROC
      :
DEFPROCread_until_host_stops
      REPEAT
          PROCrx:PROCdelay(100)
          UNTIL RS423 input buffer is empty
      ENDPROC
      :
DEFPROCquit_read(scrc%)
      PROCmessage(34)
      REPEAT
          IF the variable srsc% is TRUE THEN PROCtxbreakquit ELSE PROCtxbreak
      PROCcheck_last_host_response
      UNTIL response$ is the command line prompt ">"
      ENDPROC
REMark The scan routines
      :
DEFPROCscan_mode
      PROCdisplay_scan_menu:Reset key$ variable
      REPEAT
          Wait until keyboard buffer is not empty
          PROCgetkey(FALSE)
          IF key$ is B THEN PROCwordtx("MAIL SCAN",FALSE):PROCascert
          IF key$ is S THEN PROCwordtx("MAIL QS",FALSE):PROCascert
      UNTIL key$ is "Q" REM or y%=79???
      IF y%<>79 THEN y%=0:PROCdisplay_mail_menu
      ENDPROC
      :
DEFPROCascert
      PROCtx(13):PROCdisplay_continue_menu(0,"Scan")
      PROCwait_for_response(3,750):PROCverify
      ENDPROC
      :
```

```
DEFPROCverify
    REPEAT
      PROCcheck_last_host_response
      IF response$<>"Send, Read or Scan:  " AND response$<>"Read or Scan:
      "AND response$<>">" PROCunexpected_prompt
      ELSE IF response$=">" set key$ to "Q"
      ELSE set key$ to "Q":PROCquit_read(TRUE)
      UNTIL key$ is "Q" OR there is no response from the HOST
    IF there is no response from the HOST THEN PROCno_resp
    ENDPROC
    :
REMark The small window menus and messages
DEFPROCdisplay_mail_menu (figure 4.20)
    PROCkeypress("Mail menu displayed",TRUE)
    ENDPROC
    :
DEFPROCdisplay_scan_menu (figure 4.21)
    PROCkeypress("Scan menu displayed",FALSE)
    ENDPROC
    :
DEFPROCdisplay_continue_menu(wait%,mode$): Use mode$ string as header
    IF wait%=1 display menu as in figure 4.29 (using Q and RETURN)
       ELSE Just print "Please wait .." as in figure 4.22
    IF wait%= 1 THEN PROCkeypress(mode$+" Press Q to quit to main
    menu",TRUE) ELSE PROCkeypress(mode$+" Please wait for a
    response....",TRUE)
    ENDPROC
    :
DEFPROCdisplay_read_menu (figure 4.23)
    PROCkeypress("Read menu displayed",TRUE)
    ENDPROC
    :
DEFPROCdisplay_more_menu (figure 4.30)
    PROCkeypress("More menu displayed",FALSE)
    ENDPROC
    :
DEFPROCm21 (figure 4.31)
    PROCkeypress("Choose a letter to read" ,FALSE):ENDPROC

OVERLAY :2.0.ENV
DEFPROCletter_env
    Initialise certain parameters and arrays
    PROCdisplay(1):PROCpress_bar:PROCkeypress("Sample header",FALSE)
    Initialise certain parameters and arrays
    PROCdisplay(2)
    PROCinput(6,1,CHR$(129)+"LETTER ENVELOPE"):PROCkeypress
    lope start",FALSE)
    IF exit% key$="Q"
    ENDPROC
    :
DEFPROCinput(fields%,start%,head$)
    Initialise cursor keys
    IF start%<fields% THEN PROCkey_entry ELSE PROCpress_bar
    ENDPROC
    :
DEFPROCkey_entry
    Initialise all variables to be used
    REPEAT
      valid_character%=FALSE
      Place cursor in correct position in current field
      REPEAT
        Get key$ from keyboard when one is pressed
        PROCsend_to_keypress
        IF key$ is 'f0' THEN PROChelpnow
        UNTIL key$<>'f0'
      IF key$ is "f1" THEN valid_character%=TRUE:exit%=TRUE
      IF key$ is "c/r" OR "down arrow" THEN PROCcheck_entry
      IF key$ is "delete" THEN PROCpossible_delete
      IF no. of chars is < permitted field length AND key$ > " "
      AND key$ <="~"
        THEN increase the character count:valid_character%=TRUE
        PROCcharacter
      IF no. of chars is < permitted field length AND key$ = " "
      AND this is the title field
        THEN increase the character count:valid_character%=TRUE
      IF key$ is the up arrow key THEN PROCup_arrow
    IF NOT valid_character% PROCerror_message ELSE PROCclear_error_message
    UNTIL end% is TRUE OR exit% is TRUE
    Reset some variables
    ENDPROC
    :
DEFPROCpossible_delete
    IF no. of chars in field is > 0 AND <= length of field
      THEN valid_character%=TRUE
      Delete character from current field and remove from screen.
    ENDPROC
    :
DEFPROCcharacter
    Add new character to the field and display on the screen.
    IF no. of chars = permitted field length THEN PROCaccept_field
    ENDPROC
    :
DEFPROCaccept_field
    Save contents of current field for transmission later
    IF field%<>title field THEN increment field%
    PROCnext_field
    ENDPROC
    :
DEFPROCnext_field
    Set cursor at end of characters in next field
    PROCkeypress("Now at field "+ field number,FALSE)
    ENDPROC
    :
DEFPROCup_arrow
    IF the field% is NOT the top field
      THEN valid_character%=TRUE:PROCcheck_empty
    ENDPROC
    :
DEFPROCcheck_empty
    REPEAT
      Go up fields and test for a blank field
      UNTIL a NON blank field OR the top field is reached
    ENDPROC
    :
DEFPROCcheck_entry
    IF the number of characters in field is 0
      THEN PROCmove_field ELSE PROCcheck_and_move
    ENDPROC
```

```
   :
DEFPROCmove_field
      IF cursor is NOT in last field
         THEN valid_character%=TRUE:set field to title:PROCnext_field
         ELSE IF key$ is down arrow THEN PROCgo_to_start
      ENDPROC
      :
DEFPROCgo_to_start:valid_character%=TRUE:field%=start%:PROCnext_field
      ENDPROC
      :
DEFPROCcheck_and_move
      IF cursor is not in title field
         THEN valid_character%=TRUE:PROCaccept_field
         ELSE IF cursor is in title field AND key$ is down arrow
         THEN PROCstart
         ELSE IF field%=fields% THEN PROCcheck_fields
      ENDPROC
      :
DEFPROCcheck_fields
      IF the title field is NOT empty OR BLANK THEN PROCcheck_names
      ENDPROC
      :
DEFPROCcheck_names
      REPEAT
         IF subject field is NOT empty OR BLANK THEN end%=TRUE
            ELSE increment loop
      UNTIL end%=TRUE OR all subject fields have been examined
      IF end%=TRUE THEN valid_character%=TRUE
      ENDPROC
      :
DEFPROCsend_to_keypress
      IF key$ is "f0" THEN PROCkeypress("requested help",FALSE)
      IF key$ is "f1" THEN PROCkeypress("abort envelope",FALSE)
      IF key$ is "c/r" THEN PROCkeypress("Return",FALSE)
      IF key$ is delete THEN PROCkeypress("delete character",FALSE)
      IF key$ is down arrow THEN PROCkeypress("down arrow",FALSE)
      IF key$ is up arrow THEN PROCkeypress("up arrow",FALSE)
      IF key$ is >=" "AND key$ is <="~" THEN PROCkeypress(key$,FALSE)
      ENDPROC
      :
DEFPROCpress_bar:PROCswitch_cursor_off
      REPEAT:Get a key$ from the keyboard:PROCkeypress(key$,FALSE)
         UNTIL key$ is Space Bar
      PROCswitch_cursor_on
      ENDPROC
      :
DEFPROCdisplay(scr%)
      IF scr% is 1 THEN display "Sample letter envelope" from disc
                   ELSE display "letter envelope" from disc
      ENDPROC
      :
DEFPROChelp_mess:Display short help message (bottom of fig. 4.36)
      ENDPROC
      :
DEFPROChelpnow
      Remember cursor position:PROCdisplay_f0_help_message (fig. 4.41)
      PROCpress_bar:PROCkeypress("space bar",FALSE)
      Clear f0 help message:PROChelp_mess:Restore cursor position
      ENDPROC

   :
DEFPROCerror_message
      Remember cursor position and delay 1/4 sec so screen does not
      flash:PROCclear_error_message:Display "Error Message"
      IF key$ is c/r AND NOT in title field THEN valid_character%=TRUE
         :error$="To enter press RETURN in topic section"
         ELSE IF key$ is c/r and there is no subject name
         THEN error$="You need a topic AND at least one
                       name":valid_character%=TRUE
      IF key$ is left or right arrow THEN valid_character%=TRUE
         :error$="Left/right movement not allowed" (fig. 4.37)
      IF key$ is delete THEN valid_character%=TRUE
         :error$="No character to delete"
      IF key$ is down arrow THEN valid_character%=TRUE
         :error$="Move cursor to start of line first"
      IF key$ is up arrow THEN valid_character%=TRUE
         :error$="You must move down from this field"
      IF character is <=" " AND valid_character%=FALSE OR CHR$(x%)>"~"
         AND valid_character%=FALSE THEN error$="Character is not allowed"
      Print error$:PROCkeypress(error$,FALSE):Restore cursor postion
      ENDPROC

OVERLAY  :2.0.SEND
DEFPROCenvelope
      PROCkeypress("Start send envelope",FALSE)
      PROCmail_tx:PROCinitial_wait:PROCmessage(32):PROCcheck_input
      IF LENstar$>7 AND INSTR(star$,"ext") THEN PROCwrite
         ELSE PROCkeypress("No name selected",TRUE)
         PROCwordtx(".quit",TRUE):PROCset_host_to_command_level
      ENDPROC
      :
DEFPROCmail_tx
      PROCwordtx("MAIL ",FALSE)
      Transmit all the names and title entered in the envelope phase
      ENDPROC
      :
DEFPROCinitial_wait
      REPEAT
         PROCwait_for_response(0,750)
         IF keyboard buffer is empty PROCtx(13)
         UNTIL keyboard buffer in NOT empty
      ENDPROC
      :
DEFPROCcheck_input
      REPEAT
         PROCmessage(32):initialise variables
         PROCwait_for_response(3,750):PROCmessage(32):PROCstar
         stringck%=FNstringck
         IF LENstar$>12 AND stringck% PROCrespecify
         UNTIL LENstar$<13 OR NOTstringck%
      ENDPROC
      :
DEFFNstringck
      IF the variable star$ contains the text string "Send" OR "Read"
      OR "Scan" THEN return a value TRUE ELSE return a value FALSE
      ENDFN
      :
DEFPROCstar
      REPEAT:REPEAT
```

```
            PROCrx_char(FALSE)
            IF character received is NOT null THEN star$=star$+CHR$(chr$)
            UNTIL RS423 input buffer is empty
            PROCdelay(9)
          UNTIL RS423 input buffer is empty
    ENDPROC
    :
DEFPROCrespecify
    PROCmessage(30):PROCkeypress("Respecify",TRUE)
    REPEAT
       Wait until keyboard buffer is NOT empty
       Get the key$ pressed:PROCkeypress(key$)
       UNTIL key$ is "Y" OR "N"
    IF key$ is "N" PROCtx(c/r) ELSE PROCnew_name
    ENDPROC
    :
DEFPROCnew_name
    PROCmessage(31):PROCremember:PROCsetup_message_window
    Position cursor:namer$=FNletters(10):PROCkeypress(namer$,FALSE)
    PROCrestore:IF namer$ is " " PROCtx(c/r) ELSE PROCwordtx(namer$,TRUE)
    ENDPROC
    :
DEFPROCwrite
    PROCdisplay_send_menu
    REPEAT
       PROCrx:IF keyboard buffer is NOT empty THEN PROCs_test
       UNTIL key$ is "f0" OR "f4"
       IF key$ is "f0" THEN set key$ to "Q" ELSE PROCreset_to_>_level
    ENDPROC
    :
DEFPROCreset_to_>_level
    Set star$ variable to null
    PROCt_out(8,750):PROCstar:PROCset_host_to_command_level
    ENDPROC
    :
DEFPROCs_test
    Get a key from the keyboard
    IF key$ is "f0" PROCdiskey("f0"):PROCkeypress(".QUIT",TRUE)
       PROCwordtx(c/r +".QUIT",TRUE)
    IF key$ is "f2" PROCdiskey("f2"): PROCkeypress(".D",FALSE)
       PROCwordtx(c/r +".DISPLAY",TRUE)
    IF key$ is "f4" PROCdiskey("f4"): PROCkeypress(".SEND",TRUE)
       PROCwordtx(c/r +".SEND",TRUE)
    IF key$ is "." PROCtx(32)
    IF key$ is >=ASCII 32 AND < ASCII (127)
       PROCkeypress(CHR$(y$),FALSE):PROCtx(y$)
    IF key$ is backspace THEN PROCkeypress("delete",FALSE):PROCtx(key$)
    IF key$ is c/r THEN PROCkeypress("Return",FALSE):PROCtx(c/r)
    ENDPROC
    :
DEFPROCdisplay_send_menu (figure 4.42)
    PROCkeypress("Send menu",FALSE)
    ENDPROC
    :
DEFPROCm30:Displays text for message 30. (not shown)
    PROCkeypress("Name does not exist,TRUE):ENDPROC
    :
DEFPROCm31:Displays text to message 31 (not shown)
    PROCkeypress("Correct a name",FALSE):ENDPROC
```

<u>Overlay :2.O.INIT</u>

```
DEFPROCbegin:PROCinit:PROCcompile:ENDPROC
    :
DEFPROCinit
    This initialises some string variables, function keys, arrays
    and memory space required to allow the recording the interaction
    of the system with the user on disc. The information used to set
    the menu levels are read into certain variables.
    ENDPROC
    :
DEFPROCcompile
    Thiscompiles a machine code routine and stores it in computer
    memory to be called from the BASIC program (PROCrx_char). It is
    required because the BBC is to slow to accept characters from
    the RS 423 port at 1200 baud. Many of the ASCII character sent
    from the modem are hardware dependent and so this routine
    decodes the input and uses it to ignore certain characters and
    intercept others to carry out certain other functions.
    ENDPROC
```

<u>Overlay :2.O.EVAL</u>

```
DEFPROCevalustart
BEGIN {Algorithm of evaluation introduction}
    IF start_of_experiment is TRUE THEN
         BEGIN{introductory comments}
            display first information screen (figure 4.3)
            when user presses SPACE BAR save time on disc
            display second information screen (figure 4.11)
            when user presses SPACE BAR save time on disc
            start_of_experiment = FALSE
         END
END{Algorithm of evaluation introduction}
ENDPROC
Note: This is similar to DEFPROCevaluation in :2.U.EVALUAT
```

<u>Overlay :2.O.ADASS</u>

```
DEFPROCa_cont:PROCa_message
    REPEAT
       PROCrx:IF keyboard buffer is NOT empty THEN PROCa_test
       UNTIL key$ is either f6 or (CONTROL f8)
    ENDPROC
    :
DEFPROCa_test
    Get the key$ from the keyboard input buffer
    IF key$ is c/r THEN PROCkeypress(c/r,TRUE)
       ELSE PROCkeypress(key$, FALSE)
    IF FNnot_fn_keys AND key$ is NOT f6 THEN PROCtx(key$)
       ELSE IF the key$ is f6 THEN transmit XON with a c/r
    IF the key pressed was f6(O - to go offline) PROCconfirm_logoff
    ENDPROC
    :
DEFPROCa_message
    Display "Press f6 to disconnect " message in white text on a red
    background on last line of screen and restore cursor position
    ENDPROC
```

APPENDIX 5

EXTRA FACILITIES NOT EVALUATED

1. Compose function

This function allows the user to compose a letter offline, based on the envelope and content model, for automatic transmission. For consistency this interface was made as similar as possible to the online interactive send task. The difference was that in the offline case the user could create text only (for example for tranmission in CHAT mode). From the main menu, the user selects f3 and the compose mode is displayed and the user selects either both envelope and text or just text. If the envelope is selected a form must be filled in with the required information. The form model was chosen as it linked easily to the system model and also to the users model of an envelope. A sample envelope is displayed, as in the send task, which indicates to the user the style and constraints of the envelope. The user proceeds by pressing SPACE BAR and an empty envelope is displayed. Characters can then be entered, these must either be numeric and/or alphanumeric ('space' is only allowed in the title field). Help and error messages are displayed in the bottom part of the screen as in the send task. The user then saves the information on disc by selecting a filename and then the user is moved into text mode.

The only edit function provided was delete a character. It was reasoned that it would not be necessary to provide complex editor functions because the naive user would not be familiar with word processing. The expert user, familiar with word processing, would be expected to use a proprietary editor or use the line editor within the mailbox system.

2. PRESTEL function

This function loads and runs proprietary software to access PRESTEL therefore development of such software was not required.

3. System Characteristics facility

The main offline menu provides an indication as to the initial state of the system which can be configured as the most used state. The user may change any of these parameters using the four options. Some of these facilities are 'toggle' functions such as the type of modem used, manual or automatic access and the system communication rate (V21 or V23). When using the latter function the modem rate must be switched manually this would not be required if a software controllable modem was used. Another function allows the user to change the system to access. This was deemed to be too complicated for the naive user and more of use to the advanced user.

4. Auto Timer (or Batch) function

This option was not fully implemented but is described here for completeness. This function allows the user to set up the local terminal to enter the mailbox at a specified time (usually offpeak) to upload and download mail to be read locally later. On entry to this function from the main menu, the clock menu is displayed. The user may set the timer and the present time. The user is allowed to select the mail operations which the intelligent front end processor is to perform while in batch mode. The timer is then selected in the timer menu and batch operation begins.

5 . Conversation model

Time precluded the implementation of a direct point to point conversation mode between two microcomputers. The following is a brief description of the GOLD CHAT facility which allows direct communications through the GOLD central computer. Obviously both user must be online to communicate.

When CHAT is selected the appropriate command is tranmitted to request the display of the potential contacts in the user group. The user is asked to enter the person to be contacted. The relevant GOLD command is sent and the software interprets the response to detect whether the contact either does not exist, is not present or is present and is willing to CHAT. If CHAT is allowed a greeting message is sent to the contact and the user sees the CHAT menu. The user can enter a further message and then press the key f3 to indicate to the other user that he has finished and wishes to have a reply. The conversation is terminated by pressing f9 which sends the termination message and code to GOLD to end the link.

APPENDIX 6

STATISTICAL CALCULATIONS AND EXPLANATION

1) Analysis of the Pilot Experiment

The implications of the pilot analysis results are dealt with in chapter 5. A summary of the data and the differences within each subject by groups is shown below in table A6.1.

Subject	Period 1	Period 2	difference	mean
Group I				
1	241.91	69.99	171.92	155.95
3	142.34	70.39	71.55	106.57
Group II				
2	69.99	417.72	347.73	243.86
4	69.09	517.88	448.79	293.49

Table A6.1 Pilot Results (Times in seconds)

a) Testing for an interaction

Before analysing the data to estimate the treatment and period effects, the data must be tested for any potential interaction between the period and treatment. This will indicate that the treatment effect is not the same for both periods and therefore that the data from both periods cannot be used. The response of each subject to each treatment is assumed to be composed of a fixed amount which depends upon the treatment (T) and is the same for each subject and for both periods, apart from the random error. Let the contribution to the response of all factors, apart from the treatment, for periods 1 and 2 be e_1 and e_2 respectively and the fixed treatment effects be T_a and T_b. Then the outcome of the trial will be,

Group I subject

$$y_1 = T_a + e_1$$
$$y_2 = T_b + e_2$$

Group II subject

$$y_1 = T_b + e_1$$
$$y_2 = T_a + e_2$$

where y_1, y_2, e_1, e_2 will vary with each subject. Thus the treatment effect can be determined from the differences in each group.

$$d_I = y_1 - y_2 = T_a - T_b + (e_1 - e_2)$$

$$d_{II} = y_2 - y_1 = T_a - T_b - (e_1 - e_2)$$

where $(e_1 - e_2)$ should be a non-zero mean value for a consistent time trend over all subjects. If there is no interaction then the mean response for a subject in either group will be,

$$m = \frac{1}{2}(T_a + T_b) + \frac{1}{2}(e_1 + e_2)$$

If there is no interaction between period and treatment then the first term will be a constant. The second term will vary between subjects but if the subjects are allocated randomly to each group the mean responses over the two groups should not differ appreciably. If there is an interaction then the treatment responses in each period differ will be T_{a1}, T_{a2}, T_{b1}, T_{b2} and the mean response will be,

$$m_I = \frac{1}{2}(e_1 + e_2) + \frac{1}{2}(T_{a1} + T_{b2}) \quad \text{for group I subjects}$$

and $$m_{II} = \frac{1}{2}(e_1 + e_2) + \frac{1}{2}(T_{a2} + T_{b1}) \quad \text{for group II subjects}$$

The condition for no interaction is $m_I = m_{II}$,

$$\text{i.e. } T_{a1} - T_{b1} = T_{a2} - T_{b2}$$

Therefore a test for the equality of the mean values m_I and m_{II} is a way to detect an interaction if it is present, although it could be insensitive because it is based on the between subject variation.

	Group I		Group II
subject 1	$\hat{y}_1 = 155.95$	subject 2	$\hat{y}_2 = 243.86$
subject 3	$\hat{y}_3 = 106.57$	subject 4	$\hat{y}_4 = 293.49$
mean	$\hat{\mu}_1 = 131.25$		$\hat{\mu}_2 = 268.67$
s.d	34.92		35.09
s.e.	24.69		24.81

Table A6.2 Means of the two responses

The mean difference in the means is $D = \hat{u}_1 - \hat{u}_2 = 137.42$

$$se(D) = \sqrt{\left[s_m^2\left(\frac{1}{2} + \frac{1}{2}\right)\right]}$$

Where $s_m^2 = \dfrac{\text{pooled (i.e. averaged) sum of the squares for the 2 groups}}{1 + 1}$

This estimates the variation of the average of the pair of observations on the subject. Therefore s.e(D) = $2s^2/n$ = 35

and the t-ratio is $\frac{D}{s.e.(D)} = \frac{134.42}{35} = 3.92$

This is significant at the 10% level (2.92) which indicates that there is an interaction and the results from both periods cannot be used. Figure 5.2 shows the results in a diagrammatic form.

Some of the possible reasons for the interaction could be,

1) the subjects response in period 2 might contain a residual effect from period 1. If the residual effects of each interface are different then there will be an interaction. For instance a psychological state, such as frustration, caused by using one of the interfaces may persist until the second period and cause the subject to take more time to complete the test.

2) the treatment effects may vary according to the response level.

It was not possible to remove the interaction by transforming the scale of measurement used by taking the natural logarithm or square root of the data.

Testing results with an interaction

Only the period 1 results can be compared.

subjects	period 1		period 2
1	241.91	3	69.99
2	142.34	4	69.09
mean	192.12		69.54
(I - II)		122.58	

An estimate of between subject variance in period 1 is given by the sum of the squares (ss) which is 2500. A t-test on these results gives a value of 2.45. This is not significant at the 10% level (2.92) so there is no evidence of a difference in the times taken to use either interface in the pilot experiment.

2) Sensitivity Calculations

The results of the pilot assessment can be used to make a retrospective judgement as to its sensitivity. The sensitivity is,

$$\frac{\text{true practically significant difference (D)}}{\text{standard error ofthe estimated difference (s.e.)}}$$

It is recommended (Hills and Armitage 1979) that a trial should have an 80% power to detect a true difference D and from statistical tables this means the ratio D/s.e. is 2.8.

If the actual experiment is based on the pilot model and assuming the worst case, where a treatment X period interaction cannot be removed by transformation, then an estimate can be obtained of the power of the experiment. This will produce the number of subjects which will be required to have a probability of 0.8 of detecting a true time difference of D seconds between the time to use the two treatments, using the results of the first period only. If it is assumed that 60 seconds is a meaningfully large true difference then the sensitivity is calculated as follows.

1) A rough estimate of the between subjects variation in period 1 is calculated. From the pilot study this is given by the pooled sum of the squares (s^2) which is 2475.
2) Then the variation of the difference between the means for n subjects on A and n subjects on B is estimated by s.e. = $2s^2/n$.
3) To achieve a 0.8 chance of detecting a true difference of 60 seconds 2n subjects are required.

Therefore,

$$\frac{D}{s.e.} = \frac{D}{\frac{\sqrt{2 \times 2475}}{\sqrt{n}}} = 2.80$$

Hence n(no. of subjects) $\cong$ 22.

So a minimum of 22 subjects is required to be able to detect a pratically significant difference of 60 seconds. The within subject variation has a mean square of 2416 which is similar to the between subjects mean square 2475. This indicates that if there is no interaction (or if it can be removed by transformation) and both period results can be used then only a slightly smaller number of subjects will be required than in the case where there is an interaction between period and treatment.

3) Calculation of the LSD for the comparison of tasks

To calculate the least significant difference, an estimate of the standard error of the difference (s.e.d.) between the two systems is required. This is obtained from the variance of the difference between the two treatments, which is given by,

$$\text{var}(\hat{x}_1 - \hat{x}_2) = \sigma^2\left(\frac{1}{n} + \frac{1}{n}\right)$$

Therefore an estimate of the s.e.d is given by

$$s\sqrt{\frac{2}{n}}$$

where s is the square root of the residual mean square obtained from the ANOVA table. Combining variances from the corresponding strata gives an estimated variance of,

$$s^2 = \frac{2 * 4161 * 5304}{3} = 4542$$

and therefore the standard error of the difference is,

$$\text{s.e.d.} = \sqrt{4542\left(\frac{1}{12} + \frac{1}{12}\right)} = \sqrt{757} = 27.5$$

(The s.e.d. for comparing two systems for a task is larger than for comparing tasks within a system, because the first comparison includes task X period and period X systems as well as task X systems terms).

The t-test statistic is defined as,

$$t_0 = \frac{\hat{x}_1 - \hat{x}_2}{s/(1/n + 1/n)}$$

Now a two-tailed t-test at 1% significance gives a t-statistic of $t_{40}(.995) = 2.704$ from t-distribution tables. Therefore the least significant difference for a 1% t-test statistic is,

$$\text{L.S.D.} = \text{s.e.d} \times t_{40}(.995)$$
$$= 27.5 \times 2.704 = 74.4$$

This can also be done for the experienced subjects.

$$1\%\ \text{L.S.D.} = \text{s.e.d} \times t_{40}(.995)$$
$$= 26.33 \times 2.704 = 71.2$$
$$5\%\ \text{L.S.D.} = 26.33 \times 2.704 = 53.2$$

4) Mann-Whitney Tests on Subjective Results

a) Naive subjects.

Using the raw scores (see tables 6.5 and 6.6 in chapter 6).

Subject	Period I	Period II	Sum I+II	Rank	Diff I-II	Rank	Cross	Rank
1	17	20	37	12	-3	5.5	-3	10
2	8	20	28	4.5	-12	1	-12	1
3	9	18	27	2.5	-9	2	-9	2
4	11	18	29	6.5	-7	3	-7	3
5	11	16	27	2.5	-5	4	-5	6
6	14	17	31	8	-3	5.5	-3	10
Sum of Ranks				36		21		32
Significance Level			Not sig.			1%	Not sig.	
7	20	15	35	11	5	10.5	-5	6
8	16	16	32	9.5	0	7	0	12
9	14	9	23	1	5	10.5	-5	6
10	18	14	32	9.5	4	9	-4	8
11	16	13	29	6.5	3	8	-3	10
12	17	11	28	4.5	6	12	-6	4

Table A6.3 Wilcoxon Rank scores on total system scores for ease of use results for naive subjects

The Mann-Whitney U-test is equivalent to the Wilcoxon signed rank test and is described in chapter 5. In the Mann-Whitney test the scores in both groups are ranked and the ranks of one of the groups are summed. The medians of the groups can now be tested to see if they differ significantly. If the size of group I (i.e. those who had experienced the interfaces in the order (ab)) is n_a and n_b is the size of group II (ba). Then the two alternative hypotheses are,

H_0 : median of sum of group 1 = median of sum of group II.

H_1 : median of sum of group 1 $\neq$ median of sum of group II.

If the group sum is higher or lower than a critical value then the alternative hypothesis is said to be true otherwise the null hypothesis is true. The upper and lower critical values for the rank sum of the Wilcoxon test can be found from statistical tables and are given for $n_a = n_b$ in table 6.8 below.

significance level (%)	critical values
10	28,50
5	26,52
2	24,54
1	23,55

Table A6.4 Wilcoxon Rank Sum Values

From the table above the total of the sum terms in the smaller period is 32 which is not significant so the results from both periods can be used in the remainder of the analysis. The total sum of the difference terms is 21 which is significant at the 1% level in the Wilcoxon rank sum test and since the results for b are less than for a in 11 out the 12 results then the conclusion is that the intelligent interface b is found easier to use than the existing interface a. If a similar test is conducted on the individual scores for each task it is found that there is evidence that b is preferred to a (in terms of ease of use) for 'logon', 'compose and envelope' and 'read' (1%), and 'send' (2%).

Using the Mann-Whitney rank sum test on the sum of the total system scores for user satisfaction (table 6.7) at the 10% level we have to reject the null hypothesis of no difference between the medians because the smaller sum is 28 or less. The conclusion is that the median for group I is less than that from group II and that there is a period system interaction. The scores from both periods cannot be used and the analysis of the individual task scores for user satisfaction must be conducted on the first period scores only.

The conclusion from the tests on the individual scores are that there is evidence that b is preferred to a (in terms of user satisfaction for 'read' (at 5%), 'compose an envelope' (1%). There is only very weak evidence that b is prefered to a on the 'send' task (10%) and no evidence that 'logon' is more satisfying in either interface.

b) Experienced Subjects

The Mann-Whitney rank test did not show any evidence of period X system interaction so both periods results could be used. The test on the total scores did not show any evidence of a difference in the ease of use or satisfaction of the intelligent interface and the dumb terminal. If each task is examined independently then it is found that only the 'read' task was significantly (5%) easier and more satisfying to use (10%) with the dumb terminal than the intelligent interface.

APPENDIX 7

PRE- AND POST-TEST QUESTIONNAIRES

UNIVERSITY OF SOUTHAMPTON

Department of Electronics and Information Engineering

questionnaire Part 1

CONFIDENTIAL

1. Name:...
2. Age: (Please tick one of the following)
   under 20 20-29 30-39
   40-49 50-59 60 or over
3. Sex: Male Female
4. Are you: Deaf
   Hard of hearing
   hearing

If you are HEARING please go to question 10.

5. Do you regularly wear a hearing aid? Yes No

If the answer is NO please go to question 7.

6. If YES do you wear it: Yes No
   All the time
   or only some of the following
   TV/radio
   telephone
   face to face conversation
   in the street
   at meetings

7. Have you had this hearing loss from birth? Yes No

If the answer is YES go to question 10.

8. If NO: at what age did you become deaf/hard of hearing?

9. What was the cause of your hearing loss?

1

P.T.O

10. What type of school did you attend?
    i) ordinary school
    ii) school for the deaf
    iii) other - please state brief details ....

11. At what age did you leave school?

12. Have you taken any further education since leaving school?
    YES NO

13. Please give brief details of any qualifications (O levels, HND, Degree etc.), past, present job of work,

14. Do you use a telephone? YES NO

If the answer is NO go to question 17.

15. If YES has it been modified in any way? YES NO

If the answer is NO go to question 17.

16. If YES give brief details:

17. Do you contact people by:

i) writing letters
ii) sending messages via other people
iii) asking people to use the phone for you
iv) face to face conversation (verbal)
v) face to face conversation (signing)
vi) others (please state)

2

18. Have you heard of any of these systems? YES NO

1. Telecom Gold
2. Prestel
3. Vistel
4. Micronet
5. PSS Mailstream
6. RNID telephone bureau

If you have not heard of any of the above systems go to question 26.

19. If YES have you ever used them? YES NO
If NO please go to question 26.

20. If YES which one(s)? (In order of usage time)

21 How often do you use an electronic mail system?
i) less than twice a month
ii) once a week
iii) once a day
iv) more than once a day

22. How easy did you find it/them to use:

very easy quite difficult
fairly easy difficult

Any other comments.

23. How useful did you find the systems
i) not at all
ii) fairly useful
iii) very useful

24. If you beleived you had a need for a text telephone system, how much would you pay for one.
1. Nil
2. 0 - 200
3. 200 - 400
4. 400 - 600
5. 600 - 800
6. more than 800

? T.O

3

25. How much are your quarterly phone charges (include the use of Prestel, any mailbox systems etc.)
1) less than 30
2) 30 - 50
3) 50 - 70
4) 70 - 100
5) 100 - 150
6) more than 150

26. Have you used a computer before: YES NO
27. Have you used a typewriter before: YES NO
28. You will be using a computer and keyboard during this experiment. How do you feel about using the equipment.
1) very uncomfortable
2) slightly uneasy
3) no strong opinion
4) comfortable
5) confident

Thank you for completing this questionnaire.

4

Appendix 6.B Post-test questionaire.

Introduction

In the past many computer systems have been designed without asking possible users what they think of the system. We believe it is important to find out the opinions of possible future users of computers. The questions below are simply to find how easy or difficult you found the computer programs to use and to understand. It is also intended to discover if you found the programs enjoyable to use and to obtain any comments you have about how they could be improved.

This is not a test and there are no 'correct' answers. All we want you to do is to give us your opinions on the system to help us know how it may be improved in the future. Take as much time as you require over these questions as this is the last part of the experiment.

There are three types of questions, the tick a choice question, like question 2. For example, if you felt uncomfortable using the keyboard tick answer a). There is the rating type question where you write a number down according to how you rate the feature mentioned. Then at the end there are the open questions where you can comment freely on any part of the experiment.

Questionaire

Confidential

1. Name: ....
2. How comfortable did you feel using the keyboard.
   a) uncomfortable
   b) did not mind (no strong opinion either way)
   c) quite comfortable

5

3) In the future it seems possible that using an electronic mailbox will be as common as using a telephone. What is your attitude to the use of an electronic mailbox now?
   Do you think you would find it,
   a) very useful
   b) useful,
   c) don't know how useful (no strong opinion)
   d) of little use,
   e) no use at all.

******

Rating Questions

In the following questions you will be asked to give a rating on how easy or difficult you found it to use and understand the computer program. The higher the number you give the session the easier you found it.

Rating	
1	very difficult
2	difficult
3	not easy or difficult
4	easy
5	very easy

Use a number from 1 to 5 to describe the question. For example if you found that to send a letter in session 1 was very easy but in session 2 was difficult, then answer as follows.

How easy did you find it to send a letter using
a) session 1 ... 5
b) session 2 ... 2

*****

4) How easy did you find it to OPEN the mailbox in,
   a) session 1
   b) session 2

6

5) How easy did you find it to COMPOSE AN ENVELOPE in,
   a) session 1
   b) session 2

6) How easy did you find it to SEND A LETTER in,
   a) session 1
   b) session 2

7) How easy did you find it to READ A LETTER in,
   a) session 1
   b) session 2

In the following questions you will be asked to give a rating which is the closest to how much you enjoyed using the computer program. In all the questions session 1 is the first computer program you used and session 2 is the second program you used.

Rating	
1	dissatisfied
2	not enjoyable
3	no strong opinion
4	enjoyable
5	satisfying

Use a number from 1 to 5 to describe the question. For example if you found that to send a letter in session 1 was enjoyable but in session 2 you had no strong opinion then answer as follows.

How enjoyable did you find it to send a letter using
a) session 1 ... 2
b) session 2 ... 3

*****

8) How enjoyable did you find it to OPEN the mailbox in,
   a) session 1
   b) session 2

9) How enjoyable did you find it to COMPOSE AN ENVELOPE in,
   a) session 1
   b) session 2

10) How enjoyable did you find it to SEND A LETTER in,
   a) session 1
   b) session 2

11) How enjoyable did you find it to READ A LETTER in,
   a) session 1
   b) session 2

*****

The next set of questions are to determine how you think the use of the computer programs may be improved.

*****

12) Were there any times in session 1 that you were confused?
YES NO

13) If Yes say when and why ....
....
....
....
....
....

14) Were there any times in session 2 that you were confused?
YES NO

15) If Yes say when and why ....
....
....
....
....
....

16) Were there any times in session 1 that you felt frustrated?
YES NO

17) If Yes say when and why ....
....
....
....
....
....

18) Were there any times in session 2 that you felt frustrated?
YES NO

19) If Yes say when and why ....
....
....
....
....
....

20) Do you have any major criticisms of session 1?
YES NO

21) If Yes say where and why ....
....
....
....
....
....

22) Do you have any major critcisms of session 2?
YES NO

23) If Yes say where and why ....
....
....
....
....
....

24) How do you think the either interface could be improved?
....
....
....
....
....
....

25) Do you have any other comments about either session that you wish to make? Please feel free to make positive or negative comments.
....
....
....

9

....
....
....

26) Did you use the manual Yes No
If NO go to the finish

27) Did you find the manual to session 1,
a) a hindrance to using the system,
b) not very helpful,
c) no strong opinion,
d) helpful,
e) very useful.

28) Did you find the manual to session 2,
a) a hindrance to using the system,
b) not very helpful,
c) no strong opinion,
d) helpful,
e) very useful.

29) Have you used any other mailbox systems
YES NO
If NO go to the Finish

30) If YES how does using the session which used menus compare with them. Is it,
i) much worse,
ii) slightly worse,
iii) no difference,
iv) slightly better,
v) much better.

31) The Finish

Thank you for taking the time to take part in this experiment. I hope you found it enjoyable and not too tiring.

Keith I. Watson

10

APPENDIX 8

USER MANUALS

USER MANUAL FOR INTERFACE A

Introduction

In this manual there is a brief description of an electronic mailbox. This is followed by the instructions to help you to send, read and scan letters. You can use this manual in three ways. Either you can,

a) read it through from start to finish, or
b) look up the keyword in the back on page 10 and refer to these as you try out the system, or
c) use the flow diagram on page 11 to guide you through the system.

In the manual there are examples to help you and the following symbols are used.

Please sign on	system responses
MAIL	your input
RETURN	Press the RETURN key.
'n'	abbreviation for number.

Note:

** 1) it does not matter whether you enter the information in lower case or UPPER CASE. That is, typing mail RETURN will have the same effect as typing MAIL RETURN.

** 2) If you make a mistake use the DELETE key to rubout the previous character that you typed in.

** 3) if the information on the screen seems to have stopped and you expect more, press the RETURN key which is at the right hand side of the keyboard.

A Description of an Electronic Mailbox

An electronic mailbox is used just like a physical mailbox. To open the mailbox you need a password (like the key to the mailbox). In this version of the interface you will need to know this and so it is given in your task sheet. Once inside the mailbox you can SCAN through the envelopes to find the one you want to read. When you have found it you can then open the letter to READ it. Later you can SEND a letter to someone else.

Opening the Mailbox.

** 1) You must open the mailbox before you send read or scan letters. Once you are sitting at the terminal and are ready to dial the number and open the mailbox,

*** Press the space bar and the number will be dialled for you.**
(N.B. The SPACE BAR is the long key bar situated at the bottom of the keyboard).

** 2) When you see the blank screen with the red message at the bottom of the screen then,

*** Press the RETURN key (at the right of the keyboard) a few times until you receive the system prompt as shown below.**

** 3) Now type in the system call exactly as shown below then press RETURN. You should see the message as follows.

```
Telecom Gold Network: For assistance type
'HELP LOGIN' at the prompt 'PAD>'.
This is a Dial up pad 4 line 12 speed 1200
PAD>CALL 76 (followed by RETURN)
Welcome to Telecom Gold
Please sign on
>
```

** 4) If your ID is BKU222 and your password is the word FRIEND then enter these in the following format, (Note for your ID please see the task sheet.)

Please sign on
>ID BKU222 FRIEND (followed by **RETURN**).

If you press RETURN before the password then the system will ask you to enter your password but will make it invisible for security.

** 5) Wait while your computer talks to the distant computer. When contact has been established you should see the screen as shown below in figure 1. This is called the system level.

```
Telecom Gold Automated Services 18.3A (79)
On at (time) (date)
Last on at (time) (date)
Mail Call (4 unread, 4 read)
(A message may appear here)
>
```

Figure 1 System level prompt.

Entering the Mail System - the MAIL command.

Once the mailbox is OPEN the mail system may be entered by typing **MAIL (RETURN)** at the system level prompt (>). You will then receive the prompt: "Send, Read or Scan:", see figure 2.

Note: To increase speed you may type in commands on the same line as the > prompt. This will have the effect of missing out the next prompt (see page 6). Example **>MAIL SEND (RETURN)**

Sending a Letter (see also example on p5)

Sending a letter by electronic mail is similar to sending a normal paper letter by surface mail. One of the differences is that you can send letters to more than one person very easily. To send a letter follow the instructions below.

** 1) At the system prompt (>)

*** Enter the "MAIL" command** then the system will display the message "Send, Read or Scan:" (figure 2).

```
Telecom Gold Automated Services 18.3A (79)
On at (time) (date)
Last on at (time) (date)
Mail Call (4 unread, 4 read)

>MAIL(followed by pressing RETURN)

Send, Read or Scan:
```

Figure 2 MAIL prompt

** 2) There are now four options to choose from. (Note:- entering QUIT (and RETURN) will return you to the system prompt (>).If you wish to send a letter.

*** Type SEND or S and then press RETURN.**

To send a message it is necessary to indicate:

** a) who you wish to contact. A user may SEND messages of any length to one or more persons by specifying their familiar names or ID numbers.

** b) a subject line for the item,

** c) the text (content) of the message

So you will see the prompts as shown below.

a) To: prompt

Enter here the name or names of the person(s) to receive the message, each separated by a blank space and then press the RETURN key which is situated to the right of the keyboard.

b) Subject: prompt

Enter a subject line, or title, of the message. This is a one-line description of the message which is displayed to the recipient during a READ or SCAN function. The Subject line may be up to 120 characters in length.

Note:i) You need to type in at least one NAME and a SUBJECT to go to the text prompt.

ii) At To: or Subject use QUIT (RETURN) to quit to system level.

c) Text:

Enter the message itself, as you wish it to appear to the recipient(s). You may enter as many lines as you like; You must press RETURN after each line. The delete character key is the delete key to the right hand corner of the keyboard.

Note:i) Use .quit to quit and return to system level if required.

Text Options

** a) To redisplay the text you have typed in then:

*** Press RETURN then press .d followed by another RETURN** while in the text: prompt and your message will be displayed.
After each section has been displayed type RETURN to display the next section until the prompt "More text:" appears.

** b) If you wish to add another name which you may have forgotten to type in at the To: prompt then type RETURN to begin a new line, then .TO FRED (RETURN). This will send the letter to FRED as well.

** c) AFTER YOU HAVE FINISHED entering the text, you can post your letter. To do this,
*** Press RETURN, then enter ".SEND" or ".S" and RETURN on a line by itself.**

You will receive a message telling you this has been done.
NAME - Sent
NAME - Sent
(etc.)
If this does not happen then press RETURN again until you see the system prompt > this means the letter is now finished.

Example

If you wish to send a letter to ALF and BKU200 with title 'hello' go through the following procedure.

a) Enter MAIL at the system prompt >.
b) Enter SEND at the Send, Read or Scan: prompt
c) At the to prompt Press the keys A, then L, then F and then press the SPACE BAR. Next press the keys BKU200 and then press RETURN.
d) At the Subject prompt, enter the word, HELLO, and then press RETURN.
e) At the text prompt type in your message

You should have a screen as shown overpage in figure 3.

5

```
>MAIL
SendReador Scan:SEND (May be abbreviated to S)RETURN
To: ALF BKU200 (followed by the RETURN key)
Subject:HELLO (RETURN)
Text:
Do you both want to play golf tomorrow.
.sendRETURN

ALF - Sent
BKU200 - Sent
To: QUIT
>
```

Figure 3 Completed Letter

Note QUIT will exit you to the system prompt (>) but to leave the MAILBOX you need to use the f6 command (see page 10).

** 6) If you want to send another letter go through the same procedure again. If you wish to finish type in QUIT and press RETURN.

This is the end of the Send routine.

You may expand your use of the Command line mode of mail by putting the names of your recipients and your Send option on the command line. In addition, if the names are entered on the command line, you need not specify "SEND" - the Mail system assumes you are sending if it sees user names following the "MAIL" command.

Furthermore you may put you subject on the command line. It must be the last information on the command line and you may enclose it with quotation marks or precede it with "SUBJECT" (which may be abbreviated as "SU").

6

```
1)
>MAIL SEND followed by RETURN
To:ALF BKU200

2)
>MAIL ALF BKU200
Subject:HELLO

3)
>MAIL ALF BKU200 SU HELLO

Text:
```

Figure 4
Examples of having commands on the same line

Scanning a letter

The SCAN command of the mail system displays a summary of the letter envelopes that you have received. Only the Header lines are printed and additionally items are temporarily assigned numbers. You will receive the prompt of "Read or Scan:"

** 1) If you are starting from the command level, Enter **MAIL (RETURN)**

** 2) At the Send, Read or Scan prompt.
Enter **SCAN (RETURN)**, or **QS (RETURN)** (see figure 5)

```
>MAIL (RETURN)
Send, Read or Scan:SCAN (RETURN)

1. From: A.FRIEND (AAA022) Posted: Fri 21-Jun-85 13:03 Sys81(3)
Subject: How to win friends ...
2. From: A.N.OTHER (AAA002) Posted: Sat 22-Jun-85 21:10 Sys81(3)
Subject: Another message...
3. From: B.NICE (AAA010) Posted: Mon 24-Jun-85 10:32 Sys81(3)
Subject: ... and influence people.

Read or Scan:
```

Figure 5 Scanning mail

7

** 3) Another option that you have is to type QUICK SCAN (or QS) which will display a briefer scan of mail.

** 4) If there are only a few letters these will be displayed in the top part of the screen. The information displayed will be the date the letter was sent, who the letter was from etc. Then you will be returned to the command level. If there are more letters than can be placed on the screen then the list will stop for you to read it. To continue press the RETURN key. To QUIT the list press the f0 key.

Reading the letters

The READ command of the mail system will allow you to read a message. Reading a letter by electronic mail is similar to reading a normal paper letter.

** 1) To read you mail simply enter the Mail system by typing "MAIL" at the System Level (>).

** 2) When you are prompted with "Send,Read or Scan:",
* Enter **"READ" (or "R") and RETURN.** (or READ with one of the options shown below).

You will then be shown all your mail, one item at a time. The oldest entries will appear first except for those items sent express.

***** If you wish to read only some of your letters then you must enter a qualifier after pressing READ but before pressing RETURN. (see NEXT PAGE). NOTE - you may wish to scan your mail first to find out where you letter is.**

The Header line of an entry is printed first followed by who sent the letter, when it was sent, and its subject of the letter.

You will then see one of the following prompts:
--More-- (after the Header and before the Text is printed)
Action Required: (After the text has been printed).
(as shown in the example overleaf)

8

Example
>Mail
Send, Read or Scan: **READ 4** (may be abbreviated as "R")

To: A.N. Other (BKU200)
From: K. Watson (BKU033)
Subject: Golf
--More--**NEXT (RETURN)**
To: A.N. Other (BKU200)
From: K. Watson (BKU033)
Subject: Hello
--More--**(RETURN)**
Do you both want to play golf tomorrow.
Action Required: **QUIT (RETURN)**
>

Figure 6 Example of --more-- prompts

** 3) As mentioned earlier you may now select an option to read by typing READ with one of the following qualifiers followed by RETURN.

EXPRESS - read the EXPRESS (i.e. important) letters,
UNREAD - read all the letters that you have not read yet, i.e. your new (UNREAD) letters,
('n') - CHOOSE a letter number ('n') to read, after selecting this option you will be asked to specify the number of the letter in your list of letters. This you may have obtained when you scanned through the letters.

Example

If there were 6 letters of the following types, 2 express, 2 unread and 2 read letters then pressing READ would cause all the letters to be read starting at the express letters, then the unread letters would be displayed and then the read letters.

If at the "Send, Read or Scan:" prompt READ 4- was entered then only the letters would be read starting with the 4th letter. You would use this option if you have seen a letter you wish to read from using the scan option.

9

The --More-- and Action Required: prompts

** 4) After the first part of the first letter has been displayed you will be presented with the --More-- prompt.

This prompt occurs for two reasons.

a) First it appears after the header lines and allows you to make a decision as to whether or not you want to see the Text of the message.

b) Secondly, the mail system assumes you are using a screen terminal and pauses every 19 lines to give you time to read the text. The same prompt "--More--" is offered at each pause.

The Action Required prompt appears at the end of the text.

Options at '--More--' and 'Action Required:'

RETURN key	Display of continue displaying Text of message.
NEXT	Skip the Text and go onto the NEXT message.
Q or QUIT	Quit and return to > command level.
A or AGAIN	redisplays the text of the message.
R	do not select this option (if accidentally selected use '.quit' to escape from it).

On selecting one of these options wait for a few seconds while the next part of the letter of the letter comes onto the screen. This will either be followed by a "--More--" or "Action Required:" prompt or the computer will say "End of Mail" and will return to the "Send, Read or Scan:" prompt.

OFFLINE

The only other option at the the system level is the offline function this will disconnect you from the distant computer and lock your mailbox for you automatically. Use this function to complete the session (figure 7).

To sign off, simply press the f6 key which is one of the RED function keys at the top of the keyboard.

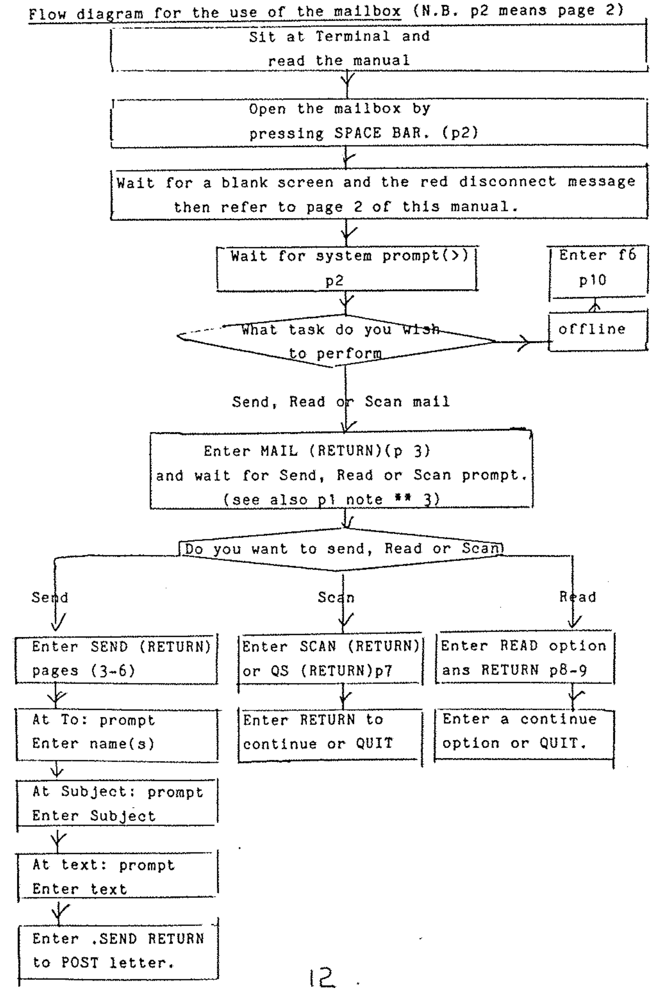
10

SUMMARY

To open your mailbox press SPACE BAR at the beginning of the program and then typing in the ID and the PASSWORD. When you are inside your mailbox you can send letters by selecting the MAIL and SEND option you are then asked to compose an envelope and then the text. To read you use the MAIL READ option and to scan you use the MAIL SCAN option. When the session is finished you select the OFF option at the system level.

Index.

Flow diagram for the use of the mailbox (N.B. p2 means page 2)



USER MANUAL FOR INTERFACE B

Introduction

The purpose of this manual is to give you the information you require to use an electronic mailbox. First there will be a brief description of what an electronic mailbox is and what it does. This is followed by instructions to send, read and scan letters.

You can use this manual in three ways. Either you can,

a) read it through from start to finish,

b) look up the keyword in the back on page 11 and refer to these as you try out the system, or

c) use the flow diagram on page 12 to guide you through the system.

A Description of an Electronic Mailbox

An electronic mailbox is used just like a physical mailbox. To open the mailbox you need a password (like the key to the mailbox). In this version of the interface you will not need to know this as the computer will open the mailbox for you. Once inside the mailbox you can SCAN through the envelopes to find the letter you want to read. When you have found it you can then open the letter to READ it. This is similar to looking through your inbox at work. You can also SEND letters to other people.

Opening the Mailbox.

You must open the mailbox before you can send letters. When you are sitting at the terminal,

*** Press the space bar and the computer will dial the number for you.**

(N.B. The SPACE BAR is the long key bar situated at the bottom of the keyboard).

Wait while your computer talks to the distant computer. When contact has been established you should see the screen as shown in figure 1. You are now ready to use your mailbox.


```
1    From: TCD100002   Mon  25-Feb-85  1
5:18   TTNS
2    From: D.NAYLOR    Tue  26-Feb-85  2
2:26   Reply to:  Questionaire Hardcopy
3    From: B.JUDSON    Wed  27-Feb-85  7
:33    REPLY TO YOUR QUESTIONNAIRE
4    From: N.WILDING   Thu  28-Feb-85  2
2:04   REPLY TO QUESTIONNAIRE
5    From: D.MYERS     Thu  28-Feb-85  2
2:53   questionnaire
6    From: I.DONKIN    Sun  3-Mar-85  22
:01    Reply to:  Reply to:  Reoly to:
QUESTIONAIRE
7    From: TCD100002   Mon  4-Mar-85  17
:19    Reply to:  Reply to:  TTNS
>
  Menu:Beginners
  M-Mail - i.e.Send, Read or Scan
  O-offhook - disconnects the phone
  A-Advanced user menu
  C-chat - to another person
  Online :Printer OFF:mins used- 1
```

Figure 1 Beginners Menu

```
On
>Password:
TELECOM GOLD Automated Office Services I
B.4M(79)
On At 13:57 08/06/85 BST
Last On At 13:46 08/06/85 BST

Mail call (2 Read, 2 Unread, 1 Unread ex
press, Total 5)

>
>
>

>
  Menu.Mail
  Q-Quit and return to main menu
  S-send a letter to another user
  R-read any mail you may have
  P-scan any mail you may have
  Online :Printer OFF:mins used- 1..
```

Figure 2 Mail Menu

```
        SAMPLE LETTER ENVELOPE

First person    :BKU073    :
Second person   :MICHAEL   :
Third person    :          :
Fourth person   :          :
Fifth person    :          :
Letter title    :R.S.V.P. to your party
invite on Saturday

               :

This section helps you to complete
an envelope like the example shown.

You will be requested to type in the
name(s) of the people you want to
contact and the title of your letter.
A maximum of five names is allowed
: denotes the maximum name length.

To finish you must enter at least one
name and press return after the title
```

Figure 3 Help message

```
          LETTER ENVELOPE

Type in the names or numbers of the
people who are to receive your letter.
Then type in the title of the letter.

First person    :          :
Second person   :          :
Third person    :          :
Fourth person   :          :
Fifth person    :          :
Letter title    :


                :
****************************************
Press f0 for a brief help summary.
Press f1 to leave the letter envelope
```

Figure 4 Letter envelope

<u>Sending a Letter</u> (see also example on page 4)

Sending a letter by electronic mail is similar to sending a normal paper letter by surface mail. The difference is that you can send the same letters to more than one person very easily. You type in the names of people you wish to contact onto an envelope and type some words inside the letter as the content. To send a letter you must proceed using the following instructions.

** 1) To select the send a letter option,

* **Press the button M on the BEGINNERS menu, see figure 1 on the facing page.**

This will display the mail menu as in figure 2 on opposite page.

** 2) In the bottom part of the screen a menu will appear. You see that there are now four options to choose from. If you wish to send a letter,

* **Press the S button for the <u>S</u>END option.**

You will then see the information as shown in figure 3.

** 3) Read this as it will help to explain to you how to compose an envelope.

** 4) To move onto the actual envelope,

* **Press the SPACE BAR and you will then see the screen in figure 4.**

This is to help you compose the letter envelope.

** 5) You need to type in at least one <u>NAME</u> and a <u>TITLE</u>.

a) to enter a name just type in the characters on the keyboard and then press the RETURN key which is situated to the right of the keyboard.

b) after entering the name or names of the people to whom you wish to send the letter to press the down arrow (↓) which is situated above the RETURN key.

c) If you wish to correct the names or the title use the up arrow (↑) to move back up the list and the delete key (which is situated at the bottom right hand corner of the keyboard) to correct the characters.

d) When you have finished the envelope you need to have the cursor in the title area and then press the RETURN button (to the right of the keyboard).

e) **Note IF YOU ARE IN TROUBLE either LOOK AT THIS MANUAL or PRESS f0 for a brief help summary.**

Example

If you wish to send a letter to ALF and BKU200 with title 'hello' go through the following procedure.
a) Press M in the beginners menu,
b) Press S in the mail menu,
c) Read the help message if you need it and then press SPACE BAR
d) Press the keys A, then L, then F and then press RETURN. This will enter the word ALF into the first name section.
e) Then press the keys BKU200 and then return.
f) Press down arrow this will place the cursor (the flashing line on your screen) next to the word title.
g) Enter the word, HELLO, and then press RETURN.
You should have a screen as shown below in figure 5.

If you make a mistake various help messages will appear at the bottom of the screen. For example, if you try to delete a character which is not there, i.e. by pressing delete at the start of the line then the message,

There is no character to delete here..

will appear in the bottom part of the screen.

** 6) The envelope will then be sent to the electronic mail system and after a few seconds you will see the screen below, figure 6.

** 7) This menu means that you can now type in the content of your letter. Use the keyboard to enter the message you required.

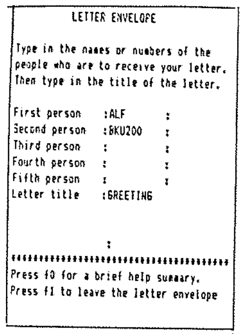


Figure 5 Completed envelope

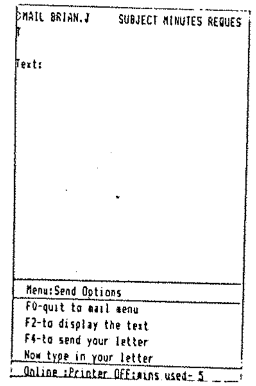


Figure 6 Text menu

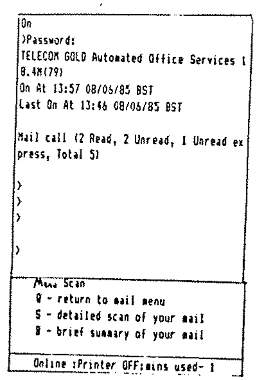


Figure 7 Scan menu

5

** 8) After typing in your message, press f2 if you wish to display the message.

** 9) When you have finished the letter then press f4 - to post the letter.

** 10) The letter is now finished and has been posted and you will see the help screen again.

** 11) To leave the send procedure press SPACE BAR and then f1.

Scanning a letter

** 1) If you are starting from the beginners menu (see figure 1 above), to select the scan a letter option,

*** Press the button M on the BEGINNERS menu, see figure 1 above.**

This will display the mail menu as in figure 2 above.

** 2) You see that there are now four options to choose from. If you wish to send a letter,

*** Press the P button for the SCAN option.**

You will then see the information in figure 7.

** 3) This gives you two options B for a Brief scan of mail and S for a more verbose Scan of mail. Select one of these options by,

*** Pressing B or S**
and then read the scan on the screen.

6

For example to select the brief scan option,
From the beginners menu (figure 1)
*** Press the M key on the keyboard.**
This will display the mail menu.
*** Press the P key on the keyboard**
This will display the scan menu.
*** Press the key B**

** 4) If there are only a few letters these will be displayed in the top part of the screen. The information displayed will be the date the letter was sent, who the letter was from etc. Then then you will be returned to the mail menu.

** 5) If there are more letters than one screenful then you will see the continue menu as shown in figure 10 on the next page.

** 6) If you have seen the letter you want to read, remember the number as you can request this particular number later when reading a letter, and press Q to quit scanning. Otherwise if you wish to continue to scan,

*** Press the RETURN key.**

Reading the letters

Reading a letter by electronic mail is similar to reading a normal paper letter.

** 1) If you are starting from the beginners menu (see figure 1 above), to select the send a letter option,

*** Press the button M on the BEGINNERS menu, see figure 1 (p2).**
This will display the mail menu as in figure 2 (p2).

** 2) You see that there are now four options to choose from. If you wish to read a letter,

*** Press the R button for the READ option.**
You will then see the information in figure 8 on the facing page.

7

```
?Password:
TELECOM GOLD Automated Office Services 1
B.4M(79)
On At 13:57 08/06/85 BST
Last On At 13:46 08/06/85 BST

Mail call (2 Read, 2 Unread, 1 Unread ex
press, Total 5)

>
>
>

>

Menu Read
 M-to mail menu Q-to beginners menu
 OR read the following
 E-express letters U-unread letters
 A-all the letters C-choose a letter
 Online :Printer OFF:mins used- 1..
```

Figure 8 Read menu

```
Mail call (2 Read, 2 Unread, 1 Unread ex
press, Total 5)

>
>
>

>MAIL READ UNREAD


  From: B.JUDSON (TFN001) Posted: Sa
t 8-June-85 7:37 BST UK BST Sys 85 (
B9)
Subject: LAUNCH OF RNID ACCOUNT
Acknowledgment Sent

--More--
 Menu More/Action required
 Q-quit to mail menu
 or continue to read using
 RETURN-continue read A-read again
 N-next letter D-delete the letter
 Online :Printer OFF:mins used- 1..
```

Figure 9 More menu

```
--More--

THIS IS TO INFORM THOSE WHO HAVE
SPECTRUM MICRO OR ARE INTERESTED
THAT I HAVE BEEN ABLE TO CONNECT
TO THE RNID TELEPHONE EXCHANGE BUREAU
USING THE SPECTRUM. THE SET UP IS
COMPLETED WITH THE USE OF PRISM VTX5000
MODEM WHICH WORKS IN PRESTEL MODE

THE PRISM MODEM IS CURRENTLY ON THE
MARKET FOR 50 PDS AND THE OLDER SPECTRU
 IS SELLING FOR AROUND 90 PDS IF
AVAILABLE AND THE SPECTRUM PLUS IS
SELLING FOR 130 PDS SO THATS MAKES
A RANGE OF 140 TO 180 PDS TO BE ABLE
TO ENTER INTO THE WOR
 Menu:Continue

 Press Q to quit to main menu
 Press RETURN to continue

 Online :Printer OFF:mins used- 2..
```

Figure 10 Continue menu

.8

** 3) You may now select an option to read by pressing one of the following keys,

Q - QUIT to the beginners menu,
M - quit to the MAIL menu,
E - read the EXPRESS (that is the important) letters,
A - read ALL the letters that you have,
U - read all the letters that you have not read yet, i.e. your new (UNREAD) letters,
C - CHOOSE a letter to read, after selecting this option you will be asked to specify the number of the letter in your list of letters. This you may have obtained when you scanned through the letters.

Example

If there were 6 letters of the following types, 2 express, 2 unread and 2 read letters then pressing A would cause all the letters to be read starting at the express letters, then the unread letters would be displayed and then the read letters.

If E was selected then only the express letters would be read starting with the UNREAD EXPRESS letters and finishing with the EXPRESS letters that had been READ before.

If U is selected then only the unread letters are read.

If the C option is chosen then you will be asked to select a number corresponding to the letter you wish to read. You may have seen a letter you wish to read from using the scan option. If you have seen the the letter you wish to read from BKU200 is number 4 then press the C option and press 4 RETURN.

** 4) After the first part of the first letter has been displayed you will be presented with the menu (figure 9).

** 5) You can then select from the following options. Press the key below to obtain the response,

*** Press one of the following.**

RETURN - to continue to read the next part of the letter,
A - to read the letter again, (e.g. this is useful if the display has been corrupted in any way),
N - will allow you to go onto the next letter without reading any more of the present letter
Q - will allow you to quit when you have finished reading the letters you require.

On pressing one of these options you will be asked to wait for a few seconds while the next part of the letter of the letter comes onto the screen. When the letter stops being displayed you will either see the menu in fig 9 again or the one in figure 10 below. If the second menu is displayed (figure 10) you have 2 options, either to (Q) quit or (RETURN) to continue.

** 6) If you wish to finish you either continue until the letters are exhausted or,

*** Press q to quit at one of the continue menus as above.**

The beginners menu will then be displayed again.

<u>OFFLINE</u>

The only other option in the beginners menu is the offline function this will disconnect you from the distant computer and lock your mailbox for you automatically. Use this function to complete the session.

** 1) In the beginners menu (see figure 1 above), to select the offline option,

*** Pressing the button O on the BEGINNERS menu, see figure 1 above.**

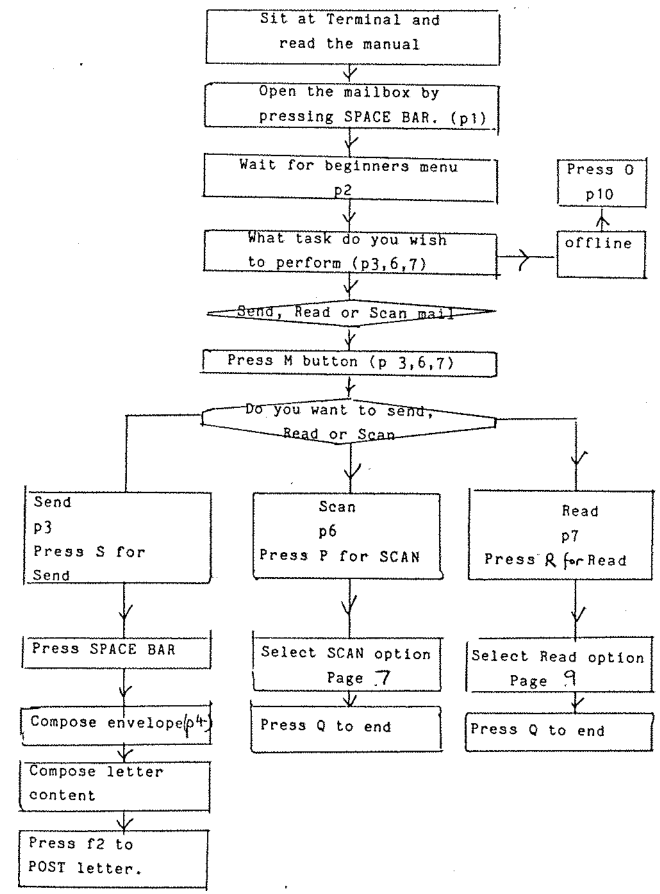
SUMMARY

The equipment helps you to open your mailbox, this is achieved simply by pressing the SPACE BAR at the beginning of the program.

The above instructions have been given to help you send, read and scan your mail. When you are inside your mailbox you can send letters by selecting the (S) send option in the mail menu, you are then asked to compose an envelope and then the text. To read you use the (R) Read option and to scan you use the (P) Scan option. When the session is finished you select the (O) OFFLINE option in the beginners menu.

Index

Flow diagram for the use of the mailbox



APPENDIX 9

SAMPLE TASK SHEETS

Subject 4

Task Sheet 1

Thank you for completing the first questionnaire and for agreeing to take part in the evaluation. BEFORE PROCEEDING ANY FURTHER PLEASE READ THE FIRST SCREEN ON THE COMPUTER AND THEN PRESS THE SPACE BAR.

You will be asked to attempt two different ways of using an electronic mail system. Each system will be set up for you and you should read the information provided on the computer screen first. The two sessions may be videotaped to record the screens of the computer.

Please do the tests at your own speed but try to imagine you will be paying for the calls and so do not take longer than is really necessary.

Please feel free to use this page to make notes on and to write down the time of the party.

In session 1 please attempt all of the following tasks,

** a) read the reply from K.WATSON about the TIME OF PARTY which will be in the mailbox and make a note of the time of the party on this sheet.

** b) send letters to MARK and JENNY containing the information 'Are you going to the party tonight'

** c) close the mailbox by pressing the disconnect key (f6) and then open the mailbox, your system number is 76, your ID is XCH009 and your password is JEREMIAH.

When you feel you have completed the tasks or think you have had enough please finish by pressing the disconnect button (f6). Please inform the experimenter (Keith Watson) that you have finished the first part.

Subject 4

Task Sheet 2

In session 2 please attempt all the following tasks,

** a) read the reply from K.WATSON about the TRAIN TICKET PRICE which will be in the mailbox and make a note of the price of the ticket.

** b) press the Offline disconnect (O) key and then open the mailbox,

** c) send letters to FRED and HARRY containing the information 'Can you make the meeting on Monday?'

When you feel you have completed the tasks or think you have had enough please finish by pressing the disconnect (O) button in the Beginners menu. Please inform the experimenter that you have finished.

APPENDIX 10

FUTURE SOFTWARE METHODOLOGY

Benbasat and Wand (1984) Human-interaction event model

The prototype interface program was developed using a recognised software methodology of state diagrams and pseudo English algorithms. In retrospect it may have been better to use a high level dialogue design specification language specifically optimised for interface design. Benbasat and Wand (1984) have developed one such method which, with certain modifications, could be suitable for this sort of design. The methodology they suggest is based on an interaction event.

> 'The model is based on a dialogue being viewed as a sequence of basic interaction events. The definition of the dialogue is made up of the event definitions and flow control which defines all the possible sequences of events. All interaction events have a common structure based on common generic elements necessary for a successful dialogue.
> An interaction event is defined as an occurrence in the dialogue where the system awaits input from the user..'

The interaction events are represented as an event table where each element of the table could be described as an input to a processor. They proposed the following set of generic elements which can be used to specify all interaction events.

1. Prompt - indicates system is awaiting input.
2. Input - (get from user or default)
3. Escape - if input escape, set next event indicator, end cycle, this allows the user to escape from an event that was accidentally requested.
4. Help - if help then display additional information, end cycle, this takes dialogue back to 1.
5. Check - input check, If errors report and end cycle, this takes dialogue back to 1. This is a conditional abort and the error generated causes a repeat of the cycle.
6. Action - invoke related processing algorithm
7. Flow - set 'next event' indicator and move to next event.

The flow control should allow the software designer to implement any desired sequence. They proposed three possible flows models based on program structures. These were the sequence, case and skip (that is, while .. do). They represented each event using a 'set reference' technique. Each event was organised into ordered tables with the associated generic elements as follows:

<ID; prompt; input(default); escape; help; check; action; flow>

Each element then referred to another table where the relevant information (e.g. message, action, etc.,) is stored. The main advantage of this arrangement is the distinction between the event descriptions, which contain the dialogue "logic" and the actual information about text, processing definition, checks, etc.' In the implementation each interaction event can be recorded either as an in-memory table or as a member of a file on disc. This allows the possibility of easy dialogue changes, multiple types of dialogues, the development of dialogue generators, etc.

Disadvantages and Improvements

The Benbasat and Wand model was specifically used with command languages and needs to be modified to make it more general and more applicable to menu-based or window-based dialogues. Further work is required to compare this methodology to other high level dialogue languages and the following modifications may be required.

In their model it seem that the escape event is only allowed in the input phase of the cycle. This is restrictive and is uncommon to many dialogues, particularly where the user is waiting for a process of action to finish. This is true in telecommunications systems where the action may be of long duration or where interim responses are required. The model should include an interrupt feature which allows the user to escape from any part of the interaction event.

This model is for a command based dialogue where messages and prompts are usually consecutive. In menu based or window based dialogues reprompts may not be required. For example if all error and help messages have their own window (as in the project software) then there is no point in reprompting the user. Alternatively if the help window does not overwrite the prompt window there will be no need to reprompt the user. Another dialogue feature is therefore required to suppress

the reprompt on certain occasions.
In the table allocated to the select function (i.e. case) their implementation only allowed a one to one mapping of ID to next event. This is fine for a command based dialogue but in a menu based dialogue different part of the program may use the same command or key to do different functions. This can be overcome by having the flow table split into three generic elements, the ID, case and next event elements.

The model only allows for the user to be able to change state for a user input. It does not allow any other input (e.g. the trapping of host responses used to set menu levels) to change the state therefore the model must allow other input to change the state. In this refined model each interaction event has at least one input condition and one output condition. A condition must be satisfied before it is possible to move to another interaction event.

Using tables it is easily to see the general structure of the program, the flow control, the different actions and checks etc. Each element refers to another table where the main procedures of the program can be found. For example, in the prompt phase each element of the table is denoted by an identifier. To find the prompt associated with event <example> one would move to the text table and then look for the ID <example> where the text would be recorded along with other information should as the display position. Similarly the flow tables contain the information to direct the reader to the next event.

Advantages

1) These tables provide a method of describing the interface which can be easily understood without resorting the decoding the program. This makes future modification easier.
2) Their method parallels structured programming. In future it may be possible to develop a high level language which may be used to build a dialogue generator.
3) All element of the same phase are in the same table so comparisons can easily be made. For example the flow table can be used to check for commands that may have been duplicated.
4) Changes can easily be made without changing the dialogue 'logic' by making an amendment to one of the tables.

APPENDIX 11

AN ESTIMATE OF THE COST OF TEXT TELECOMMUNICATIONS

A) Installation Costs

1) Cost of equipment.

The equipment required to access a text communications system is,

* home computer or terminal
* modem (either separate or built into terminal)
* communications software
* monitor or TV (optional if a screen is already in-built)

Optional extras include

* disc drive (or cassette) and printer

The prices of a few systems are given below.

a) Based on a simple terminal,

Tandata Td1404	£260
colour TV	£140
Total	£400

b) Based on a lap-held portable

NEC8201	£260
modem (acoustic)	£ 70
Total	£330

c) Based on a wordprocessor

AMSTRAD PCW8256	£458
RS423 interface	£ 68
SAGE modem and software for Email and PRESTEL	£239
Total	£755

d)

BBC Master computer with disc interface	£500
modem (Multi-baud rate, Tandata Tm512)	£250
monitor (Colour)	£200
communications software (with modem)	£ 20
Sub Total	£860
Optional extras	
disc drives (Teac 5 1/4")	£150
printer (e.g. Epson FX80)	£360
Total	£1310

2) Installation of phone line.

Current cost (£85 exc VAT) £ 97.75

Alternatively if an old style phone is already owned the cost of installing a new plug type socket is £25 excluding VAT or £28.75.

3) Registration fee to mailbox system

For a new independent user to Telecom Gold this is £100 but for the deaf there are spare mailboxes available free at the present time from the RNID or Breakthrough Trust for a nominal fee.

RNID mailbox £ 5

B) Running Costs

1) The cost to use the telephone can be split into three charges.

a) quarterly rental charges for residential users for the line without a telephone instrument £13.45 or with phone instrument £16.70 excluding VAT.

b) PSTN charges. Dialled calls are charged in units of 5p exc VAT (i.e. 5.75p). The number of seconds per unit are:

Figure A11.1 Table of Phone Charges.

	Peak	Standard	Cheap
Local	90	120	480
Up to 56km	30	45	120
Over 56km & Channel ls.	17.1	22.5	48
Over 56km (busy rates)	22.5	30	60

Peak	Mon-Fri	0900-1300
Standard	Mon-Fri	0800-0900, 1300-1800
Cheap	all other times.	

c) and PSS charges, which are 2.5p per minute at 300 baud and 3p per minute at 1200/75 baud.

2) cost of using mailbox system.(example rates for Telecom Gold)

11p per minute prime time [8a.m. to 7p.m.] for all users.

3.5p per minute (non-prime time) for business users and £5 flat rate per month non-prime time for deaf users.

Using this information we can give approximate costs for using the following systems.

i) Mailbox system communications

a) Installation costs will include the equipment (£500), mailbox registration fee (£5), and phone installation (£100).
Total is approx. £600.

b) The running costs per quarter for a deaf user of an electronic mail system is,

1) £13.45 quarterly line rental plus £3.25 quarterly phone rental exc VAT. i.e. £19.21
2) £15 per quarter (i.e. 3 * £5 flat rate special charge per month) Telecom Gold plus storage costs (minimal charge).
3) Assuming 20 minutes / day (see Walgreich 1982) in non-prime time (i.e. local offpeak using the PSS network - because this is cheaper than long distance). 17.25p for 20minutes phone charges and 60 pence per 20 minutes PSS charges (at 1200 baud). For a quarter this is about £70.00.

Total running cost for a deaf person is about £105 per quarter.

ii) Back to Back communications

a) Installation Costs will similar to those for the mailbox system except there will be no need to pay registration fees for a mailbox.

b) The running costs per quarter for a deaf user of a back to back system can be calculated roughly as below.

1) £13.45 quarterly line rental plus £3.25 quarterly phone rental exc VAT. i.e. £19.21
2) From a statistical survey done by the Belgian Telecommunications Administration (Bolle 1984) it was shown that telephone subscribers with hearing use their telephone at home for an average of about 230 minutes per month. If it is assumed that 2/3 of these conversations were with permanent contacts then we have about 155 minutes per month. In the model proposed, Bolle wrongly

assummed a conversation by text telephone would take twice as long as by ordinary phone converation. In fact a paper in the same conference (Besson 1984) and a practical study conducted by the VISICOM project (Judson 1986) showed that a text call lasts about SEVEN times longer than a spoken call. So for the deaf person this would be 155 * 7 = 1085 minutes per month.

Let us assume a model of calls as follows, where time (t) is minutes/quarter.

t(total) = t(local) + t(medium) + t(long)

where t(local) = 0.5 * t(total) = 1620 minutes

t(medium) = 0.3 * t(total) = 975 minutes

t(long) = 0.2 * t(total) = 651 minutes

This model will change depending on the proportion of local, medium, and long distance calls but it provides an order of magnitude answer. Using the figures given in table A11.1 we have,

total number of units = (1620/8) + (975/2) + (651/1.5) units

total phone charge = 1125 units * 5.75 per month

= £65 per quarter.

and including rental = £85

So the costs are comparible for a mailbox and back to back system. In practice the deaf user will probably use a combination of the two types of system because mailbox systems are preferable for long distance communications (because of the use of the PSS network) whereas text telephones are better for local communications.