A STUDY OF THE PROBLEMS OF MAN-COMPUTER DIALOGUES FOR NAIVE USERS

BY

MARTIN MAGUIRE

THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
TO THE COUNCIL FOR NATIONAL ACADEMIC AWARDS

DEPARTMENT OF MATHEMATICS,
COMPUTING AND STATISTICS,
LEICESTER POLYTECHNIC.

JANUARY 1981

ABSTRACT

The success of an interactive computing facility will depend, to a large extent, upon the effectiveness of the man-computer dialogue which it supports. Comparatively little work has been directed towards the design of effective dialogues for situations in which the 'man' is a 'naive' user i.e. a person without training or experience of computer procedures. Thus the aim of this project has been to produce a series of specialised guidelines for designers of dialogues for naive users.

An examination of the literature reveals that published dialogue guidelines tend to be of a general purpose nature and therefore cannot be applied directly to specific situations. Furthermore, as each set of recommendations is based upon a limited range of experience, authors opinions appear to contradict or be in need of further qualification.

At a practical level, a survey of computer games, intended to be self-explanatory and therefore suitable for naive users, bears out the widely held feeling that the dialogue interface is often a poorly considered aspect of interactive program writing.

Pilot studies highlight the need for experimental work into man-computer dialogues to be carried out under conditions conforming as closely as possible to a 'real world' environment.

The main study focuses upon the general public as users of a local information system developed and installed in Leicester's Information Bureau. Monitoring the public's usage of and reactions to the system has enabled a series of dialogue guidelines for public information systems to be produced. A review of the literature provides supplementary recommendations.

The influence of dialogue recommendations on the software writing community is considered. Less than half of a sample of application programmers are found to refer to material of this kind. Follow up interviews indicate that the concept of a dialogue guideline is too narrow and should be broadened to cover all types of dialogue design information. This would render it more applicable

to differing design situations. For designers who do not refer to published material, it is suggested that sound principles can be communicated via trained experts and the use of library subroutines supporting dialogue creation. An example is considered of a routine to process textual inputs.

A number of paths for future research are described concerning the development of experimental methodology suitable for testing man-computer dialogues, an evaluation of the proposed strategy for communicating dialogue design principles and the application of new input/output techniques to public information systems. It is also suggested that the likely social consequences of computerised information facilities should be determined.

ACKNOWLEDGEMENTS

This thesis has fallen victim to it's author's verbose style. In atonement, thanks are extended briefly, but no less gratefully to the following people:

Dr. E. A. Edmonds, Supervisor (Leicester Polytechnic)
Mr. K. D. Eason, Supervisor (Loughborough University)
The staff of Leicester's Information Bureau.
My family.
Helen Briggs.
Mrs. V. A. Mann.

CONTENTS

CHAPTER 1 | PAGE

CHAPTER 2 PAGE

CHAPTER 4 PAGE

CHAPTER 5 PAGE

PAGE

CHAPTER 7 | PAGE

CHAPTER 1

INTRODUCTION

THE DEVELOPMENT OF INTERACTIVE COMPUTING AND AN EVALUATION OF RECOMMENDATIONS FOR THE DESIGN OF MAN-COMPUTER DIALOGUES

The advent of interactive computing in the early 1960's transformed the computer into a tool of much greater convenience. Whereas in batch mode programs submitted to the machine were returned only after several hours, the new interactive systems could provide a response to user input in a matter of seconds. Thus the concept of a 'dialogue' between the man and the computer came into being.

Since the cost of computer hardware was very high in relation to the cost of the people using it, the effectiveness of this interface was ignored in favour of seeking ways of utilising the machine's central processing unit and storage facilities more efficiently. Thus dialogue styles tended to be conducted in computer orientated terms, agreeable only to the computer specialist. However, people from other disciplines, attracted by the convenience of on-line computing, soon expressed dissatisfaction.

To quote from Walther and O'Neill (1974):

> "As their initial fascination with conversational computing wore off, users reported experiencing feelings of intense frustration and of being 'manipulated' by a seemingly unyielding, rigid, intolerant dialogue partner, and these users began disconnecting from time-sharing services at a rate which was very alarming to the industry."

The introduction of on-line systems into many industrial, academic and public service environments has widened further the numbers of non-specialists who use computers as tools to help them carry out their tasks. This in turn has brought increased criticism of poor dialogue interfaces.

Another trend is the reversal of cost relationships. Mini and micro computer hardware is now relatively cheap compared to the cost of the person using it. Thus from an economic point of view it is necessary to provide systems which allow the user to make efficient use of his own time as opposed to the computer's time.

This project, therefore, is concerned with the need to devote greater research effort towards the design of successful man-computer dialogues. It is concerned in particular with dialogues via on-line display devices with full keyboards.

1.1. NATURAL LANGUAGE DIALOGUE

The need to orientate the man-computer interface towards the man has been interpreted by some researchers to mean that dialogue should be conducted in natural language (i.e. plain English, French, German, etc.) Since this view has formed the basis for extensive work into dialogue design, the main arguments for and against this approach are briefly discussed.

Clearly the main advantage of an English language interface is that the user can frame inputs in his own words and need not learn a formal computer language. Also a complex input can be entered immediately as opposed to building it up by the selection of items from multiple menu lists which can be a tedious process. The ROBOT system (Harris, 1977) is an example of how an English language interface can be successfully implemented if it is applied to a limited domain of discourse.

Other authors have expressed reservations about the design of natural language systems. The point has often been stated that English is so full of ambiguities, that it is unsuitable for interactive situations where precision is required. Furthermore, when the user communicates with the machine in an English-like language, he may be deceived into thinking that the computer possesses a deeper understanding of the subject under discussion than it really does.

The problem is explained by Fitter (1979):

> "So often with English there is an implicit assumption which a normal person is expected to understand though it is unlikely that a computer would have the necessary knowledge available to make the inference. Expressing the statement in plain English tends to obscure the ambiguity and hidden assumptions. In the notice 'Dogs must be carried', seen on the London Underground there is an assumption which does not apply to the instruction for motor cyclists in the Highway Code, 'Crash helmets must be worn'."

A number of programs have been devised which allow the computer to converse in natural language with virtually no understanding of the subject being discussed. Perhaps the most well known of these is the ELIZA system created by Weizenbaum (1966). By performing syntactic transformations upon user inputs, the computer responds in a surprisingly convincing way.

Consider the following example in which the computer, assuming the role of a psychotherapist, converses with a female user acting as a patient.

USER: "MEN ARE ALL ALIKE."

(Nothing within this statement is recognised by the computer and so an all purpose response is given).

COMPUTER: "IN WHAT WAY?"
USER: "THEY'RE ALWAYS BUGGING US ABOUT SOMETHING OR OTHER."

(The only word understood here is 'always' which prompts another standard reply).

COMPUTER: "CAN YOU THINK OF A SPECIFIC EXAMPLE?"
USER: "WELL MY BOYFRIEND MADE ME COME HERE."
(Here the word 'well' is discarded, the pronouns 'my' and 'me' are converted to 'your' and 'you' and the sentence is repeated).

COMPUTER: "YOUR BOYFRIEND MADE YOU COME HERE?"

By demonstrating this system, Weizenbaum has alerted designers to the dangers of users having unrealistic expectations of the computer's power.

1.2. STUDIES AND RECOMMENDATIONS ON THE DESIGN OF INTERACTIVE DIALOGUE

A more generally accepted starting point from which improved man-computer dialogues should be developed, is that they should appear as 'natural' to the user as possible. This is meant to imply that the system should appear easy and satisfying to use and should enable the user to carry out his task more effectively. A number of recommended guidelines have been put forward to assist dialogue designers in reaching these aims. During their discussion below it is of interest to note that a number of these appear contradictory.

1.2.1 Knowledge of and communication with the intended user

Hansen (1971) recognises the need for the designer to build up a profile of the user for whom the system is intended to serve. The features of this profile should include the user's education, experience, interests, how much time he has, his manual dexterity, the special requirements of his problem, his reaction to the behaviour of the system and his patience. This knowledge should be of great value in tailoring the system to a particular user's needs.

Gaines and Facey (1975) emphasise the importance of assessing the thought processes, language constructs and vocabulary which a person employs when carrying out his normal tasks; or in other words forming a model of these tasks. Therefore, when a computer tool is designed to assist him with his activities, it should appear to think and communicate in the same manner that he does himself.

They also advocate the designer showing intended users a similar system in action, before any design work takes place. Having experienced interactive dialogue, these people can then discuss their expected use of a computer from a more informed standpoint.

1.2.2 Use of a specialist intermediary or documents

The assumption that the user should always be required to interact directly with the machine is a false one. Eason (1976a) describes the problem of the manager user whose tasks are variable and unpredictable in nature. To carry out these tasks the information which he requires from the computer will also be variable and unpredictable. However, it appears that managers are unwilling to learn the necessary dialogue procedures which this form of complex information retrieval would demand. One solution is to insert a human intermediary between the manager and the interactive system. By operating the system on behalf of a number of managers, the intermediary becomes a frequent user who can thus master the full range of facilities which the complex dialogue allows.

The experience's of Gaines and Facey (1975) have led them to advocate the circulation among users of computer produced documents in certain cases as opposed to interactive information retrieval. They argue that the portability of such documents together with the human ability to rapidly scan them should not be overlooked. This approach is likely to be of value when information is required on a routine basis or in a standard form.

Given that the designer has decided to implement an interactive system for direct use, the remaining principles apply to the creation of interactive dialogue itself.

1.2.3 System response times

The question of the time which the system should take to respond to user inputs is highly controversial. Schneiderman (1979) has reviewed the main strategies which have been put forward. Three of these are as follows:

Firstly, that response times should be reduced under all conditions and ideally kept within two seconds on the majority of occasions. The argument used in support of this statement is that fast responses cause the least disruption to the person's thought processes.

In contrast, it is said that the response time should match the complexity and cost of the computer processes triggered by the input. By this method it is thought that users will be made to favour efficient commands. Miller (1968) has provided a comprehensive list of maximum response times for seventeen different activities.

The third, also conflicting recommendation is that the variability of response times should be reduced even at the expense of some increase in mean response time. A study by Miller (1977) has shown that increasing variability generates poorer performances and lower user satisfaction.

1.2.4 Command synonyms and abbreviations

Wasserman (1973) states that the unsophisticated user cannot understand seemingly unnatural restrictions imposed upon him by computer programs and quotes the example of a program that will not accept any variation of the input "YES" even when it is merely preceded by a leading blank. He then suggests that synonyms such as "Y", "SURE" and "CERTAINLY" should be allowable. Cheriton (1976), agreeing with this view suggests that "CHANGE", "MODIFY" and "ALTER" might be equivalent commands. However, notice should be taken of Schneiderman's (1979) view that novices prefer and do better with a system which has few choices and permits only limited forms of expression.

Another aspect of command flexibility, is the programming of the system to accept commands in abbreviated form. This allows a user to enter a message more efficiently with fewer key depressions. Cheriton's (1976) recommendation is to allow abbreviations of commands down to the minimal unambiguous character string. Meanwhile Newman (1978) modifies this view by noting that a facility of this kind can lead to serious misunderstandings between user and system:

"As an example, a system which would accept Algol 60 reserved words... allowed '<u>b</u>', '<u>be</u>', or '<u>beg</u>', etc. to mean <u>begin</u>, and '<u>r</u>', '<u>re</u>', '<u>rea</u>', etc. to mean <u>real</u>. However, it interpreted the word 'print' which had been inadvertantly underlined to mean <u>procedure integer true</u>".

1.2.5 Help facilities and tutorial aids

Quoting Kennedy (1974)

"A help key or command which gives a detailed description of a selected part of the system is essential if only to give confidence to a casual or naive user. It is often difficult, initially, to persuade someone to use a terminal when they are unsupervised since they have no means of asking questions. If the user can ask the computer what to do at any stage or explain how a particular function works, the procedure of familiarisation proceeds fairly rapidly".

As well as assistance for inexperienced users, all users tend to forget aspects of the system after a certain time. Help facilities can therefore be a convenient aid, especially as documentation manuals are often found to be excessively long and complex, out of date or missing pages. Gaines and Facey (1975) describe a particular technique for providing tutorial help which they call 'query in depth'. At any point where the user is requested to make an input by the system he may type a question mark. This produces information about what is required from the user or a list of example inputs.

For example consider the following data entry sequence:

```
ENTER MAKE OF CAR:              FORD
MODEL:                          CORTINA
BODY TYPE:                         ?
SALOON, ESTATE OR COUPE
BODY TYPE:
```

The user may then either type his input or enter a second question mark for more detailed assistance. Further levels of query can be established up to the point where a section of the manual is displayed or an appropriate page in the manual is referenced. The advantage is that the user can control the amount of information he receives (he will not be overwhelmed with data on typing the first question mark) and he rarely needs to search through a complex manual which has been found to fall into disuse.

An added footnote taken from Stewart (1979) is that documentation in the form of simple look-up lists, reminders, prompts and indexes can provide the user with considerable reference material in a small convenient space.

The concept of the computer being able to perceive that the operator needs help, and then providing it automatically, is under question. Wasserman (1973) recommends that the programming language should have provision for keeping a time check so that a user failing to respond can be prompted and eventually logged out. Gaines and Facey (1975) did not find this approach satisfactory and from experience give an example of a user who was carefully thinking out her actions before keying them in. On receiving a timed out response she became very annoyed and refused to use a terminal again.

An indirect method by which the computer can help the user is to reduce the need to memorise commands. Rather than type a character string or operation name from memory, the user should select the appropriate item from a list displayed by the computer. Hansen (1971) who calls this principle 'selection not entry' has found that it has worked well on the EMILY text editing system. However, Stewart (1979) shows by example that systems cannot all be designed in this way. The British Post Office's PRESTEL system was designed to allow the user to access any page in the database by means of numbered menu pages displayed on the screen. However, up to ten menu selections were required to reach certain end pages; an expensive process since most pages cost money. A facility for accessing end pages directly was clearly required and so a series of directories had to be produced to contain the necessary input codes.

1.2.6 Prevention and cure of operator errors

It is generally accepted that the system designer should try to minimise the possibility of operator error and to build facilities into the system for handling those which do occur. Stewart's (1979) suggestions for reducing the likelihood of errors include; making the sequences of operations seem as natural to the user as possible and maintaining a level of consistency in the style of input. Avoiding unnecessary shift changes can also prevent mistakes. He then goes on to recommend checking that common errors do not have catastrophic effects unlike two cases which are sighted in his paper.

Hansen (1971) notes that the learner has a powerful and reasonable curiosity to find out what happens when he does something wrong. The system, therefore, must be made as incorruptible as possible against such action.

If the user takes an incorrect dialogue path, it is necessary for this to be spotted as early as possible before many subsequent inputs have been made. Gaines and Facey (1975) discuss this problem and state that if the computer's responses are designed to identify the activity taking place, then the user's ability to recognise such an error will be strengthened. He can then take appropriate action to exit from the current activity.

The need for well designed error messages is also widely recognised. Schneiderman (1979) writes that they should be understandable, non-threatening and low-key. As an example the blaring message:

"ERROR 435 - NUMBERS ARE ILLEGAL"

would be better replaced by the simple response:

"MONTHS ARE ENTERED BY NAME"

which tells the user what is needed to put things right.

The timing of the output of error messages is a point of issue.

From Martin (1973, p. 450)

> "The (error) response should not be overly abrupt... a split-second error response in midthought is jarring and rude. The operator should be permitted to finish his thought before the error response is sent".

However, Schneiderman (1979) refers to a study by Segal (1975) which

> "......suggests that human performance improves if errors are issued immediately and the disruption of user thought processes by immediate interruption is not a serious impediment".

The advantage of immediate interruption is that a correction can be made by simply changing the invalid character.

The technique of querying inputs which although not incorrect, lie outside the expected range, has frequently been advocated. However, Gaines and Facey (1975) provide a warning that valid data can frequently exceed the prescribed 'norms' and users become frustrated in having to say 'yes I really did mean that' too often. Martin (1973, p. 451) also alerts designersto the danger of the operator coming to rely upon the system picking up all his errors and thus adopting a careless attitude to data entry.

Finally, Jones (1978) puts forward the idea that since human beings learn all tasks through a process of trial and error, then interactive systems should be designed to encourage 'operation by experiment'. Two of the main prerequisites needed to enable the application of this technique are; that the cost of a series of trials should not be excessive and that the outcome of one trial should indicate what alterations to make for the next trial.

1.2.7 Adaptive systems

There may often be the need to modify parts of the dialogue interface after it has been implemented. Hansen (1971) explains that observation of users in action will show that some commands are not as convenient to use as their frequency warrants. It will be required then that such commands can be modified.

Edmonds (1978, 1981) advocates dialogue adaption of a more general kind and by all levels of user from the untrained person to the specialist. He also emphasises the need to employ implementation methods which will allow easy and low cost system adaption.

The advantage of using the adaptive method in dialogue design is that the system:

> ".......will tend to recover from mistakes; consequently evaluation is less critical than a once-and-for-all investigation would be. The software will also be modified as the users' requirements change...(and) the development need never be considered complete". (Edmonds, 1974)

1.2.8 Feedback from the user

The designer should build adequate facilities into the system to obtain feedback on interactive sessions which take place. Thus any problems which users experience can be ironed out following system implementation.

Hansen (1971) describes several mechanisms contained in the EMILY system for obtaining feedback. Firstly, the computer automatically logs all user interactions, user errors and system errors. A command is also included which allows the user to place a message into the log. Finally, frustrated users can push a 'sympathy' button to receive at random, one of ten messages of comfort. This action is also noted by the system. The designer examines all these sources of feedback and determines whether any dialogue changes should be made.

Two further methods of monitoring system usage are noted; these are, to observe people operating the terminals and to encourage comments from the user community. Schneiderman (1979) advises the designer to:

> "Listen to their gripes with sympathy and be willing to modify your system to accommodate their requests. Remember, the goal is not to create a computerized system but to serve the user".

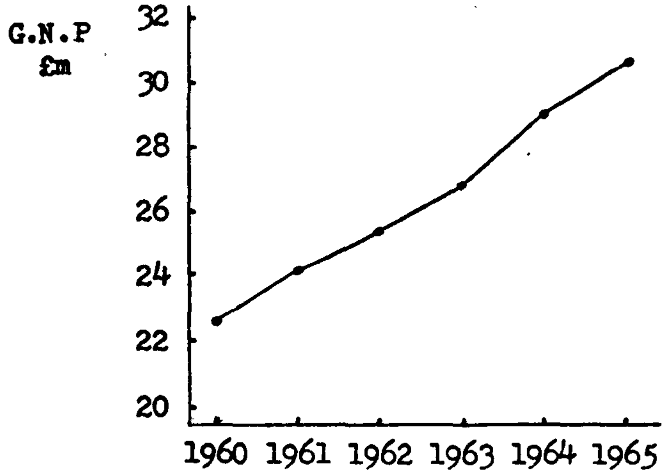
1.2.9 Alternatives to printed text

Martin (1973, p. 283) makes the important observation that while the human mind can accept information in a variety of forms, man-computer dialogues are frequently restricted to the single medium of printed text. He then sights a number of alternative media used in human communication which should be more widely extended to man-computer interaction. These include moving or still pictures, diagrams, graphs, charts and the spoken word.

The number of applications to which these forms of communication have been applied is very large and so two simple examples are taken from Stewart (1979) of information which can be presented more clearly in graphic rather than textual form.

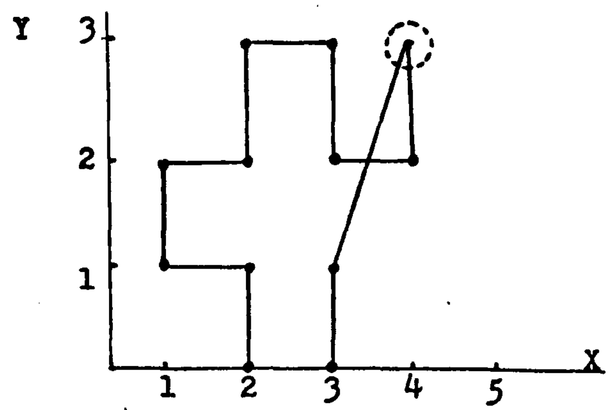
(i) Discerning trends in financial data - Gross National Product values for the United Kingdom (1960-1965)

	G.N.P.
1960	£22.7m
1961	£24.3m
1962	£25.4m
1963	£26.9m
1964	£28.9m
1965	£30.7m



(ii) Vetting co-ordinate data - the co-ordinates below should form an upright cross.

X	Y	X	Y
2	0	3	3
3	0	2	3
3	1	2	2
4	3	1	2
4	2	1	1
3	2	2	1



Value "4,3" should read "4,1" to form the correct cross shape

The reason why graphic presentation is often preferred to tabulated text is based upon the human perceptual system which constantly seeks to find structure in order to interpret the environment. As Stewart says:

> "suitably structuring a data display therefore increases the speed of interpretation and reduces the error rate".

Finally, high resolution graphic facilities are regarded as less important than an ability to interact with the machine to vary or request different graphic displays.

1.2.10 Non-verbal signals

Jones (1978) stresses the importance of non-verbal signals in everyday language such as the pitch of the voice, facial expression, arm and hand movements. Computer dialogues should therefore include non-verbal signals to enrich the conversation. Such signals include a bell or bleep, reverse video and flashing characters.

However, Schneiderman (1979) implies that such signals may not always be desirable, when he writes that ringing a bell to indicate an error will embarrass the user. Stewart (1976) states that although flashing characters can serve to distinguish important information, they may also disrupt the user's thinking process or cause headaches especially at frequencies of 5-15Hz.

Jones sees the layout of printed text on the screen as equivalent to facial expression:

> "I know of no way of making a VDU smile, but every art editor knows the value of white space in making a page look attractive".

He then suggests that spacing should be used to break up the text into logical segments and that the shorter the segment the more easily it will be read. Also relevant to ease of reading, Tinker (1963) has shown that people can read lower case more easily than upper case.

1.2.11 Handling different levels of user experience

It has been widely recognised that the user's requirements of an interactive system will alter as he gains in his experience with it. This occurs in two particular ways:

(i) While a novice user will require an interactive program to be fully self-explanatory, a user who is familiar with the dialogue will prefer quicker, abbreviated computer messages. In Wasserman's (1973) words:

"The explanations of program usage that typically are given to the first-time user become boring to a continuous user".

As an example, an extract is taken from Evan's (1970) highly successful, medical history taking program for use by patients with no experience of computer terminals.

COMPUTER:	"THIS IS A COMPUTER TALKING TO YOU. I CAN ASK YOU QUESTIONS BUT I CAN ONLY UNDERSTAND SIMPLE ANSWERS, SO YOU CAN ONLY TALK TO ME BY PUSHING EITHER THE 'YES', 'NO', OR '?' BUTTON. YOU SHOULD ONLY PRESS THE '?' BUTTON IF YOU DON'T UNDERSTAND THE QUESTION I AM ASKING YOU. NOW, DO YOU UNDERSTAND EVERYTHING I'VE SAID UP TO NOW? PUSH ONE OF THE BUTTONS".

This style of lengthy explanation would soon become tiresome if it were encountered frequently.

Wasserman's solution is to provide two modes of interaction; "NORMAL" and "QUICK" which could be requested by command at any point in the conversation. The former would provide extended explanation while the latter would give briefer messages recognisable to the experienced user. However, the division of users into 'new' and 'experienced' is too simplistic, according to Stewart (1979) and there is in reality a continuum with people at many different levels of experience and knowledge of the system. Gaines and Facey's (1975) technique of 'query in depth' (described in (6)) is perhaps better able to serve these many levels. Here the system assumes a style of brevity but provides explanatory information to a given depth in response to a user input of a given number of question marks.

Nickerson (1969) has proposed the use of this type of mechanism for handling the level of explanation within error messages. However, Kennedy's (1975) experimental work has shown that some novice users have difficulty in coming to terms with the concept of computer detection of input errors and so, in this case, a single level of message may be more appropriate.

It is also possible within the scope of a dialogue for novice users to provide a short cut mechanism for those with more experience. Martin (1973 p. 314) describes one such mechanism with reference to the following interactive sequence:

COMPUTER: "PLEASE TYPE YOUR NAME (LAST NAME FIRST)"

OPERATOR: "JONES, MARY"

COMPUTER: "PLEASE TYPE YOUR PERSONNEL NUMBER"

OPERATOR: "756093"

COMPUTER: "WHICH OF THE FOLLOWING DO YOU WISH TO CARRY OUT? ENTER LINE NUMBER

1. TEXT EDITING
2. CALCULATOR FUNCTIONS
3. COMPUTER-ASSISTED INSTRUCTION
4. MANAGEMENT INQUIRY
5. ORDER INQUIRY
6. CUSTOMER INQUIRY
7. ORDER ENTRY
8. CUSTOMER ENTRY"

OPERATOR: "5"

He then suggests that frequent users who can anticipate future inputs should be allowed to type in more than one response at a time. Thus by entering;

"JONES, MARY + 756093 + 5"

in response to the request for her name, the user could prevent the second and third computer questions being displayed. This technique is termed 'response-combining'.

(ii) The second requirement of the system which transpires with increased user experience is expressed by Schneiderman (1979):

> "Novice terminal users....are perfectly willing to follow the computer's instructions and accept the computer as the controlling agent in the interaction. With experience and maturity, users resent the computer's dominance and prefer to use the computer as a tool".

Hall's (1978) approach to this change in attitude is to allow the user to transfer freely between a dialogue style whereby he selects options displayed to him on the screen, and a style in which he enters commands to the machine from memory. Hall also presents a technical specification of this type of interface.

Palme (1979) proposes that this transference of control should take place in a more gradual manner and describes the stages through which the user might pass as his skill and confidence increase. At certain points within this scheme, the user may interrupt the dialogue to request further information or combine a sequence of commands into a macro which can then be called by entering a single command.

Eason (1979) extends the warning that such an approach would need careful management if the user is not to become satisfied by a particular level of achievement and cease to search for a higher level of control.

1.3 SUMMARY OF RECOMMENDED PRINCIPLES AND APPARENT CONTRADICITIONS

The dialogue design recommendations of the previous section are precied in columns one and two of the table below. Column three indicates, by means of arrows, groups of conflicting viewpoints and also contains a number of caveats which have been applied to certain recommendations.

Area of consideration	Recommendations	Conflicts (indicated by arrows) and caveats
(1) Knowledge of and communication with the intended user	The designer should familiarise himself with the attributes and tasks of the intended user	
	A prior demonstration of interactive dialogue will place users in a better position to discuss their requirements	
(2) Use of a specialist intermediary or documents	Do not assume that the user needs to interact directly with the machine. Appointing an intermediary or distributing computer listings may be preferable approaches	
(3) System response times	Minimize all response times	←
	Reduce response variability	←
	Link the response time to the computer operations triggered by each input	←

Area of consideration	Recommendations	Conflicts (indicated by arrows) and caveats
(4) Command synonyms and abbreviations	Unsophisticated users will expect synonyms of commands to be acceptable.	↕ (conflicts with the next recommendation)
	A wide choice of inputs will be confusing for the novice.	↕ (conflicts with the previous recommendation)
	To reduce the typing required, command inputs should be accepted up to their minimal unambiguous form.	Serious misunderstandings can arise when simple errors are made.
(5) Help facilities and tutorial aids	On-line help facilities and tutorial aids are of value since they provide assistance when and where required.	
	Create facilities which automatically detect the need for, and provide help to the user.	This can prove very annoying on occasions when no assistance is required.
	Users can be saved from the need to remember commands by conducting dialogue through a process of list selection.	Total reliance upon this approach can lead to laborious and costly interactive sequences.

Area of consideration	Recommendations	Conflicts (indicated by arrows) and caveats
(6) Prevention and cure of operator errors.	Reduce the possibility of errors by making inputs seem natural and consistent.	
	Provide the user with enough information to help him identify his own errors.	
	Error messages should be understandable, non-threatening and low-key.	
	The immediate transmission of error messages will speed up the interactive process.	←
	Point out mistakes after the completion of inputs to avoid disrupting the user.	←
	Query inputs which lie outside their expected range.	As valid data frequently lies outside 'norms', this can cause frustration.
	Encourage operation by experiment .	
(7) Adaptive systems	The lifespan of the system can be lengthened by designing it to be adaptable to user's evolving needs.	

Area of consideration	Recommendations	Conflicts (indicated by arrows) and caveats
(8) Feedback from the user	Adequate feedback facilities should be established so that system faults can be detected and suggested improvements be made.	
(9) Alternatives to printed text	Pictorial displays may be preferable to printed text, particularly when the information contains some form of structure	
(10) Non-verbal signs	The dialogue may be enriched by the use of bells, bleeps, reverse video and flashing characters.	Visual effects can cause headaches and disrupt thinking while auditory error signals may embarrass the user.
	Clear layout and the use of lower case lettering will improve the appearance of displayed data.	
(11) Handling different levels of user experience	Novice users will need explanatory dialogues while more experienced users will require a briefer form of interaction. Systems should thus contain two levels of dialogue.	There will be many more than two levels of user experience and so a more comprehensive range of dialogues should be available.

Area of consideration	Recommendations	Conflicts (indicated by arrows) and caveats
	Provide a command to request further explanation of computer outputs.	Regarding error messages, the novice user may not be sufficiently familiar with the concept of computer error detection to make use of this facility.
	Establish short cut mechanisms such as response combining.	
	Devise a strategy which allows the operator to gradually take control of the dialogue as he gains in experience.	

1.4 RESOLVING THE CONTRADICTIONS WITHIN THE RECOMMENDATIONS

The contradictory nature of many of the recommended dialogue principles may cause them to be regarded with scepticism by system designers. It might be suggested that experimental research is required to determine which one of two or more pieces of conflicting advice is correct. However, the two experiments concerned with response times described below, indicate that this may not be an adequate solution. (These details are taken from Schneiderman, 1979).

(i) Goodman and Spence (1978) arranged for thirty subjects to modify five parameters in a curve matching task. Computer response times were fixed at 0.16, 0.72 and 1.49 seconds. The average solution time for the task was found to be significantly smaller for the two shortest response periods.

(ii) Grossberg et al (1976) studied four subjects performing thirty six interactive tasks involving calculations upon numeric arrays. As response times were increased from 1 to 64 seconds, subjects displayed more care before issuing commands to the machine. However, the total task time did not appear to vary greatly with increasing response times.

Schneiderman suggests that the difference between the results of the two experiments can be attributed to differences in available commands, the tasks to be carried out and the subject's level of experiences.

Here it seems is a pointer to the reason for the contradictions in the recommended guidelines. Each specific application for an interactive situation is likely to consist of a unique combination of attributes which determine the nature of the dialogue required. Since recommended principles are based upon the writers' experiences of particular applications, it is not surprising, therefore, that conflicts of opinion will occur.

Developing this argument; when a list of guidelines are presented, they are based upon a series of unstated assumptions regarding the nature of the user and his task. For example, a frequent user carrying out a simple task will require far less in terms of explanation and help facilities than will an infrequent user with a complex task. Thus it might be more accurate to regard many of what seem to be conflicting guidelines as alternative strategies to be applied in different situations.

It should also be remembered that guidelines are constantly being put forward as researchers' knowledge and experience of man-computer interaction develops. Clearly some of the older guidelines will be found to be invalid or in need of revision. Therefore, the preservation in the literature of both old and new recommendations is another cause for contradiction.

1.5 PROJECT AIM AND APPROACH

The previous discussion seems to show that the recommended principles tend to contradict since they are drawn from different application areas. In the light of this conclusion, it was decided to concentrate upon a particular application of man-computer interaction. The area chosen was that of information retrieval by naive users. The project's aim was to construct a series of specific guidelines to aid future designers of this type of system. It was considered that such recommendations would not be generally applicable to other systems which would themselves require research.

The approach of the project was to design a number of information retrieval dialogues for use by naive operators and to evaluate them under experimental conditions.

1.5.1 Naive users

The naive user, for the purpose of this project is defined to be a person who has had little or no previous experience or training in using a computer terminal. It is also assumed that he operates the terminal on an infrequent basis and therefore does not become highly experienced.

Evans (1976) emphasises the importance of naive users in the future:

> "When we reach the point, which may not be too far off, when hardware is so cheap to produce that it can be given away and the only services that computer companies will want to sell are those concerned with the supply of software, then the really big markets will be the man-in-the-street or more generally the world of non-computer experts".

1.5.2 Information retrieval

Information retrieval is also a relevant topic since the amount of information available to ordinary people is growing all the time. The task of obtaining a particular piece of information may involve a person in an exhausting search. However, for data held in computer memory, interactive dialogue can provide a means for retrieving highly specific information with the minimum of effort on the receiver's part.

1.6 SUMMARY OF THE EMERGENT PROGRAM OF WORK

Having stated the initial objective of the project, this section describes its path of development and summarises the work which took place.

CHAPTER TWO

The initial step was to survey the problems experienced by naive users of a series of self-explanatory computer games. Since many of the games were found to contain deficiencies which prevented them from being played or understood easily, this confirmed the supposition that interactive dialogues are often poorly designed.

CHAPTER THREE

A preliminary study was then initiated to enable the author to gain a practical appreciation of the problems associated with designing and testing interactive software. One major pitfall to be identified was the tendency to rely upon untested assumptions regarding the acceptability to the user of particular dialogue features. The need was recognised to place greater reliance upon the opinions of actual users when testing dialogues rather than those of computer specialists. Finally, the problem of choosing a basic dialogue technique upon which to base an interactive program was found to be a difficult task.

CHAPTER FOUR

A second pilot study was carried out with the aim of comparing a group of four basic dialogue techniques. To achieve this comparison, it was necessary to create four interactive programs, each embodying one of the chosen techniques but all performing a similar task. It was found that user's preferences for a given technique were influenced more by implementation considerations rather than the techniques themselves.

Therefore, although a fair comparison could not be made, a number of possible improvements were noted which proved to be of value when incorporating the techniques into later dialogues.

CHAPTER FIVE

At this point the focus of the enquiry was narrowed down to consider a specific group of naive users; the general public. The task of information retrieval was also restricted to that of retrieving data from a public information system situated either in a reference library or information centre. It was thought that specifying a particular area of application would increase the validity of any dialogue recommendations produced.

Embarking upon the main study, an interactive system was designed to provide local information for Leicester and Leicestershire. The City's Information Bureau agreed to provide facilities to make this available for direct use by members of the general public.

The system itself (christened ISLA) was designed to provide a variety of information including train timetables, street directions, lists of public houses, hotels and restaurants together with a cinema and theatre guide. Considerable time was devoted towards compiling and classifying this information as well as assembling it onto the computer. Programming the dialogues and access routines was also a lengthy process, the subroutine to provide street directions constituting a problem of much interest.

Although the system was designed to investigate certain aspects of man-machine communication, methods had to be devised of achieving this without destroying its function of being a practical tool. Therefore, as an alternative to posing the user questions such as "What is your opinion of this dialogue?", the system was programmed to provide alternate users with differing dialogues and to record the performances of each. Other information gathering mechanisms were also employed such as collecting written comments (see Appendix II) and the observation of user behaviour.

A full description of the results obtained from this experiment is presented together with a series of inferences which were drawn from them.

CHAPTER SIX

The findings of the main ISLA study were drawn together and moulded into a series of guidelines to assist designers of information systems for the general public. The relevance to this application of the previously described published recommendations (Sections 1.2 and 1.3) were then discussed in the light of the author's experience. This led to the formation of an additional series of dialogue guidelines for public information systems.

However, a survey carried out amongst a group of application programmers revealed that fewer than half would refer to published guidelines when writing interactive programs. The mere compilation of guidelines, therefore, was thought to be an inadequate method for influencing dialogue designers in general.

The next step was to initiate a follow up survey to obtain the reasons for non-referral to published material. An analysis of these reasons resulted in the proposal of a strategy for communicating dialogue design knowledge more effectively. It was thought that the arrangement of design information into classes ranging from general theories to specific techniques would provide the necessary flexibility for application to differing design environments. The strategy also involved the identification of new channels for transmitting good design principles. Two channels discussed were dialogue design experts and general purpose subroutines for inclusion within interactive software. This raised the question of how difficult the creation of these subroutines would be. As a test, it was proposed that an investigation be carried out into the possibility of writing a routine to handle textual inputs.

CHAPTER SEVEN

Having established a case in favour of dialogues embodying textual inputs, a number of problems associated with this approach were identified. An algorithm, called TEX, was then developed for overcoming these problems. The emphasis of the method was placed upon simplicity so that its employment within a textual input handling subroutine could easily be accomplished.

CHAPTER EIGHT

Finally the whole project was reviewed and a series of conclusions formed which in turn provide a basis for further research.

CHAPTER 2

A SURVEY OR PROBLEMS ENCOUNTERED BY NAIVE USERS WITH GAMES PROGRAMS

The object of this survey was to study the problems experienced by naive users of existing interactive software. The results obtained provide an insight into the effectiveness of current program design.

The survey was conducted at a Polytechnic and the material consisted of a series of interactive computer games obtained from various internal and external sources.

These games were considered suitable for naive users because they were generally thought of as being self-explanatory, amusing to play and simple to understand.

2.1 APPROACH OF THE SURVEY

Of the fourteen games available, four were selected (as being of interest) for experimental study. The aim was to record any difficulties encountered by user subjects and to identify the program defects which caused them.

Following this, a complete list of defects was compiled and all fourteen games were inspected to find out which defects they contained. A table summarises the results.

All flaws listed were limited to those of a 'clear cut' nature. In other words, the existence of a program flaw was to be indisputable rather than a matter of opinion. This enabled the survey of the fourteen games to be carried out objectively.

2.2 FOUR GAME EXPERIMENTAL STUDY

2.2.1 Method

Twenty three naive user subjects took part in the experiment. Each received direction in logging onto the computer and use of the carriage return key. They were also supplied with details enabling them to 'run' the four selected programs.

Each subject was asked to limit the time spent on each game to allow all four to be tried. Help was given by the supervisors of the experiment, in the initiation of program runs and in program termination when problems arose. In this way the subjects transferred from one game to the next as smoothly as possible.

Subjects were asked not to communicate with each other. Thus the role of introducing and explaining the games was left with the programs themselves.

Each subject was given a sheet of paper and asked to write down any difficulty experienced with any game together with the last message sent by the computer. This allowed each difficulty to be pinpointed within the appropriate game.

One subject's dialogue was logged and is later used to illustrate certain problems which occurred.

After the experiment, subjects were asked to judge whether each game they tried had been used successfully or unsuccessfully. Overall comments on the games were also requested to supplement the previous information.

2.2.2 Results

The four games programs are now considered in turn. The deficiencies described for each program were all brought to light by subjects comments made during and after the experiment.

(1) Game One - played by 19 subjects.
This is a version of the 'take numbers' game. The user played against the computer, each taking turns to subtract any number from one of a given set of numbers. The last person to subtract, won the game.

The obvious limitation of this game was the lack of any introduction or explanation of the rules. The label "self-explanatory" clearly did not apply, the first message to the user being:

"INPUT MAX ROWS"

Most subjects expressed the need for some kind of explanation. A number of them proceeded with the game by giving random number inputs. This approach did not appear satisfactory as three subjects noted difficulty in getting moves accepted. Rejected inputs produced the messages:

	"ILLEGAL MOVE"
or	"ILLEGAL PARAMETERS"

The program did not explain why inputs were illegal which might have shed light on the game. One subject stated that: "there was no satisfaction when one could not understand the rules of the game and playing it blindly did not help".

In the follow up questions, only two users of the nineteen felt they had used the game successfully.

Another problem arose for three users who received the computer orientated message:

"BASIC ERROR **** NUMBER BADLY FORMED"

at which point the program terminated. This occurred when a numerical input was required by the program and a non-numerical input was supplied.

Understandably this caused bewilderment since the nature of the error was not clearly explained and no opportunity was given to rectify it.

The limitations of this program can be summarised thus; no introduction or explanation was given, inputs were rejected without explanation and program termination occurred when a numeric input was required and not given.

The logged dialogue typifies the problems experienced:

COMPUTER:	"INPUT MAX ROWS"
USER:	"6"
COMPUTER:	"INPUT MAX ROW VALUE"
USER:	"5"
COMPUTER:	"ROW 1 = 1 ROW 2 = 5 ROW 3 = 1 INPUT ROW, NUM"
USER:	"3"
USER:	"4"
COMPUTER:	"ILLEGAL MOVE INPUT ROW, NUM"
USER:	"1"
USER:	"3"
COMPUTER:	"ILLEGAL MOVE INPUT ROW, NUM"
USER:	"5"
USER:	"2"
COMPUTER:	"ILLEGAL MOVE INPUT ROW, NUM"
USER:	"7"
USER:	" "
COMPUTER:	"YOU CONCEDE I WIN"

(2) Game Two - played by 23 subjects.

Based on the television series 'Star Trek', the user was placed in command of the spaceship, 'USS Enterprise'. Given a choice of ten different manoeuvres, the object was to destroy an enemy ship. To command each manoeuvre, the appropriate line number had to be keyed in.

Although the program began by printing out an introductory passage, seven subjects commented that the game was poorly explained and its object was not clear. The subject whose dialogue was recorded was one of this group, the comment being made after the game which involved forty manoeuvres.

The opening passage printed by the program is given below:-

```
WELCOME TO
                         STARTREK
THE USS ENTERPRISE, ON A 5 YEAR MISSION THROUGH THE GALAXY,
IS ON PATROL IN THE ARCTURAS SECTOR NEAR GAMMA-TRISKELLION 7,
WHEN IT IS CONFRONTED BY A KLINGON BATTLE CRUISER!

       WEAPONRY                          RANGE

2 FORWARD PHASER BANKS             3ØØ,ØØØKM
1 REAR PHASER                      3ØØ,ØØØKM
2 FD PHOTON TORPEDO PROJECTORS     2ØØ,ØØØ-6ØØ,ØØØKM
1 REAR PHOTON TORPEDO              2ØØ,ØØØ-6ØØ,ØØØKM

                  MANOEUVRES

Ø.  END PROGRAM
1.  FIRE FORWARD PHASERS
2.  FIRE REAR PHASER
3.  FIRE FORWARD PHOTON TORPEDO
4.  FIRE REAR PHOTON TORPEDO
5.  ACTIVATE AUTO DESTRUCT
6.  SURRENDER
7.  ATTEMPT TO BREAK CONTACT - GO INTO WARP DRIVE
8.  MOVE CLOSER TO KLINGTON
9.  TURN OVER CONTROL TO MR. SPOCK
1Ø. ASK MR. SPOCK FOR A RECOMMENDED MOVE

KLINGON COMING INTO RANGE - SHIELDS ON
THE RANGE =82433. 24Ø496 KM AT A VECTOR HEADING OF 119. 92559Ø82
DEGREES
WHAT IS YOUR MOVE?
```

The terminology used in this passage clearly assumes familiarity with the television series on which the game is based. Despite this, the only indisputable limitation arising from subjects' notes was that the player's aim is not stated, i.e. To destroy the enemy ship.

Eight subjects pointed out the problem that the manoeuvre list scrolled off the VDU screen soon after commencement and could not be retrieved. This left subjects guessing at appropriate moves. The recorded dialogue shows that the option number to end the game had been forgotten.

```
COMPUTER: "IF YOU WANT ANOTHER BATTLE, TYPE 1"

USER:     " "

USER:     "run(ph2al02)bjack"

COMPUTER: "BASIC ERROR*** NUMBER IS BADLY FORMED

          £U-DS @ 0182900, 0204260, 0204578, LINE 0000251,
          0205560

          £ET=10:10.7 PT=1.7 10=0.4"
```

Unable to finish the game by simply pressing carriage return, the subject issued the instruction to commence the next game. The program, still requiring numerical input, terminated after displaying a computer orientated message.

It could be argued that running the program on a hard copy device such as a teletypewriter would eliminate the problem of losing the manoeuvre options. Contrary to this, the inclusion of a command to redisplay the options would be both simple and desirable bearing in mind the widespread use of VDU interfaces.

One comment highlighted the fact that no instructions were given for entering moves. Referring back to the opening introduction; although possible manoeuvres are numbered, the message asking for input does not state that the appropriate number must be entered.

Finally, as the following extract from the recorded dialogue, shows, invalid inputs were rejected without explanation.

```
COMPUTER:  "THE RANGE = 41846. 382315 KM
            WHAT IS YOUR MOVE?"

USER:      "3"

COMPUTER:  "MOVE IMPOSSIBLE -- TRY AGAIN
            WHAT IS YOUR MOVE?"

USER:      "4"

COMPUTER:  "MOVE IMPOSSIBLE -- TRY AGAIN
            WHAT IS YOUR MOVE?
```

To summarise the defects found;

The game's object was not stated, memorisation of move options was needed, no instructions were given for entering inputs, program termination occurred when required numeric input was not supplied and inputs were rejected without explanation.

(3) Game Three - played by 18 subjects.
The computer acted as dealer in the card game 'blackjack'. The user competed with the computer to obtain the better 'hand of cards'.

Few difficulties were recorded by subjects using this game. A partial description of the rules was provided although the object of the game was not included, i.e. To obtain a hand of cards as close to 21 as possible without exceeding it.

The rules, as presented by the program are given below.

```
WE ARE GOING TO PLAY BLACKJACK ACCORDING TO HOYLE.

IF YOU HAVE A PAIR FOR OPENERS (FIRST TWO CARDS) YOU MAY
SPLIT THEM.

IF YOU HAVE FIVE CARDS WITHOUT GOING OVER 21,
YOU WIN.

THE DEALER MUST DRAW IF THE TOTAL
OF HIS CARDS IS LESS THAN 17, UNLESS OF COURSE THE
PLAYER HAS ALREADY GONE OVER 21 OR HAS BLACKJACKED.

YOUR OPENERS ARE KNOWN ONLY TO YOU.

ANSWER QUESTIONS WITH Y FOR YES AND N FOR NO.
ALSO, TO THE QUESTION:
             ANOTHER GAME?

YOU MAY REPLAY:

Y   =   YES
N   =   NO, END THIS SESSION OF BLACKJACK.
S   =   RESHUFFLE THE DECK, PLEASE.
C   =   I WANT TO CHANGE MY BET.
T   =   TOTAL MY WINNINGS, OR LOSSES.
```

Users, having previously played blackjack with 'real' cards were obviously at an advantage. One subject said "the game was easy to follow as I already know how to play it" and another wanted "more explanation for those who don't play cards".

The reason for fourteen of the eighteen subjects declaring their success in playing this game was probably due to the full commentary given as the hands were played. Thus users in doubt could pick up the rules as they went along.

```
COMPUTER:  "YOUR HAND : MY HAND
            Q HEART     *****
            6 SPADE     4 HEART

            ANOTHER CARD?  (Y, OR N.)"

USER:      "Y"

COMPUTER:  "5 CLUB
            MY HOLE CARD IS 2 HEART
            1 DRAW 5 HEART
            1 DRAW A DIAMOND
            1 DRAW 9 HEART
            YOU HAVE 21
            I HAVE 21
            STAND-OFF"
```

Two final observations are worth making in regard to this program:

Firstly, the five commands available to the user (see rules) did not need to be memorised. Before each game, the computer sent a message which included the command options.

i.e.

```
"ANOTHER GAME?  (Y,N,S,C,T OR X FOR INFO.)?"
```

If the command meanings had been forgotten, the input option "X" would redisplay them.

Secondly, a number of subjects expressed particular enjoyment in playing the game. This could perhaps be attributed, not only to the game itself but also to the style of the dialogue. For instance the computer, using the subject's name, occasionally added a remark such as:

"BEGINNERS LUCK, MISS JONES"

This type of message would be unsuitable for more serious applications but provides evidence that naive users appreciate friendliness in dialogue with a computer.

(4) Game Four - played by 13 subjects.
In this game the user assumed control of a ficticious country for a ten year period. His responsibilities covered the production of grain and its consumption by the population. An assessment of his performance was given at the end of the game.

The monitored dialogue is presented overleaf.

COMPUTER: "TRY YOUR HAND AT GOVERNING ANCIENT SUMERIA
SUCCESSFULLY FOR A TEN YEAR TERM OF OFFICE

SIR, I BEG TO REPORT TO YOU
IN YEAR 1, 0 PEOPLE STARVED; 5 CAME TO THE CITY
POPULATION IS NOW 110
THE CITY NOW OWNS 900 ACRES
YOU HARVESTED 3 BUSHELS PER ACRE
RATS ATE 300 BUSHELS
YOU NOW HAVE 2400 BUSHELS IN STORE

LAND IS NOW TRADING AT 21 BUSHELS PER ACRE
HOW MANY ACRES DO YOU WISH TO BUY"

USER: "10000"

COMPUTER: "SIR, THINK AGAIN, YOU HAVE ONLY 2400 BUSHELS OF GRAIN
HOW MANY ACRES DO YOU WISH TO BUY"

USER: "1500"

COMPUTER: "SIR, THINK AGAIN, YOU HAVE ONLY 2400 BUSHELS OF GRAIN
HOW MANY ACRES DO YOU WISH TO BUY"

USER: "80"

COMPUTER: "HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE"

USER: "2000"

COMPUTER: "SIR, THINK AGAIN, YOU HAVE ONLY 720 BUSHELS OF GRAIN
HOW MANY BUSHELS DO YOU WISH TO FEED YOUR PEOPLE"

USER: "20"

COMPUTER: "YOU STARVED 75 PEOPLE IN ONE YEAR ....
YOU HAVE BEEN STONED TO DEATH ....
ITS OVER NOW THOUGH ....
DO YOU WANT ANOTHER GAME Y/N"

It can be seen that no object for the game is declared. A situation is presented to the user but no rules with which to work. Three subjects made comments of this nature. One subject wrote; "There was no indication of how acres bought, bushels used and acres seeded affected the population". They went on to say that moves were largely a matter of guesswork until one could discern input values which would prevent being 'stoned to death'.

One virtue which the example illustrates is that invalid inputs were not rejected out of hand but were accompanied by a message explaining why rejection occurred.

A single comment applying to all four games, was that the layout of the text could be improved. It is possible that other subjects were inhibited by text layout without consciously noticing it.

2.2.3 Synthesized Fault List

The next step was to bring together all the deficiencies found in the four games and compile them into a single list.

This consisted of six items.

(1) No introduction or explanation of the game is given. The user is immediately asked for his first move.

(2) The object of the game is not stated.

(3) The memorisation of alternative moves and their related code numbers is required.

(4) No instructions are given for entering inputs.

(5) Invalid moves are rejected without providing guidance for entering a correct move.

(6) The program terminates when a purely numeric input is required and is not supplied.

2.2.4 Success experienced in playing the four games

The subjects were asked to express success or failure in using each game program. It was stressed that the word "success" implied the ability to play a game rather than to win it.

Presented below are the results of this assessment.

Game	One		Two		Three		Four	
	Success	Failure	Success	Failure	Success	Failure	Success	Failure
Number of Subjects	2	17	13	10	14	4	8	5
Percentage of Subjects	11%	89%	57%	43%	78%	22%	62%	38%

As the table shows, a significant proportion of the subjects expressed failure in using all four games. From the protocol analysis carried out, it would seem that this failure was largely due to the program deficiencies listed. It seems a fair inference therefore, that similar programs which contain a significant proportion of these faults would reflect similar success and failure rates.

All fourteen games were then inspected to find which faults they contained. The results of this survey are described in the next section.

2.3 FOURTEEN GAME SURVEY

While the survey was being carried out, two important deficiencies were identified and added to the list of six faults.

These were:

(7) The program accepts and acts upon invalid inputs. This could cause confusion for the user. One game for instance, required the user to land a space module on the moon's surface. The rate of burning fuel was restricted by the rules to within the range 0-200 pounds per move. However, inputs outside this range were accepted by the program as valid.

(8) The program terminates abruptly for no apparent reason. As an example, one game which simulated a round of golf, finished in the following way:

```
COMPUTER:  "HOLE 1 IS 420 YARDS,  PAR 4
            CHOOSE YOUR CLUB AND SWING"

USER:      "1"

USER:      "9"

COMPUTER:  "WHAT A BANG
           #8797 INV OPERATOR @ 002:0000:0*
           #F-DS @ LINE 00000100.
           #ET=2:51.3 PT=0.6 IO=0.3"
```

2.3.1 Results

The results of the survey are tabulated below, crosses indicating the presence of a particular fault.

FAULT	GAME													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
No introduction or explanation	X					X				X				
Object not stated		X	X	X			X	*	*			*		
Memorisation of move options required		X						X	X			X		
Lack of input instructions		X							X					
Invalid moves rejected without guidance	X	X				X			X			X		
Termination when numeric input required and not given	X	X	X	X	X		X	X	X		X	X		
Acceptance of invalid inputs					X					X	X		X	
Termination for no apparent reason							X	X	X					

* In these cases the object was not declared but seemed self-evident from the nature of the game, i.e. One was the traditional game of noughts and crosses and two others simulated a round of golf.

Note that for games lacking an introduction, no assessment was made regarding statement of object and the presence of input instructions. It was assumed that the inclusion of an introduction would cover these features.

2.3.2 Further dialogue examples

Two examples particularly devoid of clarity when played, are quoted.

Example 1 - A game simulating business activity.

```
COMPUTER: "ORDERS ARE PLACED ON THE FACTORY IN UNITS OF CAPACITY
           RANDOMLY BETWEEN 1 AND 5 UNITS IN SIZE
           DELIVERY DATES ARE AT RANDOM DURING 60 WEEKS
           THERE IS A LINEAR FORECAST GROWTH IN ORDERS
           IN THE RATIO 1 IN WEEK 1 TO 1.67 IN WEEK 60
           ORDERS SHOULD BE PRODUCED IN DELIVERY DATE WEEK
           ORDERS ARE REQUIRED TO BE PRODUCED DURING THE NEXT 60 WEEKS
           DURING WHICH CAPACITY WILL BE INCREASED FROM 60 TO
           HOW MANY ORDERS DO YOU WISH TO LOAD?"

USER:      "5"

COMPUTER: "AFTER 5 ORDERS LOADED

              0       20      40      60      80      100
              I.......I.......I.......I.......I.......I
           1  .*
           2  .*
           3  .*
           4  .*
           5  .*
           6  .*
              .
              .
              .
              .
              .
              .
          57  .*
          58  .*
          59  .*
          60  .*
              I.......I.......I.......I.......I.......I
              0       20      40      60      80      100"
```

At this point the game ended.

Example 2 - A game of three dimensional noughts and crosses.

```
COMPUTER: "WHO GOES FIRST, YOU OR ME?"

USER:     "YOU"

          (No indication was given at this point of
           what should occur next.  It transpired
           that a random number input was required to
           initiate the game).

USER:     "163"

          (From here, all user inputs appear to the
           right of the exclamation marks, the
           remainder of the text being output from the
           computer).

            1. B10      !      Z12
           BAD MOVE.
            1. B10      !      HELP
           BAD MOVE.
            1. B10      !      0
           BAD MOVE.
            1. B10      !      A1
            2. B11      !      S1
           BAD MOVE.
            2. B11      !      D3
            3. C10      !      F1
           BAD MOVE.
            3. C10      !      A2
            4.*D10      !      C3
            5. A10      !

           MY GAME
           B10 D10 A10 C10
           SHALL WE HAVE ANOTHER?
```

2.4 CONCLUSION

The results table shows that the list of eight faults applied widely to the fourteen programs considered. Only one was completely free from these defects. Clearly greater consideration should be devoted towards the man-computer dialogue if interactive software is to be both more effective and appear more attractive to the user.

CHAPTER 3

PILOT STUDY OF THE PROBLEMS CONNECTED WITH DESIGNING AND USING INTERACTIVE SOFTWARE

The survey of games programs in the previous chapter illustrates that poor dialogue design can cause serious hazards for the user, even though the mechanics of the game itself are sound. For example, many of the programs halted without clear explanation when an alphabetic input was supplied in error to a numeric one.

This chapter describes a pilot study carried out to investigate further, the problems experienced by users of interactive software. Since this was based upon software written by the author, it was also possible to examine the process of interactive design from an empirical standpoint and by implication, possible reasons for the existence of poor man-computer dialogues.

3.1 AIMS OF THE PILOT STUDY

To summarise, the broad aims of this initial investigation were:

(1) To study the process of interactive program writing.

(2) To study the problems encountered by users of interactive software.

3.2 DESCRIPTION OF THE ESTATE AGENTS PROGRAM

A program was devised to simulate a facility which would possibly be available for general public use in an estate agents office. A database, containing an imaginary list of houses for sale, was accessed by the program. The customer communicated a description of the type of property he wished to purchase and received via the VDU, a list of those houses meeting his requirements.

The basic dialogue format consisted of a sequence of computer messages, each one informing the user which item he was to specify and providing instructions for making his reply.

The complete specification covered the following attributes: price, type (terraced, semi-detached, or detached), number of bedrooms, number of garages, and distances from the town centre. All of the items, except for house type, required the user to enter numerical values. Each could either take the form of a single number or a numerical range.

3.3 PROGRAM DESIGN CONSIDERATIONS

The programming language BASIC was selected for development of the system, the main advantage being the clarity of input and output instructions,

e.g.

```
PRINT "HELLO, WHAT IS YOUR FIRST NAME?"
INPUT N$
```

This simplicity allowed the program's interactive structure to be displayed clearly and therefore dialogue modifications to be easily made.

3.3.1 Selection of method for user input

Since four out of the five inputs were numeric, this aspect was given the greatest consideration. In the case of house price, the input by the user of his chosen price range appeared at first sight to be a straightforward matter of presenting a menu list of possible options:

e.g.

(1) 0 - 5,000 POUNDS
(2) 5,000 - 10,000 POUNDS
(3) 10,000 - 15,000 POUNDS
....etc.

However, even a sizeable list may not contain an option close enough to his particular requirements (e.g. 9,500 - 14,000 POUNDS).

An alternative method would have been to invite the user to type in the upper and lower limits as separate inputs. However, it was thought that this approach could become quite complex if it were extended to cover single limit inputs (e.g. MORE THAN 5,000 POUNDS) or inputs consisting of single values (e.g. 2 GARAGES).

The technique chosen, was to provide a restricted vocabulary of words so that the specification could be entered as a series of English language statements. The user was given the freedom to construct his replies in any way desired, provided he confined himself to the given vocabulary. The Honolulu Police System (1970) is based upon this approach. Given that the words used were permissible, the following type of enquiry could be entered:

"THOSE PEOPLE WITH A SALARY GREATER THAN 1200"

3.3.2 Familiarising users with the vocabulary

The next stage in program design involved the question of conveying to users the available vocabulary. It was thought that naive users would not wish to undergo any form of training course and in any case this would be impractical for a system used casually by the general public.

Thus the vocabulary had to be displayed to the user at the time of interaction.

The initial idea was to list the vocabulary on a sheet of paper or card and attach it to the terminal for user reference during interaction with the program. However, it was felt that the user could experience difficulties in memorising the allowable words while typing in each specification. This could involve him in a laborious process of frequently switching his attention between the list and the VDU screen.

It was decided to display the vocabulary on the screen immediately prior to the user inputting each of his requirements. While the total vocabulary contained twenty six words, it seemed sensible to restrict the words displayed under each heading to the appropriate subset of the total. Clearly, this would simplify the user's task of scanning each word list in formulating his inputs.

An example extract from the dialogue follows, showing the vocabulary subsets and possible user inputs.

```
COMPUTER:  "GO AHEAD AND DESCRIBE YOUR HOUSE PURCHASE
            REQUIREMENTS"
           "PRICE RANGE"
           ******************************************
           * ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,NINE, *
           * HUNDRED,THOUSAND,MORE,LESS,OR,THAN,                 *
           * BETWEEN,AND,TO,POUNDS                               *
           ******************************************

USER:      "LESS THAN FIVE THOUSAND FIVE HUNDRED POUNDS"

COMPUTER:  "TYPE"
           ********************************
           * SEMI,DETACHED,TERRACED,ANY TYPE,AND,OR *
           ********************************
USER:      "DETACHED OR SEMI"

COMPUTER:  "NUMBER OF BEDROOMS"
           **********************
           * ZERO,ONE............NINE, *
           * MORE,LESS,OR,THAN,        *
           * BETWEEN,AND,TO")          *
           **********************
USER:      "TWO OR THREE"
```

For purposes of consistency, it was considered beneficial to require that numbers be entered as words. Note that the vocabulary restriction of numbers to below ten thousand provided sufficient scope for experimental purposes.

3.3.3 Processing the inputs to the program

The comparatively small vocabulary subset applicable to each attribute greatly simplified the processing of user inputs. A keyword recognition routine was sufficient to handle the 'house type' requirement although a distinction had to be made between "SEMI-DETACHED" AND "SEMI OR DETACHED".

The method for handling a numeric range was a little more complex. Consider the following example:

"MORE THAN TWO LESS THAN FIVE THOUSAND POUNDS"

The first step was to discard "noise" words. This produced the character string:

"MORE TWO LESS FIVE THOUSAND"

Secondly, the construct

"NUMBER * NUMBER THOUSAND"

was recognised and transformed to give

"NUMBER THOUSAND * NUMBER THOUSAND"

where "*" represents any word, in this case, "LESS"

This created the string

"MORE TWO THOUSAND LESS FIVE THOUSAND"

Thirdly, numerical values were put in to replace the corresponding words. This gave:

"MORE 2000 LESS 5000"

This string was then in a form suitable for identifying the price range.

It was realised that a mechanism would be required to handle user errors. If a word was not recognised by the program, a message was displayed.

For example:

```
COMPUTER:  "GREATER IS NOT A LISTED WORD.
            EITHER TYPE IN ANOTHER WORD TO REPLACE IT
            OR TYPE IN "AGAIN" IF YOU WANT TO
            REWRITE THE WHOLE STATEMENT"
```

This facility was designed to enable the user to correct typing or spelling errors without having to re-enter the whole input line.

The lengthy description given to the development process of this program illustrates the kinds of problems which face dialogue designers. The author was surprised by the large amount of time and effort needed to resolve these problems.

3.4 DESCRIPTION OF THE EXPERIMENT

Twenty three student subjects were invited to take part in the experiment. They expressed varying amounts of previous experience in using computer terminals including some with no experience. Each subject was presented with three descriptions of houses as might be expressed by potential house buyers to an estate agent. The attributes covered by these descriptions were equivalent to those incorporated into the estate agents program.

An example is given below:

> "A terraced or semi detached house under £5500 with 1 or 2 bedrooms. One garage or no garage is acceptable and not more than three miles out of town".

The program was set up to accept three house specifications and each subject was asked to interact with the program to enter the three descriptions given to them. Once received, each specification was compared with the database of houses. Those which corresponded were displayed to the subject who recorded their addresses. It was known which houses in the database matched the three descriptions, and so the subject's success rate could be calculated. Finally, a set of statistics was displayed relating to:-

(i) the number of input words which the program failed to recognise.

and (ii) the number of input lines which the program could not understand despite being composed of valid words, e.g. "BETWEEN THREE MILES".

These were recorded by the supervisor of the experiment.

3.5 RESULTS

The following statistics summarise the results of the experiment:

The mean rate of success for the twenty three subjects in obtaining the correct houses was 66%, a fairly moderate score. However, the number of unrecognised input words and misunderstood input strings appeared to be quite low. Thirty two words were not identified by the program out of a total of 355 input strings. This is equivalent to 9% of the input strings containing a single invalid word and the remaining 91% being error free.

Regarding computer failure to understand the input strings themselves, 4% came into this category.

3.6 DISCUSSION OF RESULTS AND SUBJECTS' COMMENTS

While the results themselves reflect the use made of a limited vocabulary dialogue by naive users, it is more interesting to relate them to observations made during the experiment.

Comments made by subjects are also discussed since they throw light on the failings of the dialogue.

3.6.1 Observations from the experiment

Three points arose:

(i) That in general subjects were interested in taking part in the experiment and quickly became involved in their task. Particular attention was devoted to entering inputs correctly which may have contributed to the low number of errors made. The phenomena of good task performance taking place under experimental conditions has been widely recorded. The classic Hawthorne studies (Roethlisberger and Dickson, 1939) illustrate this most impressively.

(ii) A number of subjects failed to use the limit setting words, "MORE" and "LESS", correctly. For example, a descriptive statement such as "not less than 2 bedrooms", would cause the user to type in the number as,

"MORE THAN TWO"

rather than

"TWO OR MORE"

Thus two bedroomed houses in the database would be wrongly excluded from the list of houses retrieved. This type of error could have been due to the user misinterpreting the descriptive statement. Alternatively, the user may not have been able to think of the "TWO OR MORE" construction and so typed in "MORE THAN TWO" as a final resort. If the latter explanation applies then it would indicate that the choice of vocabulary words or the order in which they are displayed should be reconsidered.

(iii) Subjects with previous experience of using computer terminals were noticeably more confident than those without such experience. Also they displayed greater adeptness when using the keyboard.

3.6.2 Subjects comments

The following comments were expressed with regard to the interactive dialogue.

(i) The most widely held view was that numbers should be input in digit form rather than word form. The latter approach was considered to be both unnatural and tedious.

(ii) Some users found it difficult to avoid the words they normally used in speech and to restrict themselves to the vocabulary displayed. For example, "UNDER" was used instead of "LESS THAN". Unfortunately, a count was not kept of the frequency of this type of error.

(iii) One subject related that, on being informed by the computer of having typed in an invalid word, he then wished to replace it with two words. Since the program's editing facility only allowed single word replacement, this forced him to re-enter the whole line.

3.7 CONCLUSIONS

This pilot study has illustrated that the design of man-computer dialogues poses more problems than may at first be apparent.

Selecting an appropriate dialogue technique is by no means a straightforward decision. The technique of limited vocabulary input chosen for this study was shown to provide greater flexibility of input than does a simple menu list system.

A second problem concerns the writing of the software to support the chosen dialogue. It appears that the designer can easily make the wrong assumptions regarding what would be most acceptable to the user. An example was the assumption that to be consistent with the concept of a vocabulary of words, numbers should be entered in word form. However, users commented that this approach was both unnatural and tiresome.

Thirdly, careful design of trials for testing dialogues is needed. It should be borne in mind that the novelty associated with experimental conditions will tend to create artificially high levels of user performance. Also dialogues designed for use by people with no experience of computer terminals should be tested by such people rather than by users who are merely unfamiliar with that particular program.

A final comment concerns the design of limited vocabulary dialogues in particular.

The displaying of appropriate subsets of the total vocabulary prior to each input seems to offer a solution to the problem of conveying the words which the operator may use. However, care needs to be taken with the selection of these words and the format for displaying them on the VDU screen. Failure to do this may prevent users from expressing their intended inputs accurately.

CHAPTER 4

PILOT STUDY TO COMPARE FOUR MAN-COMPUTER DIALOGUES TECHNIQUES

The pilot study of the previous chapter identified certain problems in the construction, testing and usage of a particular man-computer dialogue. It was decided that the investigation should be broadened to compare a number of dialogue techniques. The results of this study would allow these techniques to be put into order of acceptability for naive users.

4.1 AIMS OF THE PILOT STUDY

(1) To compare four techniques of man-computer dialogue for naive users.

(2) To record the features of each technique which presented problems to users and to recommend modifications for their improvement.

(3) To gain a clearer idea of the characteristics of naive users in order to assist the design of dialogues for them.

4.2 DESCRIPTION OF THE PILOT STUDY

A set of portrait photographs were collected from magazines and mounted. Each was assigned a randomly chosen name and height and labelled accordingly. The name, facial features and height for each portrait were then stored in a file held in computer memory.

Each subject taking part in the experimental study was asked to select a portrait and communicate his description of it to the computer. This description was compared with the stored file of portrait details to find those providing the closest match. Finally, the names of the matching portraits were displayed.

```
e.g.
    COMPUTER:  "THE PORTRAITS OF
                JOHN, ROBERT, JAMES
                MOST CLOSELY FIT YOUR DESCRIPTION"
```

Thus the subject's aim was to receive the name of the portrait which he had chosen and described.

Four interactive programs were developed for independent use in performing the above task. Each was based upon one of the four dialogue techniques being compared.

4.2.1 The four dialogue techniques

A brief description now follows of the four dialogue techniques, illustrated with examples of interaction.

(1) Tertiary input dialogue

This dialogue presented the user with a complete list of facial and height attributes. For each he responded by pressing one of the three keys 'Y', 'N' or 'D'. 'Y' indicated that the attribute applied to his particular portrait, while 'N' meant that it did not.

If the user was unsure, then he pressed 'D' for 'don't know'.

e.g.

```
COMPUTER:  "BLUE EYES"
USER:      "Y"
COMPUTER:  "BROWN EYES"
USER:      "N"
COMPUTER:  "GREY EYES"
USER:      "D"
```

This simple dialogue was used very successfully for interviewing hospital out-patients before seeing a doctor. (Evans, 1970)

(2) Menu list input

Here the attributes were grouped under appropriate headings such as eye colour, hair colour and hair length and presented to the user as a series of numbered menu lists. The user responded to each list by typing in the numbers of those attributes which applied to the portrait. Additionally he could deem an attribute as not applicable by typing the letter 'X' immediately prior to its number. Other attributes of which the user was unsure were to be omitted from his specification.

e.g.

```
COMPUTER:  CONSIDERING COLOUR OF EYES
           WHICH ONE OR MORE DO YOU WISH
           TO SPECIFY?
      (1)  BLUE EYES
      (2)  BROWN EYES
      (3)  GREY EYES
      (4)  HAZEL EYES
      (5)  "I DON'T WISH TO SPECIFY EYE COLOUR"
USER:      "2  X1  X3"
```

In this example the user described the picture having brown eyes, excluded blue and grey and was unsure about hazel.

(3) Restricted vocabulary dialogue

This dialogue required the user to describe the portrait using a list of words supplied by the computer. It was designed along similar lines to the estate agents program of Chapter 2. The same principle was applied of displaying relevant subsets of the total vocabulary under each attribute heading. However, numbers could be typed in digit form rather than in words.

An informal study was undertaken to determine an acceptable layout for the vocabulary. It was found that users preferred words of the same class to be grouped together rather than a simple alphabetical list. This conforms to a recommendation by Stewart (1979) who says that "grouping similar items in a display improves their readability". It was also thought that synonyms should not be included within the list as they were unnecessary. These findings were borne in mind when developing the dialogue.

e.g.

```
COMPUTER:  "PLEASE SPECIFY HAIR COLOUR, STYLE AND
            LENGTH USING THE WORDS PRESENTED BELOW.
            LIGHT    DARK      COLOUR      HAIR
            BLOND    RED       BROWN       BLACK
            CURLY    WAVY      STRAIGHT    STYLE
            SHORT    MEDIUM    LONG        LENGTH
            ANY   AND   OR   BUT   NOT"

USER:      "LIGHT AND DARK BROWN STRAIGHT SHORT"
```

To fully comprehend the input, certain transformation rules were required. In the above example, the string "LIGHT AND DARK BROWN" would be converted to "LIGHT BROWN AND DARK BROWN".

(4) Unrestricted vocabulary dialogue

This program presented the main attribute headings to the user and invited him to describe the portrait in his own words. The processing needed to understand inputs was similar to that for the restricted vocabulary dialogue. However, a list of synonyms was required to convert the descriptive words of the user into those used in the file of portrait descriptions. For instance, a number of portraits were listed in the file as having 'smiling' expressions. Thus when the words: 'joyful', 'cheerful', 'laughing', 'carefree', 'happy', or 'pleasant' appeared in inputs, they would be converted to 'smiling'.

For the input of height, this could be accepted from the user in any form (e.g. 5 FOOT 4 INCHES, 6 FT, 5'2"). A listing of the routine for handling these inputs is presented in Appendix I. This version has been extended to accept height ranges as well as single heights (e.g. 6 FT 2 IN - 6 FT 4 IN).

If the program failed to interpret a line of description, then a system of back up was put into effect. The computer displayed a full list of available options and requested the user to rephrase his input.

4.2.2 Experimental design

The study was carried out in a room attached to a polytechnic library. Ninety six people were invited to take part, consisting of students and library staff. To gain a representative sample, subjects were selected to give an equal balance of the sexes. They were also split up into equal groups of 'naive' and 'experienced' users. The former group had no previous experience with a computer terminal, while members of the latter group had used a terminal at least once before. It was intended that a comparison of the performance of both groups would highlight particular problems faced by naive users.

The task for each subject was to select one of the portrait photographs and to communicate a description of it to the computer via one of the four dialogues. The computer compared these details with its file of portrait descriptions and displayed the names of those providing the closest match. The subject then selected another portrait and the procedure was repeated for a second dialogue. The twelve permutations of two from four dialogues were tested on equal numbers of male and female, and naive and experienced users.

Finally, to receive feedback from the subject, he was posed the following questions:-

(a) Did you find one of the two dialogues easier to use? If so, which and why?

(b) Did you consider either dialogue more effective in allowing you to describe the portraits?

(c) Can you think of any improvements which could be made to either dialogue?

Additional comments from the subject were also noted together with a record of the computer interaction which took place.

In the previous study, subjects were invited to comment on a single dialogue. By asking subjects to compare two dialogues it was thought that they could make clearer comments on the strengths and weaknesses of dialogue design.

4.3 RESULTS

A scoring method was adopted to determine the over all preferences of subjects regarding 'ease of use' and 'effectiveness' of the four dialogues. Within the process of comparing dialogues in pairs, each one was tested by twenty four naive users and twenty four experienced users. As a result of each comparison, the dialogue judged as the 'easier to use' was awarded one point. When the two were thought equal, each received half a point. Thus the maximum score possible awarded by either the naive group or the experienced group was twenty four.

A similar method was adopted for 'effectiveness of use'.

The total scores are tabulated within the following two subsections.

4.3.1 Ease of Use

SCORES (OUT OF 24)		
	Naive Subjects	Experienced Subjects
Tertiary	16.5	17.5
Menu List	14	12
Restricted vocabulary	12.5	9.5
Unrestricted vocabulary	5	9

The scores appear to show that the dialogues allowing least freedom of input were regarded as the easiest to use. At one extreme the tertiary dialogue, (based upon 'yes', 'no' and 'don't know' responses) scored a combined total of thirty four out of forty eight. At the other extreme the unrestricted vocabulary dialogue scored fourteen.
Statistical analysis using a chi-square test shows that these combined values are significantly different at the 5% level.

The main reasons given by subjects for their preferences are now discussed:

The task of typing in option numbers with the menu list dialogue was thought by many to be easier than entering descriptions in English. However, a number of subjects stated that they failed to understand the method of excluding options, or 'X' command, and in fact 50% of the naive users ignored the facility. It was suggested that the word 'NOT', rather than 'X' might have been a better choice.

With the two natural English dialogues, users were unsure of what constituted acceptable inputs. Some thought the computer would reject single words or phrases and only accept whole sentences. A computer message of clarification may have alleviated this uncertainty.

Concerning unrestricted inputs, subjects felt dissatisfied when, having considered and entered an input, the computer failed to understand it. The user was then presented with a list of options and asked to try again. This situation would have occurred less frequently if a more comprehensive dictionary of synonyms had been available.

4.3.2 Effectiveness of Use

SCORES (OUT OF 24)		
	Naive Subjects	Experienced Subjects
Tertiary	12.5	10.5
Menu List	10	12.5
Restricted vocabulary	11	16.5
Unrestricted vocabulary	14.5	8.5

Applying a chi-square test to the scores for effectiveness, no significant preference was shown towards any of the four dialogues. The deficiencies thought to detract from the effectiveness of each dialogue are described below:-

The tertiary dialogue required the user to respond to each attribute as it was presented. This caused some subjects to reject every attribute under a given heading (such as eye colour) because particular choices they had in mind were not displayed. Were the whole list to have been displayed before requiring any user response, then subjects could have based their selections on a full knowledge of the options available. For example, if a subject had wanted to specify an unavailable option such as 'green eyes' he would choose the closest alternative colour; hazel.

It has already been stated that the 'X' facility within the menu list dialogue was poorly understood. This was felt by subjects to reduce the effectiveness of the menu technique.

The restricted vocabulary dialogue was thought to be less descriptive than other dialogues. For instance, a number of subjects failed to appreciate the value of the word 'not'. Perhaps this problem could have been solved by including example inputs with the vocabulary list, such as 'LIGHT BUT NOT BLONDE'. However, subjects felt that the flexibility to use words in any order was a distinct advantage.

The unrestricted vocabulary dialogue was clearly regarded as more effective by naive subjects than by experienced subjects. Naive subjects frequently commented that the freedom to enter any input reflected considerable power in the dialogue. However, the experienced subjects were probably more aware that the dialogue was not capable of fully understanding complex descriptions such as 'LONG HAIR PARTED IN THE MIDDLE WITH FLECKS OF BLACK BUT GENERALLY RED'. Understandably, these subjects felt that the other three techniques which specified the available options were more effective.

4.3.3 General observations regarding users

Many of the naive subjects displayed nervousness while entering their first few inputs and wanted reassurance from the experimenter that they were not going to "break the computer". Evans (1976) noticed a similar phenomena with the hospital out-patients interviewing system. As with Evans' system it was found that having embarked on interaction, the user was happy to continue.

Subjects were daunted by lengthy instructions appearing on the screen. This was mixed with a desire to begin the task as quickly as possible. The English language dialogues, which required the most typing, frustrated some people who wanted to dispatch their inputs quickly to observe the computer response. This desire for simplicity of instructions and method of input was reflected in a number of comments that a given dialogue was "so simple to use".

When the computer, in the middle of a normal interactive sequence, suddenly gave a slow response, this provided further cause of concern for naive users in particular. Stewart (1979) makes the statement that "variability in delay is more disruptive than the length of delay".

A fairly common input mistake made by naive users was to omit separating commas or spaces between numbers and words. For example, the three numbers three, four, five, would be entered as the single number, '345'. Presumably the user had equated typing with human conversation where linguistic clues such as intonation, pauses and context serve to separate individual words.

4.4 CONCLUSIONS

One of the aims of the study was to place the four dialogue techniques into a preferred order. However, the preferences and comments made by subjects were more related to the particular implementations than to the techniques themselves. A completely different set of preference values could perhaps be obtained if certain modifications were made to the dialogues. On this basis a fair comparison of the four dialogues cannot be made. A more cogent approach would be to compare differing versions of a particular technique with the aim of perfecting it. For example, Schneiderman (1979) reports that:

> "one informal study showed that users preferred three separate menu lists rather than three menus on the screen at once. Although more typing and more interactions were required for the three separate menus, the user preferred doing one small thing at a time".

Certain design faults were identified within the dialogue programs and a summary of the recommended improvements now follows: The tertiary dialogue required the user to respond to each option as it was presented. However, it is necessary to display the full list of options before the user makes a response so that he can plan his selections more effectively. The menu list technique allowed users to specifically exclude options. Subjects felt that it was more natural to imply the exclusion of an option by not selecting it. The two dialogues which allowed natural English input left some users unsure how to proceed. One answer would be to provide guidance by displaying example inputs. The main flaw with the unrestricted vocabulary dialogue was that it failed to comprehend many reasonable inputs. Practical application of this technique will only be acceptable then if it is supported by a sophisticated level of machine understanding.

By observing the reactions of subjects it was concluded that dialogues for naive users should be both easy to understand and should allow the user to commence interaction quickly.

Regarding the pilot study itself, it was realised that the experiment had three artificial conditions imposed upon it. Firstly, all the subjects were invited to take part in the study. This may have increased their motivation to excel in the task they were asked to perform. Secondly, although the subjects were interacting with information retrieval programs, the information being provided was not of personal interest to them. Finally, the sample of naive users consisting of students and library staff was clearly unrepresentative of the general population of naive users. Therefore to improve the environment for the further testing of man-computer dialogues, it was decided to negate these three conditions.

4.5 SUPPLEMENTARY STUDY TO COMPARE DIRECT AND INDIRECT COMPUTER USE

The four dialogue techniques studied all share the common attribute that they require direct user-computer interaction. In a supplementary study the tertiary dialogue was compared with two further techniques based upon indirect contact with a computer.

With one dialogue, the user communicated the attributes he wished to specify to a human intermediary who entered them into the computer. The other dialogue required the user to record his attribute specification by filling in a form. An experienced terminal user then transferred these choices to the computer. Subjects taking part were asked to use all three dialogue techniques for the portrait description task.

No clear preference was shown for any of the three dialogues but the comments made by subjects highlighted the strengths and weaknesses of each approach. The advantage of having a human intermediary was that it enabled the user to discuss his selections in a familiar way with another person. However, there was the danger that his choices might be misinterpreted by the intermediary. The form filling technique allowed the user to see all the available choices simultaneously although it was felt that choices could easily be omitted by mistake. Some subjects stated that the tertiary dialogue put them under less pressure and allowed clearer thinking than the other two dialogues. It was also thought that direct computer contact made this task the most enjoyable of the three to perform.

CHAPTER 5

AN INVESTIGATION INTO THE USE MADE BY THE GENERAL PUBLIC OF A COMPUTER ACCESS SYSTEM PROVIDING LOCAL INFORMATION

Following completion of two pilot studies under controlled experimental conditions, the main investigation was carried out within a real world environment. For this purpose an information system was developed for direct use by the general public. This provided information of interest to people in Leicester and Leicestershire and was installed for a period of one month in the local Information Bureau. Access to the system was provided by a VDU/full keyboard terminal (Hazeltine 1400) linked to a mainframe computer at Leicester Polytechnic (B6700) via the G.P.O. telephone system.

5.1 AIMS OF THE INVESTIGATION

The over-all aim of the investigation was to determine how successful members of the public were in using a self-explanatory information system. The program was also designed to allow various tests regarding certain aspects of man-computer interaction to be carried out. However, it was envisaged that unforeseen behaviour patterns on behalf of users would also be observed. From the results of testsand observations it was hoped to formulate a series of guidelines which would be useful to designers of similar programs for the general public.

5.2 DESCRIPTION OF THE INVESTIGATION

The system, christened ISLA (Information System for the Leicester Area), was embodied in a program which called eleven main subroutines. Each provided information on a particular topic.

These were:

(1) Train information at Leicester Station.
(2) Films, plays and pantomimes currently showing.
(3) City centre street directions.
(4) Suggestions for an evening out.
(5) Public houses, hotels, restaurants and discos.
(6) A walking tour of the City.
(7) Historical visits in and around Leicester.
(8) The weather forecast.
(9) Leicester City Football.
(10) Famous people of Leicester - a quiz.
(11) Pantomimes outside the City.

A record of the use which was made of the system was kept in two files. One stored a series of counts for the number of times each topic was looked at, thus giving an indication of where people's interests lay. The other file monitored interaction with the computer, recording users' enquiries, strategies and decisions. Most of the results obtained from the investigation were based upon this data.

Programming was carried out in an extended version of the standard FORTRAN language. This was preferred to the previously used BASIC language which did not contain the facilities necessary for the creation of a large interactive program.

It was decided that the strongest test of the system would be obtained by allowing people to use it on a voluntary basis. A notice was erected above the terminal informing people that the system was of an experimental nature and designed to be self-explanatory. They were asked therefore, not to seek help from the staff of the Information Bureau.

5.2.1 Barriers to general public use of the information system

It is worth discussing briefly the technical barriers which it was thought could discourage use of the system and the solutions adopted to remove them. A meeting with one of the Information Centre Staff prior to installing the system was of value in recognising these barriers.

Previous experiments had shown that naive terminal users frequently forget to complete their inputs by pressing the 'return' key. No completely satisfactory solution to this was found, but a notice of reminder was placed near the terminal. Since users are known to lack confidence when they first interact with a terminal, the program provided a period of initiation. The first message already present on the screen gave an abbreviated list of the subject topics available. The user was then asked to press 'return' to begin. There followed a shorter message highlighting the 'rub out' key. The user then typed 'OK' and pressed 'return' to receive the full list of topics. Having selected a topic, the user entered the appropriate number and continued. This procedure, though perhaps more extensive than necessary, was designed to build up user confidence and practice with the 'return' key.

The possibility of users abandoning the dialogue without signing off correctly was also realised. One solution considered was to have a timed out resetting facility so that when a certain period had elapsed without user input, the system would automatically reset itself to the introductory message. Gaines and Facey (1975) found that a timed out 'help' message could cause annoyance to users who wanted to consider their responses slowly and carefully. Clearly such a user would be even more distressed if in the middle of the dialogue the computer ended it abruptly. The solution adopted was to convert one of the terminal keys into a 'reset button'. A notice attached to the keyboard informed the user that if the introductory message ("Hello this system is called ISLA") was not present on the screen, then the reset button should be pressed.

5.2.2 Obtaining feedback from users

During the initial stages of running the system, a number of differing methods were tested for obtaining user feedback. Firstly, discreet observation of people using the terminal was carried out. However, users soon became aware of the presence of an onlooker and appeared more anxious as a result. This may have been due to a fear of making mistakes and of 'breaking the system'. A second approach was to include a message at the end of the dialogue inviting users to fill in a questionnaire sheet located near to the terminal. The questions related to their use of the system and which parts they felt could be improved. However, over the period of a week no questionnaires were completed. The method finally used, was to allow members of the general public to express any remarks they had of the system in a 'comments' book. This system proved to be quite effective and a fairly large number of comments were obtained. As a supplement, during the second half of the experimental period, the user was asked at the end of the dialogue session about his intentions prior to using the system. The response to this question is presented in the results section of this chapter. A selection of the comments are reproduced in Appendix II.

The ISLA system installed at the Information Bureau in Leicester.

The following three sections will be concerned with results obtained from the investigation and the discussion thereof. Section 5.3 describes the overall use made of the system and the general attitude displayed towards it as informally observed by the Information Staff.

In Section 5.4 the results of particular tests incorporated into the design of the system are presented.

Finally, Section 5.5 discusses a series of observations which became apparent from a study of the interactive log.

5.3 OVERALL USE OF THE SYSTEM

The information system was available for 29 days and accessed by 316 visitors to the Information Centre. A breakdown of the numbers using each topic is given below:

Topic	Number of Users
Train information	67
Films, plays, pantomimes	64
Street directions	52
Evening out suggestions	65
Pubs, hotels, restaurants, discos	66
City walking tour	19
Historical visits	27
Weather forecast	61
City football	36
Famous people - quiz	38
Pantomimes outside Leicester	8

The five most popular topics were those providing information on trains, films and plays, evening out suggestions, public houses (etc), and the weather forecast. Three possible distinguishing features are given to explain this grouping.

Firstly, it could be said that they are all topics of interest to most people. On the other hand topics such as walking tour, historical visits and local football are more likely to appeal to a minority of people.

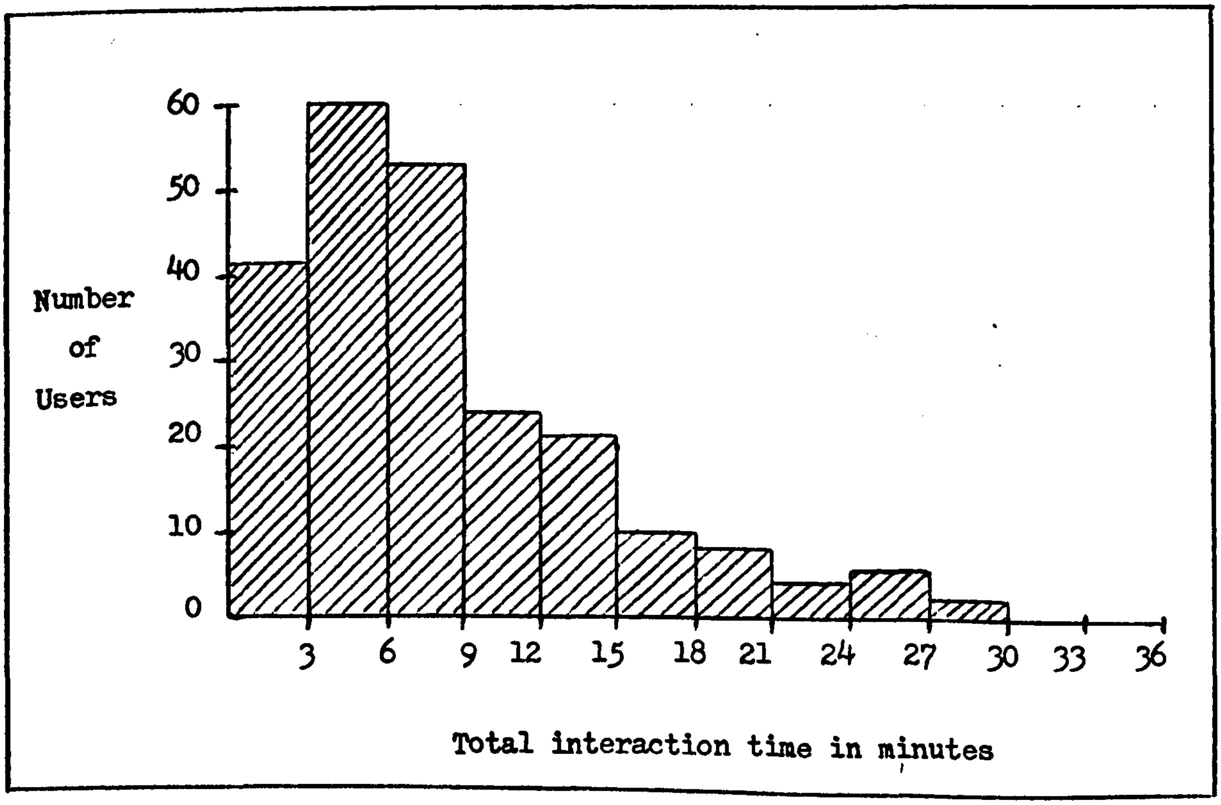
Secondly, to some extent, the popular topics represent rapidly changing information which therefore demand greater attention. This is particularly true of the evening out suggestions and weather forecast which were updated every day, and the films and plays which were updated once a week. Also the experiment took place during the period that rail fares were increased.

A third possible factor relates to the fact that four out of five topics were amongst the first topics listed. Since people generally scan lists downwards, those near the top were considered first. It might be thought that users would give due consideration to all topic titles before selecting one. However, discussion appears to show that this is not the case. A similar phenomena was reported at the National Gallery in the Sunday Times (1980). When one painting was moved from the middle of a long wall where it was over-shadowed by other paintings, to an end wall on its own, sales of the postcard reproduction increased by 20%.

5.3.1 Time spent using the system

Users devoted on average 8 minutes 20 seconds to interacting with the system.

The histogram presented below of total interaction times shows that the majority (66%) were prepared to spend less than 9 minutes at the terminal.



This fact could be seen as a reminder to designers of publicly usable systems, not to make them over complex and difficult to operate. Within the space of a few minutes, it should be possible for the user to learn to operate the

the terminal, specify his enquiry and receive the appropriate information.

5.3.2 The reset key

The implementation of a key to reset uncompleted dialogue sessions has been described. The log showed that 49 conversations ended in this way, requiring the next user to reset the system. Before expressing this as a percentage figure, it should be noted that 35 of the total of 315 sessions were interrupted by either computer or program failure or normal system closure. Discounting these cases, 17.5% of conversations were abandoned. This significant percentage indicates that the reset button performed a useful role.

5.3.3 The transfer rate

Comments entered in the book were of a highly varied nature. One criticism that frequently arose was the speed of the display, set to 300 baud (37.5 characters per second), which was too slow for many users. Unfortunately, the rate of information transfer via the G.P.O. link was limited to this maximum.

This appears to conflict with Evans (1970) finding that users of a hospital outpatients screening system did not object to the slower teletype speed of 110 baud (13.75 characters per second). Possibly the cause of these different findings is rooted in the nature of the two systems. The interrogative nature of computer screening requires the user to read each question carefully, while the ISLA system which essentially displays pages of information, allows the user to omit certain parts at will. Another reason is that in recent years people have become conditioned to instant televisual displays as provided by the teletext services and computer games. In comparison the display rate of ISLA would have seemed very sluggish.

5.3.4 Breakdown messages

A number of people attracted by publicity notices in the local newspaper and on local radio visited the Information Centre with the intention of using the system. Unfortunately, due to computer breakdown, this was not always possible which caused both critical and despondent comments to be made. As breakdowns could occur at any time, the action of the Bureau Staff in erecting a notice of apology could not be relied upon. Thus potential users were faced with an array of meaningless random characters on the screen. The lesson of this experience was that the computer should generate fault messages automatically, to ensure that some kind of explanation is provided during breakdown periods. In a similar way to the MINNIE system, which informs the operator how long he must wait for a computer response (Spence 1976), ideally the breakdown message should indicate when the service would resume.

5.3.5 The system as a browsing facility

During the investigation period a questionnaire was incorporated into the ISLA system. Before coming to the end of a session, each person was asked about his intentions prior to using the system. This was worded as follows:

"DID YOU USE THE SYSTEM PURELY OUT OF INTEREST OR WAS THERE A PIECE OF INFORMATION THAT YOU PARTICULARLY REQUIRED?"

Of the 117 people questioned, 72 stated that they had used the system out of interest, while the remaining 45 had required certain information in advance. As one would perhaps have expected, most people simply noticed the terminal when visiting the Information Bureau and decided to investigate it.

A follow-up question was posed to the group of 72 investigative users asking if they had found any information which they considered of value. The greater proportion - 54 admitted that such information had been discovered. This sub-group which represents nearly half of the total sample can be said to have successfully used the system as a browsing facility.

This is an area where the terminal enjoys an advantage over the human information provider. Whereas the computer system will allow the user to uncover various types of information via many diverse paths and at his own speed, the human is unlikely to be able to provide a similar service.

5.3.6 The introduction and reluctance to use the system

It was observed by one of the Information Officers that some users, particularly within the older age groups, appeared very wary of the terminal and were reluctant to use it. Many spent some time (presumably) contemplating taking part and then walked away. As was seen in the pilot studies, those who did use the system rapidly gained in confidence, and as the list of comments shows, by the end of the session found themselves in a position to suggest improvements. The introductory messages and dialogues should therefore be regarded as an important determinant of whether a system for the public will be used or ignored.

Consider the introductory sequence of the ISLA system.

HELLO MY NAME IS

```
***   ****   *          *
 *   *       *         * *
 *    ****   *        *   *
 *        *  *       * *** *
***   ****   *****  *       *
```

(INFORMATION SYSTEM FOR THE LEICESTER AREA)

IT IS EASY TO USE - JUST FOLLOW THE INSTRUCTIONS

INFORMATION AVAILABLE :
TRAINS, FILMS & PLAYS, PANTOMIMES, PUBS, RESTAURANTS, DISCOS.
HOTELS, WEATHER, FOOTBALL, HISTORICAL VISITS & STREET DIRECTIONS

TO BEGIN, PRESS THE BUTTON MARKED <RETURN>

(new page)

BEFORE WE START, PLEASE NOTE :

EVERY MESSAGE YOU TYPE IN MUST BE FOLLOWED
BY PRESSING THE <RETURN> BUTTON

THIS SENDS THE MESSAGE TO THE COMPUTER

ALSO IF YOU MAKE A TYPING ERROR, PRESS <RUB OUT>
YOU MUST THEN TYPE THE WHOLE LINE AGAIN

WHEN YOU HAVE READ THIS, TYPE "OK" AND PRESS <RETURN>

(new page)

TYPE IN THE NUMBER OF THE TOPIC
YOU ARE INTERESTED IN AND PRESS <RETURN>

(1) TRAIN INFORMATION AT LEICESTER STATION
(2) FILMS, PLAYS AND PANTOMIMES SHOWING
(3) CITY CENTRE STREET DIRECTIONS
.
.
.

Some users of the system felt that this dialogue was too loquacious and should have begun at the stage where the topic list was displayed. (The notes regarding the 'return' and 'rub out' keys would have to be incorporated at a later stage).

In addition to shortening the introductory process, this modification may have produced a further benefit; Although the initial messages were intended to usher the user gently into the system, this may have led some people to believe that the system would be difficult to use and that a long series of instructions or rules would have to be learned. On the other hand, immediate presentation of the less formal topics list may have been more attractive to the user. Furthermore, the simple instruction to type in the number of the topic required, implies that the subsequent instructions will also be simple.

While the above comments are no more than supposition, one can conclude that the introductory dialogue plays an important role and should be designed with care.

5.4 RESULTS OF PARTICULAR TESTS INCORPORATED INTO THE DESIGN OF THE SYSTEM

A number of special facilities and contrasting dialogue styles were built into the system to enable various aspects of man-computer interaction to be investigated. The interactive log provided data to support this work. In the following sections the results of tests relating to those aspects under study will be described.

5.4.1 Short cuts within menu lists

The subroutine entitled 'train information at Leicester Station' supplied details of train times and fares between Leicester and 45 main line stations. Also available were two display screens advertising bargain excursions and explaining the conditions attached to the various kinds of return ticket. The user guided himself through the subroutine by choosing options from menu lists of possible actions. These lists offered possible short cuts. For example, having printed a list of train times to Derby, the following options were displayed:

(1) LOOK UP THE TIMES FOR A DIFFERENT DAY
(2) FIND OUT ABOUT TRAINS RETURNING FROM DERBY
(3) LOOK UP THE FARE TO DERBY
(4) SUBMIT A NEW ENQUIRY ABOUT TRAINS
(5) RETURN TO THE MAIN LIST OF TOPICS

The first three options attempt to anticipate the user's next question since he is quite likely to require further information about the journey to Derby. Without these options the user would be faced with a long path of interaction via option (4) to submit a new enquiry about trains. The interactive log showed that of the 42 users who were in a position to select short cut options, 23 (or 54.8%) made use of them. This indicates that these short cut options served reasonably well to anticipate users' questions and can be regarded as a useful facility to include within similar types of dialogue.

5.4.2 The occurence of errors within input commands

Within the train information section the user specified his required destination by entering it as a command. Two further commands were required to specify the direction and day of travel. In these latter cases the user was told to type either "TO" or "FROM" and either "SAT", "SUN", or "WEEK".

Similarly, the topic 'pubs, hotels, restaurants, and discos' provided details of a particular establishment upon receipt of the appropriate name typed in as a command.

Both of these subroutines incorporated error checking procedures to detect single spelling or typing errors within commands. These consisted of a single letter being either omitted, extraneously added or mis-typed. Also included was the case of two adjacent letters being swopped. It was of interest to discover whether this facility had been adequate in coping with the mistakes which occurred.

Presented below is a table of the error rates:

COMMAND TYPE	NUMBER OF COMMANDS INPUT	NUMBER OF COMMANDS IN ERROR
Station name	66	3
Direction of travel	43	1
Day of travel	52	0
Pub, hotel, etc.	61	7
TOTAL	222	11

The total error rate of 5% appears low for a sample of inexperienced terminal users. This is partly due to the fact that the commands for direction and day of travel were both short and simple which made them easy to enter (i.e. "TO", "FROM", "SAT", "SUN", "WEEK"). The table reflects this as only one such command was mis-typed. The longer commands for stations, public houses, and other establishments provided more difficulty. However, it is probable that users formed queries around places and names which were familiar to them and which, therefore, they could spell.

Of the eleven incorrect commands, nine contained single errors and were thus successfully recognised by the system. The remaining two were identified by presenting a list of the closest matching commands to the user for clarification.

These results seem to show that there may be no need to create over-sophisticated typing and spelling error detecting mechanisms. Perhaps the correction of single errors together with some kind of back-up system when no matches are found, would be sufficient.

Deeper issues concerning the recognition of textual command inputs were later pursued as a separate branch of the project. A report which covers this research is presented in Chapter 7.

5.4.3 Users' ability to memorise commands

One of the main disadvantages for the naive user is that he is unfamiliar with computer commands. The experienced operator on the other hand, having used a set of commands, will find that he has soon committed them to memory.

In an information system where each command may retrieve a screen page of information, the solution might be to re-display the set of commands between pages. However, where the transfer rate is low (as with this system), constant re-showing of the command list would slow down the interaction considerably. Therefore, where the full list of commands was small, it seemed desirable to investigate the ability of naive users to remember those required in advance of their input.

This was carried out within the subroutines on films and plays providing details of entertainment currently available at the City's six main cinemas and theatres.

The user typed in one of six 2 or 3 letter commands to obtain information about each. There were two further commands for returning to the main list of topics, or to sign off. The full list is shown:

ABC	:	FILMS SHOWING AT THE ABC CINEMA
CC	:	FILMS AT THE CINECENTA
ODN	:	FILMS AT THE ODEON
HYM	:	PRODUCTIONS AT THE HAYMARKET
LT	:	PRODUCTIONS AT THE LITTLE THEATRE
PHX	:	PRODUCTIONS AT THE PHOENIX
TOP	:	I WOULD LIKE TO RETURN TO THE MAIN LIST OF TOPICS
BYE	:	I WILL SAY GOODBYE NOW

The choice of this kind of 'transformation' code was preferred to numeric 'associative' codes. Stewart (1979), in comparing the two types, says "both types of code have their place, but the transformation code is easier to remember and appears more friendly".

The user received the list on entering the subroutine and was invited to type in one of the commands. He was also told to try and remember any other commands which he would need. Having entered a command, the list was erased from the screen as the details of the chosen cinema or theatre were displayed. Following this, the user either typed in a further command from memory or simply pressed 'return' to obtain the original list. If an incorrect command was typed, then the list was displayed automatically.

How successful were users in memorising this list of commands? Discounting the initial command which was always entered with the list in view, 42% were entered correctly from memory, 6% were incorrectly typed causing the list to be re-displayed and 52% were input after the user had specifically requested the list. This final percentage indicates that on just over half the occasions, the user not only failed to recall the command but felt unable to attempt an input.

A look at users' interactive sequences is also of interest. Consider the following sequence:

LIST - COMMAND
COMMAND
LIST - COMMAND

This indicates that the user entered three commands. The first, as for every user, was given with the list in view. The second was correctly given from memory while the third required re-display of the list. Abbreviating the notation to the symbols "L/C" and "C", the number of users associated with each of the four possible interactive sequence permutations is presented below:

NUMBER OF USERS	6	7	10	5
INTERACTIVE SEQUENCE	L/C L/C L/C	L/C L/C C	L/C C L/C	L/C C C

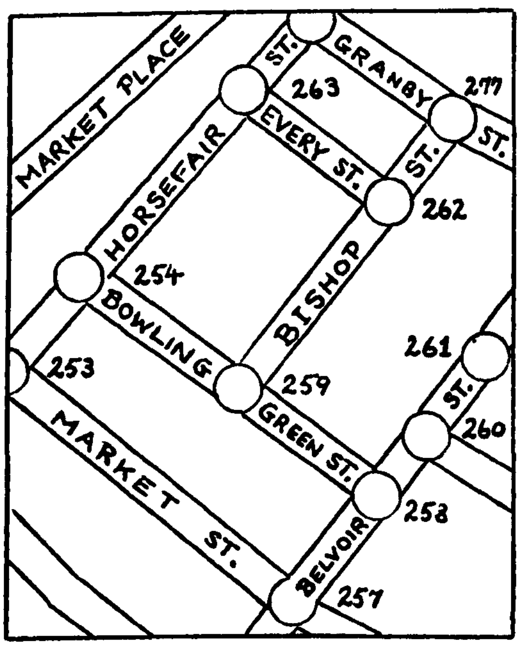
It can be seen that 23 of the 28 users had to refer to the list at least twice to enter three commands. When one studies the interactive sequences of four commands, 8 out of 16 users referred to the list at least three times.

This data gives evidence that the ability of naive users to memorise even a short list of commands is severely limited and should not be relied upon. An alternative strategy when pages are to be displayed in response to commands, is to include within each page a column of available commands. If the commands themselves bear some relation to their meaning, it is likely that users would only rarely need to call for re-display of the list of command meanings.

5.4.4 User reaction to a long interactive process

The section entitled 'Street Directions' provided a tool for studying the reaction of the general public to completing a lengthy interactive process to reach their intended goal. This routine automatically constructed a route from the Information Bureau to any street in the City centre specified by the user. However it was modified to involve the user in the route creation process.

The computer's internal representation of the street plan was a network structure with the nodes representing junctions, and branches forming the streets themselves. Starting from a node representing the Bureau, the aim was to reach one of the nodes forming the destination street. This took place in a number of stages. At each stage the user's task was to guide the computer from the current node by selecting the branch which ran most directly towards the nearest destination node.



Two dialogue forms were developed to allow the user to take part in this guiding process. They were termed "linguistic" and "graphic". The following examples of a stage within the route finding process illustrate the users' task within each dialogue.

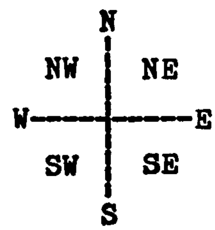
A STAGE IN THE LINGUISTIC DIALOGUE

COMPUTER: "YOUR DESTINATION LIES 400 YARDS NORTH

IN WHICH DIRECTION DO YOU WANT TO GO?

(1) NORTH EAST
(2) SOUTH
(3) SOUTH WEST

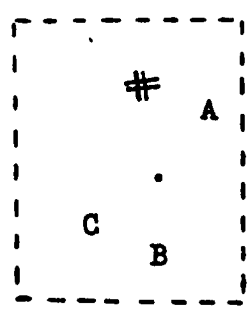
TYPE IN THE NUMBER AND PRESS <RETURN> "



USER: "1"

EQUIVALENT STAGE IN THE GRAPHIC DIALOGUE

COMPUTER: "YOUR DESTINATION LIES 400 YARDS IN THE DIRECTION OF THE SYMBOL # SELECT WHICH PATH YOU WISH TO TAKE BY TYPING IN ONE OF THE LETTERS A TO C"



USER: "A"

A comparison of the two dialogue approaches was made in terms of 'thinking time' required by the user at each stage to respond to the computer's question. It was predicted that the graphic dialogue would prove the easier of the two. In fact the mean 'thinking time' for each was very similar (linguistic = 34.2 seconds, graphic = 31.3 seconds) and a student t-test showed that the difference between them was not statistically significant.

As the process of creating each route could take as many as 12 stages, the user was frequently faced with a long interactive sequence. At regular intervals he was given the opportunity to abandon the task. The results show that abandonment took place after an average of 5 stages and a total thinking time of 170 seconds. However, whereas one user devoted 475 seconds to the task, another retired after 88 seconds.

Clearly one should be wary of these results since the user was artificially compelled to assist with a task which the computer could easily have performed automatically. It may be that abandonment by some users occurred because they felt that this method of route finding was greatly inferior to simply studying a map. Nevertheless, it may be concluded that users will vary widely in their tolerance towards a long interactive process. A system for the public must cater for the impatient user if it is to be generally accepted.

5.4.5 Multiple choice selection of items from a list

The task of dialogue for an interactive information retrieval system is often to assemble a group of attribute values which represent the user's enquiry. For example, if the system was required to output a list of places of historical interest, then the user might firstly need to select a certain period in history from a list such as:

PRE-ROMAN	TUDOR
ROMAN	STUART
NORMAN	GEORGIAN
MEDIEVAL	VICTORIAN

The enquiry could then be built up by displaying further lists to the user for selection of, say, a particular aspect of history (costumes, art, furniture) and particular setting (castle, hall, manor). The final enquiry consists therefore of a set of values, one for each attribute. However, were the user to be allowed to select more than one item from within each list, this would increase the scope of his retrieval request. This process may be termed 'multiple choice selection'.

The ISLA subroutine 'Historical visits in and around Leicestershire' was concerned with comparing two techniques for handling this type of selection process. The two techniques were:

(1) Multiple choice menu.

This technique was based upon traditional menu selection. Each list was displayed as a numbered series of items and the user entered his choices by typing in the corresponding numbers separated by spaces or commas.

e.g.

```
COMPUTER:     "SETTINGS
               --------
               (0) I DO NOT MIND WHICH SETTING
               (1) CASTLES
               (2) HALLS, HOUSES, OR MANORS
               (3) HISTORICAL SITES
               (4) MUSEUMS
USER:          "2,4"
```

(2) Individual option presentation.

The computer initially displays the list to the user in a brief form and asks him if he wishes to specify any values. If so, then he is guided through the list so that an explicit acceptance or rejection of each option must be given.

e.g.

COMPUTER: "WOULD YOU LIKE TO CHOOSE ANY PARTICULAR SETTINGS? (CASTLES, HALLS, SITES, MUSEUMS)

PLEASE TYPE 'YES' OR 'NO' AND <RETURN> "

USER: "YES"

COMPUTER: "FOR EACH OF THE FOLLOWING:
TYPE "P" AND <RETURN> IF YOU PREFER IT.
SIMPLY PRESS <RETURN> OTHERWISE.

CASTLES?"

USER: (Presses return key)

COMPUTER: "HALLS, HOUSES, OR MANORS"

USER: "P"

...

...

During the investigation, the dialogue switched between the two techniques for alternate users of the section. Thus equal numbers of users were exposed to each method. The total number of available choices was sixteen, divided into four lists based upon historical setting, aspect, period and distance from Leicester.

Comparison of the two approaches was based upon two measurements.

(1) The average number of choices selected by the user. If one technique could be shown to elicit a significantly higher number of choices, then it could be regarded as the more effective in representing the user's complete enquiry.

(2) The average time taken for each choice to be selected. The technique which produced the shorter time may be looked upon as the more efficient.

The results of these two measures are summarised below:

TECHNIQUE	SAMPLE SIZE	MEAN NUMBER OF CHOICES SELECTED	MEAN TIME FOR CHOICE SELECTION
Multiple choice menu	16	3.25	2.47 seconds
Individual option presentation	16	4.31	1.65 seconds

The application of a students t-test revealed that the differences in the mean number of choices selected and the mean times for choice selection were statistically not significant.

It is perhaps of no surprise that on average more choices were selected when users were required to respond to each option in the list. In contrast to the menu list, the user's attention is focused upon every item and the chances of inadvertently failing to select one are reduced.

Considering the mean time for choice selection, the multiple choice menu approach was found to be the slower of the techniques. This may have been partly due to differences in the methods of user response, the typing of numbers and separators being more complex than typing in a single letter.

In summary then, no significant differences were discovered between the two multiple choice selection techniques in terms of effectiveness or efficiency. This is of interest because at first sight the approach of presenting list items individually seems a cumbersome one. Such an approach may be of value when the designer of a system wishes to make certain that the user pays attention to all available choices within a given list.

5.4.6 The maximum number of information pages acceptable to the user

When it is necessary to display a certain quantity of informational text to the user, the system designer may need to spread this over a number of screen pages. It was of interest, therefore, to determine an acceptable maximum number of pages beyond which the reader feels saturated and does not wish to continue.

The subroutine concerning historical visits around Leicester invited the user to specify the types of places he would like to visit and then provided him with a suitable list. The name and a short description of each place was displayed, two per screen page. This took approximately 36 seconds. Following each pair of descriptions, the user elected either to view a further page or to halt the listing.

From a total of 32 users, 29 terminated the output before seeing the whole list, and on average 3.25 screen pages were displayed to each user. Since over 90% of users terminated the list, this figure can be regarded as generally acceptable maximum number of information pages.

However, a number of reservations should be stated with regard to this conclusion. Firstly, it is based on a given screen size (19 x 80 characters) and transfer rate (300 baud) and cannot be generalised to display devices with different characteristics. Also the test was restricted to an application where users browsed through information. Where information was required to satisfy a particular need, such as obtaining a list of train times, users might wish to view different numbers of pages.

5.4.7 Pictorial versus textual displays

In the weather forecast subroutine, three presentations were available:

(1) A pictorial description.
(2) A written description.
(3) A brief description.

A portion of the pictorial version is shown below:

```
=============================================================
! LEICESTER'S WEATHER        WINDS          THIS EVENING'S    !
!   THIS AFTERNOON        >>>--1--->S          WEATHER        !
!                            ..1..                            !
!    (  )     (     )       / . . \       (   )     !   /     !
!  (......)  (         )    1: < :1      (        )   ! /     !
!            (..........    \ . . /     (           ) 0 - -   !
! /    /     /      /        1!  1     (...      .....)\      !
!                            1   1       (......)      \      !
!     /           /          1 ! 1                            !
!          /                 1!  1                            !
!                            1   1                            !
!    /        /              1   1                            !
!                            1 ! 1                            !
!     /    /     /   /       1 ! 1                            !
!                           --  --                            !
!       /     /            1      1                           !
=============================================================
```

It was required to discover the relative importance that users placed upon the attributes 'pictorial', 'written', and 'brief'. This was based upon a comparison of the number of times each option was chosen. To remove any effect incurred by the order in which the options were presented, a random ordering was selected for each user. The application of a 'chi-square' statistical test to the overall frequencies for each display showed no significant user preferences.

However, it was noted that many people were interested in seeing more than a single display. When the same test was restricted to the first presentation requested by each user, the frequency counts were as follows:

Pictorial	-	30
Written	-	19
Brief	-	14

The number of requests for the pictorial display was significantly larger than for the other two. Nevertheless, if one considers both the written and brief displays as a single category, 'textual presentation' with a total count of 33 (19 + 14), this does not significantly differ from the pictorial count.

Only slight evidence exists, therefore, to indicate a preference for pictorial displays.

5.4.8 The value of users comments in adding to the database

It may be possible for the information system to regard the general public user not only as a consumer but also as a provider of information. The quality of information received via the keyboard was tested in the section on pubs, hotels, restaurants, and discos. Before leaving this section, the user was asked whether he had a favourite establishment. If he answered in the affirmative, then he was invited to type in both its name and the main reason for his liking it. Those comments deemed suitable were inserted into the computer's list for subsequent users to see. How useful were the comments?

From a total of 53 users who reached the point of being invited to name a favourite establishment, 36 names were entered. Of these, 3 were not accompanied by a comment and 8 of the comments received were regarded as facetious. From the remaining 25, 15 were thought suitable for inclusion in the database. These conveyed information which could be of interest to other users. For example, "six courses for six pounds", "hot spicey drinks served" and "supposedly haunted".

It would seem that people are willing to submit comments via the terminal although the open ended nature of the question caused few to be of value. Questions which put greater restrictions upon the scope of the answers might produce more useful information.

5.5 OBSERVATIONS ARISING FROM A STUDY OF THE INTERACTIVE LOG

The following observations were made during a global examination of the interaction log. The argument that these observations typify naive user behaviour is strengthened by the fact that each was detected within two or more of the topic subroutines.

5.5.1 Users' tendency to exceed unnatural information boundaries

It was occasionally found that enquiries put to the system went beyond the limits of the information available. Two examples of this are described.

The subroutine devoted to the Leicester City Football Club attempted to answer users' queries phrased in natural language about the team's results and fixtures. (A simple keyword - keyphrase recognition approach was adopted to achieve this). On entering the section, a series of example questions were displayed to reinforce the domain of acceptable queries. It was found that although every question related to Leicester City Football Club, they were not all concerned with results and fixtures.

For example:

"Who is the chief scout at the club?"
"When did Peter Shilton leave?"
"How old is Mark Wallington?"

A similar situation arose in the section concerned with Leicester's public houses. On a number of occasions users either put an enquiry or provided a comment, for a pub outside Leicester, in one of the surrounding small towns or villages.

In these two examples it appears that the information boundary perceived by the user differed from that expressed to him by the computer. It is argued that this mis-match is likely to occur when an unnatural boundary is drawn around the information which the system can provide. Since all the football questions posed concerned Leicester City Football Club as a whole then this can be regarded as having been the natural boundary. Similarly, the county of Leicestershire was the natural boundary for the public houses section as all those referred to fell within that area. This may be particularly true for people who prefer to travel out of town for a social evening.

Although it would seem preferable to satisfy users' expectations, in drawing limits around the information available, this may encompass too large an amount of data. In these cases, more artificial limits have to be accepted. When an enquiry over-steps the boundary it would be advantageous to be able to recognise this situation. An appropriate message could then be displayed to the user which reminds him of the system's capabilities.

e.g.

COMPUTER:	"TYPE IN THE NAME OF THE PUBLIC HOUSE OF WHICH YOU REQUIRE DETAILS"
USER:	"WHEATSHEAF, WOODHOUSE EAVES"
COMPUTER:	"UNFORTUNATELY THIS SYSTEM DOES NOT COVER WOODHOUSE EAVES INFORMATION IS ONLY AVAILABLE FOR PUBS IN LEICESTER".

5.5.2 Memory refreshment

The interaction log revealed that 22 users requested a certain piece of information on two separate occasions within their terminal session. In some cases the user took advantage of a built-in facility to have the previous page of information re-displayed. It was possible, for instance, to retrieve a set of street directions immediately following their disappearance from the screen. In other cases, having left a particular subroutine, the user had to re-enter it in order to repeat a previous display.

These findings suggest that the user had failed to absorb certain information and wished to refresh his memory, as frequently occurs in man-man interaction. Unlike human beings, computers are not prone to giving an impatient response when asked to repeat information, and the receiver is therefore likely to be less wary of making such a request. Dialogue designers should capitalise on the computer's tolerance by providing options which allow the user to retrieve previous displays as easily as possible.

5.5.3 The problem of taking an incorrect dialogue path

One user was found to have searched for details of an establishment within the hotels section of the database when in fact it had been stored as a restaurant. After failing to obtain details corresponding to the name originally submitted to the computer, the user made a further attempt by modifying the name. He was clearly unaware that his enquiry had been directed to the wrong part of the database.

In more general terms, the user had taken an incorrect path in attempting to reach his target information. To provide assistance to him in this situation there are three levels on which the problem can be approached.

(a) The message informing the user of the failure to satisfy his enquiry also includes a summary of the process which took place.

e.g.

COMPUTER: "REYNARDS IS NOT LISTED UNDER HOTELS"

This method depends on the user realising that a wrong path has been taken and that a new route to his target information should be tried.

(b) Supplement the above message with a suggestion of how the interaction miscarried.

COMPUTER:	"REYNARDS IS NOT LISTED UNDER HOTELS PERHAPS IT IS LISTED AS A PUB OR RESTAURANT"

Here the problem would be to design the system so that sensible suggestions are made.

(c) Automatically expand the search for the required information, to other parts of the database.

COMPUTER:	"REYNARDS IS NOT LISTED UNDER HOTELS, BUT I HAVE FOUND REYNARDS RESTAURANT IN STONEYGATE. DO YOU WANT DETAILS OF THIS?"

This approach assumes that the extra search costs involved are not too great. Also it may not be a simple matter to decide which parts of the database are most likely to contain the information required.

5.5.4 Information retrieval structures to satisfy users' requirements

The original design of the subroutine for street directions, necessitated that the user specified the name of his destination street. Later it was realised that the user may have a destination in mind (such as the cinema or bus station) but may not know the street in which it was situated. This led to an extension of the subroutine to include routes for places of general interest.

The interaction log revealed that of the 55 routes produced, 22 were requested for a particular place as opposed to a particular street. Also on 4 occasions, the user enquired about a certain place and on being told of its street location, decided that they did not then need the street directions.

It can be seen then, that the addition of this information retrieval structure served to make the route finding section more effective.

However, during two separate visits to the Bureau, it was informally observed that the retrieval structures for information on films and plays was inadequate. While this routine allowed users to request details of performances at any cinema or theatre, the inverse was not possible. For instance, one user wanted to pose the questions "Where is the Monty Python film showing?" Unable to do this, he was forced to take the less desirable path of requesting details for each cinema in turn.

One can conclude that care should be taken to design into the system, retrieval paths that will correspond to most if not all of the enquiries that the user is likely to have.

5.6 SUMMARY

A brief summary of the investigation's findings together with the inferences drawn from them are presented in the table below:

SECTION	FINDING	INFERENCE
5.3.1	66% of users spent less than 9 minutes on the ISLA system	This desire for short interactive sessions means that user operation of the system should be achieved quickly.
5.3.2	17.5% of dialogue sessions abandoned without signing off.	A facility such as a reset key is required.
5.3.3	Transfer rate of 300 baud was too slow for many users.	
5.3.4	Annoyance occurred when no explanation was given for system failure.	An automatically generated computer message would guarantee that an explanation was provided.
5.3.5	System used mainly as a browsing facility.	
5.3.6	Reluctance to use the system by members of older age groups.	Careful design of the message of introduction is needed.
5.4.1	54.8% of people using the train information section made use of the short cut facilities in menus.	The inclusion of similar short cut facilities is desirable.

SECTION	FINDING	INFERENCE
5.4.2	5% error rate for simple textual commands.	Error correcting facilities need not be too sophisticated, providing that a back-up system is supplied.
5.4.3	42% of commands in an 8 item list were entered from memory.	Asking naive users to remember commands was not successful.
5.4.4	Wide variations were recorded in users' willingness to carry out a lengthy interactive process.	The system should be designed for the least tolerant user.
5.4.5	No significant difference found between the two methods of enquiry specification: (i) menu selection and (ii) individual option presentation.	Latter approach may be of value when the user is required to consider every item in a list.
5.4.6	Users were prepared, on average, to receive a maximum of 3.25 information display pages (at 300 baud) in a single sequence.	
5.4.7	Some evidence that pictorial presentation of information was preferred to textual display.	

SECTION	FINDING	INFERENCE
5.4.8	15 comments, useful for inclusion in a database, were obtained from 53 users.	A style of questioning which puts greater restriction on the scope of the answers might produce more useful data.
5.5.1	Tendency to pose queries outside unnatural information boundaries.	Create, where possible, natural boundaries around available information.
5.5.2	22 users requested the display of information previously viewed.	Users occasionally require to refresh their memory. The system should allow them to accomplish this as easily as possible.
5.5.3	One user was recorded to have searched for a restaurant in the hotels section.	Assistance should be given to users who take a wrong path to their target information.
5.5.4	The section 'cinemas and theatres' allowed certain enquiries to be answered but could not accept inverse forms which at least two users required.	The system should provide retrieval paths which satisfy as many enquiries as possible.

5.7 CONCLUSIONS

The investigation has produced a series of results and observations pertaining to the general public's use of a computer based information system. Where appropriate, inferences have been drawn from these results. It is also interesting to consider three of them together; i.e. users tended to spend only a short time at the terminal, desired a faster transfer rate, and suggested cutting down the introductory sequence. Taken together, one requirement of the naive user seems to be a quick route to his target information. Interestingly, this precept is normally associated with the needs of the experienced user.

A study of people's written comments indicates reasonable success in using the system. One of the staff at the Information Bureau also stated that users appeared to have few difficulties in handling the terminal keyboard. This is not surprising if one considers the wide exposure of similar press button facilities to the public such as pocket calculators, computer games, and cash dispensers.

The approach of building test facilities into a working interactive system was found to be a powerful tool for studying man-computer dialogues. However, the traditional open-ended method of observing subject users or asking for their comments, frequently highlighted unexpected deficiencies in the system. For example, one user was surprised by the appearance on the screen of the letters 'DEL' (delete) after she had pressed the 'rub out' key. Combined with the two letters she had already typed, the resulting character string 'AXDEL' caused her to exclaim that she hadn't typed the last three letters. Such confusion can only be detected by observation.

Comments relating to the implementation of test facilities were also useful in preventing invalid conclusions being reached. In one case it was intended to study the effect of inserting pauses during the output of a long textual passage. The supposition was, that such time lags would help to prevent the reader from being overwhelmed by the display. However, a technical consideration demanded that the text be printed from the bottom of the screen. Where long passages are concerned, this mode of printing is difficult to read as the text is constantly moving upwards. It was commented that the pauses had been useful, not in breaking up the passage into sections, but in providing periods during which the text could be read without this distracting movement. Thus the implementation was shown to invalidate the intended test.

CHAPTER 6

PROVIDING EFFECTIVE ASSISTANCE TO THE DESIGNER OF DIALOGUES FOR GENERAL PUBLIC USE.

It was argued in the introduction (Chapter 1) that although general principles exist for the design of interactive dialogues, they need to be related to particular applications and classes of user. The main investigation (Chapter 5) concentrated upon the area of on-line information systems for direct use by the general public. The findings and associated inferences drawn from the study, have been used to develop a series of recommendations for systems of this kind. A proviso for each recommendation was that it be based upon experience gained from the main study. The thirteen guidelines described below, therefore, are not intended to constitute a complete list and it is likely that further study would reveal others of importance.

6.1 DIALOGUE GUIDELINES FOR PUBLIC INFORMATION SYSTEMS ARISING FROM THE MAIN STUDY

It is apparent that guidelines relating to the design of dialogues cannot be entirely divorced from system design recommendations in general. For instance, successful interaction will only take place if suitable boundaries are drawn around the information contained within the system. The dialogue guidelines will therefore be seen to impinge upon wider aspects of system development.

6.1.1 Involve information experts and the general public in system development

Two specific groups of people should be consulted at every stage of system development. These are firstly, information officers who have direct experience of answering queries from the public, and secondly, the general public themselves.

Given a certain topic, information officers will be able to predict the range of queries that will arise and any problems that may occur during interaction. The opinions of experts who have worked in information fields unrelated to that in which the system is to be applied, will also be useful.

When an interactive system is being developed to serve a small user population, the whole group can be assembled to discuss their needs and requirements. Systems for the public are less straightforward since the designer must rely upon samples of the user population. It is important then, that samples be representative of the general public's wide aptitude and intelligence range. This will assist in identifying a generally acceptable level of complexity for the dialogue. Novice users will quickly become 'experienced' when operating trial versions of the system and so frequent use of a particular sample should be avoided.

6.1.2 Draw natural boundaries around available information

A major problem with an information system is conveying to the user the information which it contains. Usually little more than subject headings can be given in the introduction. The user may therefore find that he has spent some time interacting with the system only to find that it cannot answer his particular enquiry. One solution is to draw natural boundaries around the information available. In other words, given a subject heading, the system should be able to answer most, if not all reasonable enquiries which are likely to arise. For example, to comprehensively cover theatre information, details should be provided for all nearby theatres of current and forthcoming productions, seat prices and ticket availability.

The problem arises of how one decides where, for a particular class of information, the 'natural' boundary lies. Clearly the designer will need to carry out research among members of the general public to assist him in making decisions of this kind.

6.1.3 Consider first impressions

It is important to consider the initial impression conveyed by the system to the user. If it does not give the appearance of being easy to use, then interaction may not be attempted.

The introductory display should be both clearly laid out and kept as brief as possible. This may not be a straightforward task since the introduction must include subject headings for the available information and instructions for using the keyboard.

The keyboard itself should not be over elaborate as this can also be a source of discouragement. The ISLA system was based upon the alphanumeric range; A-Z and 0-9, and the three keys 'return', 'rub out' and 'reset'. This did not appear to be too wide a range for the general public to handle efficiently.

6.1.4 Devise simple, concise and unambiguous output messages

The nervous user will feel reassured if the computer appears to be communicating with him in a way that he can easily understand. Output messages should, therefore, be expressed in the simplest of terms. Since the general public consists of people with widely differing mental abilities, simple dialogues are also needed to satisfy as broad a section of this range as possible.

Concise computer messages are also important since users will not wish to spend long periods absorbing instructions. However, in striving for brevity, the dialogue designer must be careful to avoid ambiguity. As an example, the presentation of a list of numbered options might be followed by the instruction;

"KEY NUMBER REQUIRED"

This would cause some users to wonder whether a special 'key number' was required! By expanding the message to read;

"KEY THE NUMBER OF THE OPTION REQUIRED"

the correct meaning becomes apparent.

It has been found that consultation with potential users is the most rigorous method for highlighting deficiencies in computer messages.

6.1.5 Provide quick access

The user will wish to reach his target information as quickly as possible. This means that the introduction must be designed to allow him to embark on interaction without delay and that the dialogue sequence which follows should involve no more than a few inputs.

The information itself should be displayed at a fast rate and presented in a compact form so that retrieval of incorrect or irrelevant data does not delay further interaction. Users are eager to compare the access times of a computer system with those for alternative facilities. If the computer proves the slower it may be rejected even though it is superior in other respects.

Once the computer has answered an enquiry, the next stage will be to offer the user a list of options for continuing the dialogue. If the system designer can identify particular enquiries which are likely to follow, then these can be included within the option list. In this way the interactive process can be short-circuited and thus speeded up. As an example; once a person had obtained a description of a certain tourist spot, the computer could then offer to provide a route for getting there.

6.1.6 Build in all desired retrieval paths

It is important that the dialogue designer considers all the information retrieval paths which the user is likely to require. Within the ISLA program, although the details of productions at a certain cinema could be requested, it was not possible to ask where a particular film was showing. To answer the latter question, the user was forced to scan the information for each cinema.

As with the guideline; 'draw natural boundaries around available information', the designer must carry out the necessary market research before determining which retrieval paths should be included.

6.1.7 Ask appropriate questions

Once the user has specified the subject of his enquiry, the computer will then ask him a number of questions to obtain the details of it. Care should be taken in the design of this type of dialogue.

One possible downfall is that the user may be over questioned. For example in specifying a train timetable enquiry it might seem appropriate to ask the user his destination, day of travel and approximate time of departure. However, during trials of the ISLA program, which originally employed this line of interrogation, it was found that users did not wish to specify a travelling time but preferred to select from the whole list of trains running on their chosen day.

A second danger is putting a question to which the user either does not know the answer, or feels is irrelevant. A hypothetical example is given below:

COMPUTER: "IN WHICH BRITISH RAIL REGION IS YOUR DESTINATION?"

(1) NORTH WESTERN	(2) NORTH EASTERN
(3) WESTERN	(4) SOUTHERN
(5) MIDLAND	(6) SOUTH WESTERN

"ENTER THE APPROPRIATE NUMBER"

USER: "4"

(Here the user may not know the correct region and so is forced to make a guess)

COMPUTER: "SPECIFY THE SERVICE YOU REQUIRE?"

(1) INTER-CITY ONLY
(2) ALL SERVICES

(While this question may be important to some users, many would consider it irrelevant).

6.1.8 Avoid unnecessary clarification

Many texts put great store upon the value of obtaining user clarification that a series of inputs is correct. In the case of a public information system this procedure should normally prove unnecessary. Casual or naive users tend to adopt great care in forming their inputs correctly. If the recommendation that the system should provide quick access, is heeded, then the user will be able to maintain this cautious approach.

Additionally, since the public wish to reach their desired information quickly, the need to clarify inputs will be seen as a chore rather than a safety net. This was found to be the case in the train information section of the ISLA program.

In spite of this argument, certain situations can arise where input verification will be necessary, for instance when the cost of an incorrect search of the database is high. Also, if an incorrect search is likely to take a long time or produce a great quantity of data, the clarification of inputs can be a valuable preventative tool.

6.1.9 Provide facilities for memory refreshment

As with any reading task, it is likely that the user will not fully absorb the information presented to him on the display screen. When this is replaced by a subsequent display, he may, therefore wish to refresh his memory. The dialogue should provide facilities to allow this information to be recalled easily. The ability to 'revolve around' or 'backstep through' a sequence of display frames are possible solutions. Also, when moving from one interactive stage to the next, the list of possible courses of action could include an option to see the last piece of information again.

6.1.10 Suggest the causes of errors

It may be helpful if the computer could suggest the cause of an invalid input entered by the user. Nevertheless, the dialogue designer should only employ this technique where he is confident that the suggested source of error will usually be correct.

This was not the case with the route finding section of the ISLA program. When the computer failed to recognise the street name entered, it was suggested that the user had made a typing error. However, the system was restricted to the recognition of city centre street names, so this suggestion could be misleading if a street outside the city had been typed in. A more suitable proposal as to the cause of error could be made in the following dialogue situation:

```
COMPUTER:  "WHICH OF THE FOLLOWING SETTINGS ARE
            YOU INTERESTED IN

            (1)  CASTLES
            (2)  HOUSES
            (3)  MUSEUMS

           "TYPE IN THE APPROPRIATE NUMBERS
            SEPARATED BY COMMAS"

USER:      "13"

COMPUTER:  "I HAVE RECEIVED THE NUMBER 13 PLEASE TRY
            AGAIN REMEMBERING TO SEPARATE THE NUMBERS
            WITH COMMAS".

USER:      "1,3"
```

6.1.11 Design the system to be incorruptible

The system should be designed so that it cannot be corrupted by invalid inputs. This is particularly important for the general public whose lack of terminal experience can cause frequent errors to occur. As Martin (1973) says:

> "It can be a demoralising experience for an operator without computer training when his dialogue with the machine goes off course".

The designer should also bear in mind that some users will deliberately enter extreme or incorrect inputs to discover what the system's responses will be.

The user should be protected from seeing any system program messages by substituting them with clear dialogue error messages. For example:-

> "YOU HAVE MADE AN ERROR IN TYPING
> A NUMBER BETWEEN 1 AND 10.
> PLEASE TRY AGAIN."

6.1.12 Encourage feedback

Once the system is operational, it is necessary to gain direct feedback from the public. This might consist of notification of system faults, incorrect information on display or comments regarding the interactive dialogue.

There are a number of mechanisms for collecting this data such as providing a comments book or a facility for remarks to be entered through the terminal. An alternative method is to invite users to pass on their observations to a human receptionist. This approach has the advantage of permitting the receiver to probe more deeply into statements made and to request that the user enlarges upon them. The receptionist must, of course, guard against distorting this flow of information whilst transcribing it.

It has been found that, unlike frequent users of a system, casual users such as the general public will tend to accept deficiencies in the service without formal complaint. Thus feedback should be encouraged by the use of a notice stating that comments will be taken into account when improvements are being made.

6.1.13 Implement a reset key

It is likely that a user will abandon his dialogue with the computer without formally signing off. Subsequent potential users will thus be faced with a confusing series of messages on the screen which could prevent them from attempting interaction. A simple solution is to provide a reset key which, on being pressed, immediately recalls the introductory display. A notice will also be required to draw attention to this facility.

6.2 SUMMARY OF DIALOGUE GUIDELINES

The following table provides a brief summary of the recommendations described in the previous section.

(1) Involve information experts and the general public in system development.	Information officers can predict the likely range of queries and possible problems during interaction. Frequent use of a particular sample of the general public will cause them to gain experience and become untypical of naive operators.
(2) Draw natural boundaries around available information.	Given a particular subject area, the system should be able to answer all reasonable enquiries which arise.
(3) Consider first impressions.	The appearance of an over complex introductory display or keyboard will discourage people from using the system.
(4) Devise simple, concise and unambiguous output messages.	Adherence to this requirement will ensure that the system caters for a wide range of mental abilities.
(5) Provide quick access.	Users should be allowed to reach their target information as quickly as possible. The list of options following an information display should include short cut paths to likely supplementary questions.
(6) Build in all desired retrieval paths.	Include all the paths for retrieving data that the user is likely to require.

(7) Ask appropriate questions.	Ensure that the user is not required to specify parts of an enquiry which he may prefer to remain flexible. Also avoid posing questions to which few people will know the answer or which may seem irrelevant.
(8) Avoid unnecessary clarification.	Clarification of inputs will normally be an unnecessary technique as general public users tend to proceed carefully when specifying an enquiry.
(9) Provide facilities for memory refreshment.	Users may not fully absorb the information presented on the screen and should be provided with a mechanism for recalling it.
(10) Suggest the causes of errors.	Computer generated messages suggesting the cause of an input error or the failure to obtain the required data will be helpful provided they are usually accurate.
(11) Design the system to be incorruptible.	This is particularly important as some users will deliberately attempt to make the system fail.
(12) Encourage feedback.	A comments book or computer file are suitable channels for testing public reaction. However, a human receptionist will be able to gather more comprehensive feedback.
(13) Implement a reset key.	This facility is needed to recover from interactive sessions which are abandoned without being formally terminated.

6.3 A CONSIDERATION OF THE PERTINENCE OF PUBLISHED GUIDELINES TO PUBLIC INFORMATION SYSTEMS

This section reviews the published recommendations set out in the Introduction (Chapter 1) and in the light of the project's findings, discusses their relevance to on-line public information facilities. While some of the comments are subjectively based, others can be supported by evidence from the ISLA study.

Two principles are clearly appropriate to all systems involving man-computer interaction. These are:

(a) Knowing and communicating with the intended user.

(b) Obtaining user feedback following implementation.

However, the main study has shown that the application of these principles is not a straightforward matter in the case of systems for the general public.

One of the main assumptions underlying the project has been that of direct user contact with the computer. Thus the suggestion that an information specialist could act as an intermediary between user and computer was not extensively tested. Nevertheless, it should be noted that the presence of an intermediary may inhibit the possibility of browsing through information, which was found to be an important requirement of the ISLA system.

No direct evidence was obtained to support any one of the three conflicting strategies for system response times. In the author's opinion, users will require standardised delays if they are to feel at ease during interaction, although their need for swift access to information dictates that short response times are also desirable. Since few members of the public will have specialised computing knowledge, the approach of linking the response delay to the complexity of the computer process in operation, will not be appropriate.

The suggestion that the computer be programmed to accept synonyms of commands was countered by the argument that novice users may be confused by too wide a range of expression. Regarding systems for the general public, the computer should normally display a list of command options alongside each request for input and therefore the acceptance of synonyms becomes unnecessary. Where the range of input options is large and consists of elements which are publicly known (such as the basic colours, railway stations or football grounds), then it has been found to be more convenient for the user to enter his input without initially receiving a command list. In such cases, the public will express particular inputs in a variety of ways and so facilities for accepting synonyms becomes an essential feature. For example the names "CHELSEA'S FOOTBALL GROUND" and "STAMFORD BRIDGE" would be regarded as equivalent.

The traditional concept of input abbreviation allows commands to be truncated up to their minimal unambiguous form. Thus for the commands 'SAVE', and 'SCRATCH', the minimal forms 'SA' and 'SC' are sufficient to distinguish them. However, the general public will only assume the use of abbreviations which are present in everyday writing and speech, such as 'PUB', 'CAFE', 'USSR' or 'IBM'. It is these forms which should be included within the valid input range.

Attention is drawn to the guideline stated in section 6.1; that natural boundaries should be drawn around the information provided by the system. This approach can also be applied to the designation of valid inputs. In other words, any synonym or abbreviation of a given command which is in common use should also be appointed as a legal input.

During the ISLA study, some users appeared daunted by interaction with the system and seemed likely to abandon it if faced with any confusing or unclear computer messages. Coupling this with the need to provide quick and simple access to information, it may be preferable to design a highly explanatory surface level dialogue rather than a cryptic form supplemented by a help key or command to provide fuller descriptions.

This is not to dismiss the proposal to include help facilities and it may be appropriate to provide the option for particular output statements to be explained more fully.

The approach of conducting dialogue through a process of menu selection was regarded as valuable since it absolves the user from the need to commit commands to memory. Clearly this philosophy is relevant to the general public who cannot be given training in computer usage. However, menu selection will not be appropriate when the list of options (of say book titles, place names or company names), is too large to be displayed conveniently. As already mentioned, an alternative technique would be for the user to enter the option name without the aid of a reference list. Of course this may entail further dialogue to clarify the choice intended.

The suggestion that a simple reference list is a means of providing a substantial amount of information in a small area, may provide an additional solution to the problem of displaying a large menu list.

Consider the illustrative example shown overleaf.

COMPUTER:	"DO YOU KNOW THE NAME OF THE MEMBER OF PARLIAMENT WHO'S ADDRESS YOU REQUIRE? (ANSWER YES OR NO)"
USER:	"NO"
COMPUTER:	"CAN YOU TELL ME THE NAME OF THE MEMBER'S CONSTITUENCY? (ANSWER YES OR NO)"
USER:	"NO"
COMPUTER:	"PLEASE REFER TO THE LIST OF M.P.s. AND THEIR CONSTITUENCIES DISPLAYED ON THE WALL DIRECTLY IN FRONT OF YOU. IF YOU RECOGNISE THE NAME OF THE M.P. REQUIRED, TYPE IN THE NUMBER UNDER WHICH HE OR SHE IS LISTED. IF YOU ARE UNSUCCESSFUL, TYPE Ø"
USER:	"184"

The idea of a reference list could also assist in solving another problem identified in the main study. It was found that the single display screen comprising the introduction to the ISLA system was barely adequate to describe the topics of information contained within it. This caused frustration for those users who spent time interacting with the system only to find that their queries could not be answered. A wall poster, therefore, could provide a larger space to display fuller details of the subjects contained within the system.

Although the proposal for an automatic help facility was not directly tested, an associated area was considered in connection with the abandonment by users of unfinished dialogues. To cope with these occurrences, a time delay was built into the system, after which the introductory section of the dialogue would be displayed for the following user. Difficulty was experienced in the selection of a time interval which would allow the dialogue to be reset reasonably quickly and yet did not interrupt the slower user involved with interaction. This led to withdrawal of the facility and to the author becoming wary of similar automatic computer action.

The recommendation that user errors should be restricted by making inputs both natural and consistent would seem to be a desirable goal. This is particularly relevant to general public users without typing skills or experience of terminal keyboards. However, the problem arises of what constitutes a 'natural' input sequence. Experimentation would be needed to resolve this question.

One of the published guidelines stated that error messages should be understandable, non-threatening and low key. To these attributes can be added the recommendation from the ISLA study that they should also be simple, concise and unambiguous.

The nature of the communication channel upon which the ISLA system was based, meant that an error message could only be displayed following completion of each input (on depression of the 'return' key.) This arrangement seems to be more suitable for novice users than the approach of immediately issuing error messages which interrupt uncompleted entries. As well as having a startling effect, the user might feel that since the computer is able to monitor inputs as they are being typed, then for standard entries such as 'YES' or 'NO', the 'return' key is an unnecessary feature.

Finally, regarding user errors, the idea was put forward that the user should be encouraged to operate the system on a trial and error basis, thus mimicking the process by which all tasks are learned. Obviously people with definite enquiries will not wish to use an information system in this way but will expect to be given clear instructions for the achievement of their objectives. However, the concept is relevant to public information systems in the sense that many people will use them for browsing purposes. The dialogue, therefore, should be designed to support this requirement by suggesting new paths of enquiry on suitable occasions during interaction.

The principle of designing easily adaptable systems was vindicated during trials of the ISLA system, when a number of alterations were required. Although the general public would be restricted to the mere suggestion of modifications, it may be possible for the information centre staff to implement simple changes themselves. These could be in respect of computer messages, input strings or database contents.

The following dialogue shows how changes to the system could be effected on-line. The example concerns the inclusion into the database of the first wine bar to be opened within a particular locality.

COMPUTER: "WHICH OF THE FOLLOWING ARE YOU INTERESTED IN? - TYPE 1, 2 OR 3

(1) PUBLIC HOUSES
(2) HOTELS
(3) RESTAURANTS"

STAFF USER: "MODIFY/HOUSES/HOUSES AND WINE BARS/"

COMPUTER: "WHICH OF THE FOLLOWING ARE YOU INTERESTED IN? - TYPE 1, 2 OR 3

(1) PUBLIC HOUSES AND WINE BARS
(2) HOTELS
(3) RESTAURANTS"

STAFF USER: "ADD TO DBASE/TREVINOS BAR, MAIN STREET, EAST BANK/"

COMPUTER: "ENTRY RECEIVED - CONTINUE".

(The user now checks the entry)

STAFF USER: "1"

COMPUTER: "STATE WHICH AREA - TYPE 1, 2, 3 OR 4

(1) NORTH SIDE
(2) SOUTH END
(3) EAST BANK
(4) WEST PARK"

STAFF USER: "3"

COMPUTER: "PUBLIC HOUSES AND WINE BARS IN THIS AREA ARE AS FOLLOWS:

THE RED LION, SAFFRON ROAD
TREVINOS BAR, MAIN STREET

. . .

. . ."

Due to limited VDU facilities, it was not possible to fully investigate either:

(a) Alternatives to textual information presentation.

(b) The use of non-verbal signals within dialogues.

It was felt, however, that the novice user could become overloaded if presented with too many parallel signals such as printed text, a bell and flashing areas of the screen. A comparison can be drawn with the difficulties experienced by the uninitiated operator of a modern 'one arm bandit' gambling machine containing flashing signals, alarms and a variety of control buttons.

The expressed need for systems to cater for different levels of user experience will not, by and large, apply to services for the public who, as first time or infrequent users, will generally require dialogue at the most explanatory level. One group who may interact on a frequent basis, are the staff of the reference library or information centre in which the system is installed. In this case a second, more succinct form of dialogue may be needed. This would necessitate the inclusion of a dialogue switching command to allow staff members to call the terse dialogue and then to return to the normal style for public use. If the system were inadvertently left in the terse dialogue state, this would, of course, confuse later users. A simpler solution may be to retain a single level of dialogue and to build in a short cut facility such as response combining (described in section 1.2.11).

6.4 SUMMARY OF ADDITIONAL DIALOGUE GUIDELINES FOR PUBLIC INFORMATIONS SYSTEMS

The previous section discussed the relevance of published dialogue design recommendations to the creation of public information systems. This has enabled a number of additional guidelines to be established regarding dialogues for the general public. These are summarised below.

Guideline	Description or remarks
(1) Standardise system response times.	A standard response delay would be preferred by the general public provided this is also of short duration.
(2) Draw natural boundaries around the selection of valid input lists.	Since many inputs can be expressed in a variety of ways, the system should accept synonyms and abbreviations which are in everyday use.
(3) Design explanatory dialogue which avoids the need of help facilities.	Users will require to access information quickly and so it may be preferable to design highly explanatory dialogue not requiring supplementary help facilities. If the user needs to search for assistance he may lose confidence in the interaction.
(4) Appreciate the virtues and deficiencies of menu selection.	It should be appreciated that menu selection is a valuable technique to employ within dialogues for the public. However, for large option lists, the approach becomes clumsy and alternative methods should be employed.
(5) Consider the employment of look up lists.	A poster on the wall can be used to display information which is too extensive to condense onto the visual display surface.

Guideline	Description or remarks
(6) Be wary of mechanisms which automatically detect that help is required.	Since it is difficult to specify the conditions which indicate that the user requires help, an automatic help facility may either assist at the wrong moment or fail to assist at the right moment.
(7) Avoid dialogue which is prone to user error.	Taking care that required inputs are both natural and consistent, can help to prevent user error.
(8) Make politeness an attribute of error messages.	Error messages should be understandable, non-threatening and low key. Furthermore they should be displayed following input completion since an interrupting message will appear unfriendly to the user.
(9) Suggest new paths of enquiry.	The proposal by the computer, of new paths of user enquiry will promote browsing through the system and encourage people without specific enquiries to seek information on a trial and error basis.
(10) Design the system to be easily adaptable.	Modification to the system following implementation is likely to be necessary and should therefore be easily achievable. An on-line dialogue editing facility would be particularly useful.
(11) Consider the use of non-verbal forms of information presentation.	Alternatives to textual presentation should be considered with the proviso that the user is not overloaded with parallel signals.

Guideline	Description or remarks
(12) Provide additional dialogue facilities for staff associated with the location in which the system is installed.	If the staff of the centre containing the information system wish to use it on a regular basis, then an additional terse dialogue may be required. Alternatively, short cut facilities can be built into the standard dialogue used by the general public.

6.5 SURVEY TO DETERMINE THE USE MADE OF DIALOGUE GUIDELINES

The task of designing good interactive software can be approached in a number of ways. For example the designer may consult the intended user, seek advice from colleagues or borrow ideas from existing programs. The provision of dialogue guidelines constitutes an additional technique. However, the question arises of whether designers do, in fact, refer to these published principles. Meister and Farr (1967), who studied a group of equipment designers at work, found that little or no interest was shown in guidelines for improving the human aspects of their designs. (Human factors information) The programmer's use of guidelines for interactive programs may be regarded as a parallel situation.

To explore this question, a survey was carried out among a group of eleven experienced application programmers. (A copy of the questionnaire appears in Appendix III). Each person was asked to study a list of design techniques and to indicate which of them they would employ. The percentage of the sample selecting each technique is presented in the table below:

TECHNIQUE	% OF SAMPLE SELECTING IT
Rely solely on previous programming experience	0%
Discuss ideas with colleagues	82%
Refer to books or research papers for guidelines on interactive system design	45.5%
Consult with potential users of the system	91%
Search the software library for similar programs	82%

It can be seen that less than half felt that they would refer to books or research papers for guidelines.

Part two of the questionnaire presented a list of guidelines derived from a number of sources: Martin (1973), Gaines and Facey (1975), Shackel (1979) and Stewart (1979). Each programmer was then asked whether, on balance they would find these principles useful in the design of an interactive dialogue. Here 82% stated that they would be of value.

The survey seems to indicate that while a set of rules would have an important role to play in the interactive design process, there is a possibility that the designer will not take the trouble to refer to them.

6.5.1 Reasons for non-referral to guidelines

A number of subjects in the survey had stated that they would not utilise guidelines from books or research papers when designing an interactive system. Members of this group were questioned to discern the reasons for their attitude.

One subject felt that the few sources of reference material available, covered a very limited area of application. Therefore information was required which could be applied to a wider range of systems. Comments from two members of the sample drew attention to a series of technical standards for documentation and programming within which they were required to work. It was thought that dialogue guidelines taken from published literature could conflict with these standards. However, further questioning revealed that conflicts would not arise since the standards dealt with general lay-out and operation of software rather than the more fundamental aspect of dialogue design.

The following recorded remarks highlight further barriers to the use of dialogue guidelines;

One subject stated that the computer literature tended to present concepts in an over complex form and that it was more convenient to refer to a colleague who could explain ideas in simpler terms.

Another response was that software frequently had to be produced to strict deadlines which did not allow time to be spent in a library, searching for design information.

Finally one person expressed a disinclination to use dialogue recommendations as this would impair the enjoyment of devising the interface for himself.

6.5.2 Discussion of subjects' attitudes towards dialogue guidelines

The previous section appears to contain a random set of reactions by designers opposed to the use of dialogue guidelines. However, by linking these comments to the backgrounds of the individuals who expressed them, then an interesting picture emerges.

The subject who called for more general guidelines was a senior level programmer who had come from an academic environment. This type of designer would be familiar with technical literature and with the task of adapting general information, such as concepts of dialogue design, to suit a particular application.

This contrasts with the views of the two junior level programmers who preferred to avoid dialogue principles on the false assumption that they might conflict with the technical standards. It is possible to speculate, therefore, that the deference shown towards standards indicates a desire, on the part of junior staff, to work within a well defined set of rules. Perhaps dialogue recommendations should be presented to this group in a similar form.

In summary, one might theorise that senior level designers require dialogue design information in the form of general principles whereas junior level programmers will prefer more specific data.

The remaining comments are indicative of a predictable lack of motivation among programmers to devote time towards obtaining apparently inessential information. The design of the man-machine interface may well be seen as one of the more creative tasks to be performed without interference. When assistance is required, a colleague's opinion, tailored to the particular problem in hand can be easily obtained.

6.6 A STRATEGY TO PROMOTE THE USE OF DIALOGUE DESIGN KNOWLEDGE

The previous survey revealed that programmers do not, as a matter of course, refer to published guidelines when designing interactive systems. This suggests that one of the major problems facing the improvement of man-computer dialogues lies not only in establishing guiding principles but also in determining a strategy for communicating them to the system designer.

6.6.1 The need to disseminate design information in a variety of forms

It is to be expected that the growing number of interactive systems will be designed within differing constraints of time and expenditure. Where these commodities are in abundance, the design team may be prepared to consider open-ended guidelines such as:

Tailor system response times to particular input operations.

This may entail carrying out a series of experiments to evaluate the optimum response times for various user inputs. Conversely, if the designer has a highly restricted budget, he can only respond to more specific recommendations such as:

Constrain all system response times to two seconds or less.

The stage reached within the development of a system will also determine which dialogue principles can play an effective role in influencing the designer. If he refers to recommendations at an early stage in the design then guidance regarding the overall interactive strategy could be of value. This would not be the case later on during the work when only recommendations of a more superficial nature can be considered.

It seems necessary, therefore, to classify design information in terms of generality and specificity so that recommendations can be selected which are applicable to both the prevailing time/cost restrictions and the current stage of system development.

This classification is also valid in relation to a third factor within the design process; the rank and background of the designer himself. It was postulated in section 6.5.2 that senior and possibly academically trained designers require information along the lines of general principles while designers of junior status prefer more precise data. If this is true, however, it will merely represent a tendency in the appropriation of various types of design information. Clearly a conscientious designer will consider all classes of data relevant to the particular application.

The term 'dialogue guideline', which has been used to denote all kinds of dialogue design information, is inadequate for this purpose. A general theory of man-computer interaction, for example, is one source of information which does not come under this heading. In fact, it is from such theories that dialogue guidelines may be generated. Thus it is proposed to employ the more global term, 'dialogue design information'.

6.6.2 Classification of design information

This section describes a possible classification for dialogue design information. The three main classes are as follows:

(1) General theories and laws.
(2) Guidelines.
(3) Techniques.

They are ordered in accordance with their degree of generality, the first class allowing wide interpretation and the third being applied in a highly specific way.

6.6.2.1 General theories and laws of man-computer interaction

A designer who wishes to approach the creation of an interactive dialogue from first principles may begin by reviewing theories and laws on the subject. At present, little or no information of this kind exists. However, work has been carried out in establishing the basic variables underlying interaction and the ways in which they interrelate. Eason (1976b) has identified six factors which apply to a variety of naive user groups. These are classed as task and job/role factors. It has been found that the relationship between these variables is an important determinant of whether the user will be able to devote sufficient effort to operate the system which he requires.

Given the dearth of general laws, a useful starting point may be to reflect upon the general characteristics of the user for whom the system is to serve. The author's own experience has revealed a number of common attributes of the general public which should be taken into account. These are listed below:

(1) Initially nervous of computer terminals but quickly gaining in confidence.

(2) Possessing a wide range of mental abilities.

(3) Critical of a computer based system which is slower or more difficult to use than traditional sources of information.

(4) Impatient and only prepared to spend a short period of time at the terminal.

(5) Lacking the knowledge and experience to overcome difficulties arising during interaction.

6.6.2.2 Guidelines

Guidelines for the design of an interactive system can be generated from general theories, laws or basic user characteristics. As an example, the characteristic of many users; that they will attempt to confuse the system by entering unusual inputs, leads to the guideline that the system should be made incorruptible. Thus a designer who works from first principles can produce his own set of guidelines or be sufficiently well informed to draw selectively upon published recommendations. Where a designer is unable or unwilling to devote resources to a deep analysis of first principles, he may still refer to tried and tested suggestions. It is perhaps for systems created in this way that there is the greatest need for guidelines which are tailored to particular applications such as public information systems.

A distinction can be made between 'general guidelines' and 'specific guidelines'. The former group represents overall principles while the latter consists of operational directives. Thus in terms of the recommendations extracted from the ISLA study:

'provide quick access'

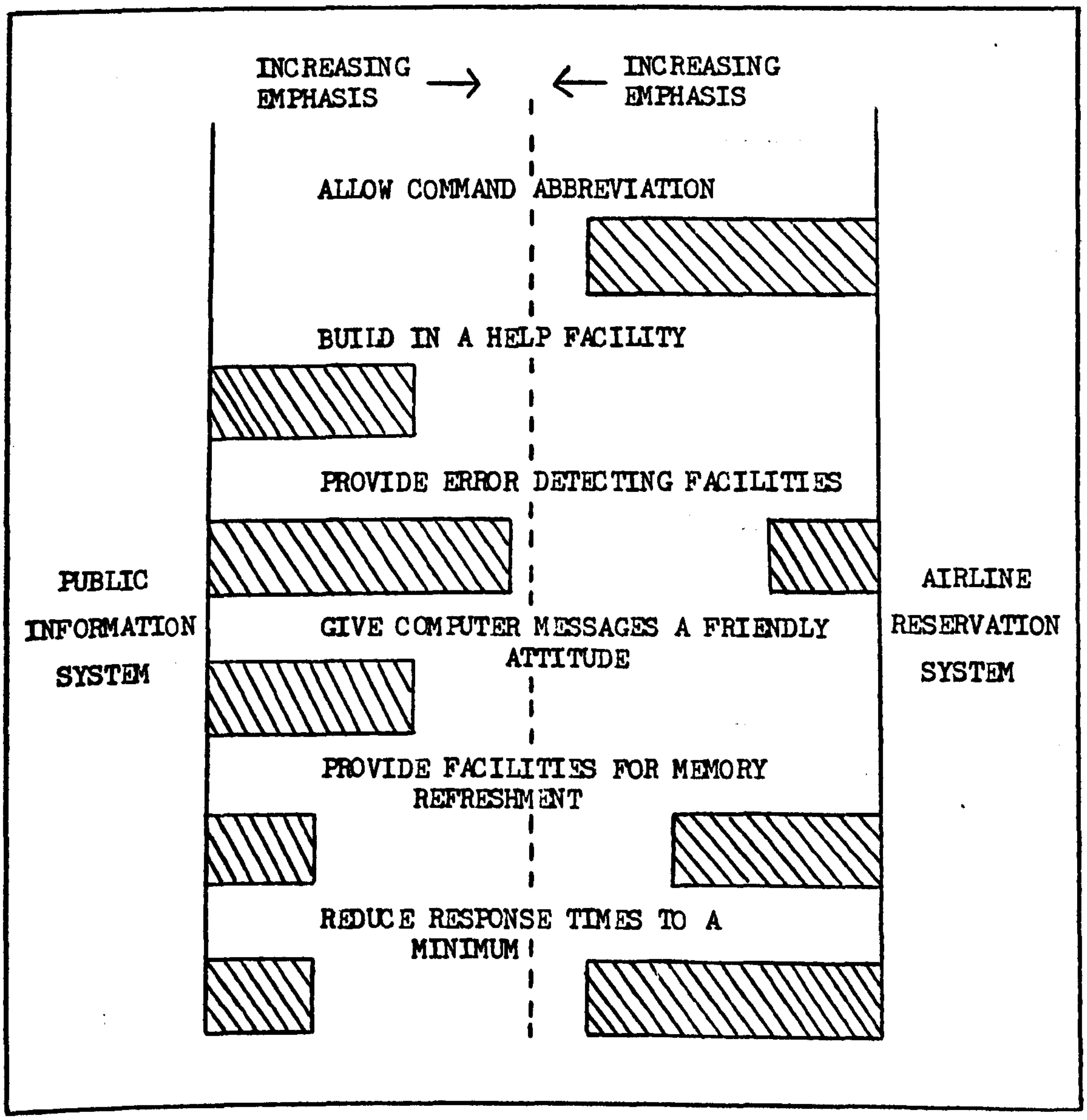
would be a general guideline conveying nothing in terms of its achievement, while:

'provide facilities for memory refreshment'

is a specific guideline which can be applied with little further enumeration. Whether the designer will prefer either group of directives will depend upon the prevailing circumstances but the distinction may be a useful one.

Since dialogue design may have to be achieved within a very limited budget, it may be necessary to place guidelines for a particular application into priority order so that resources can be devoted to those aspects of greatest importance.

The bar chart shown below illustrates a method for achieving this. The left hand side relates to an information system for direct use by the general public and the right hand side, to an airline ticket reservation system operated by trained clerks.



It should be noted that the recommendations were selected solely for the purpose of illustrating the need for priority ordering and do not constitute a complete list. Also the bar lengths are based upon the author's subjective opinion and do not result from formal study.

There are two further issues relating to guidelines which should be mentioned. Meister and Farr (1967) found that recommendations for the design of industrial equipment became more acceptable if they included pictorial illustrations. In terms of man-computer dialogue, the inclusion of dialogue extracts are equivalent to illustrations. An indication of the power of such extracts is given by the fact that they are frequently reproduced in the literature from their original sources.

Meister and Farr also suggest that placing emphasis upon the consequences of failing to adhere to guidelines would carry weight with designers. Foley and Wallace (1974) adopt this approach in relation to graphics systems by offering remedies to avoid user boredom, panic, confusion, frustration and discomfort.

6.6.2.3 Techniques

Just as guidelines can be produced from laws and theories so techniques of interaction may be extensions of guidelines. The use of menu selection with the PRESTEL system, for instance, might well have resulted from a guideline to provide a very simple, uniform method of data access. Thus designers who have been provided with a set of guidelines for the system, may utilise these to develop dialogue techniques.

However, designers who, without using guidelines, have predetermined ideas for the general configuration of a dialogue, may simply wish to refer to particular techniques to assist in overcoming specific problems. Hebditch (1973) describes the problem of designing input commands to be both economical and legible. The use of short economical command forms will make it very difficult for a less experienced person, later referring to a hard copy version, to understand the dialogue which took place. Hebditch then proposes a technique whereby the user enters a command with maximum efficiency and the computer types back the equivalent description with maximum legibility. An example is given below:

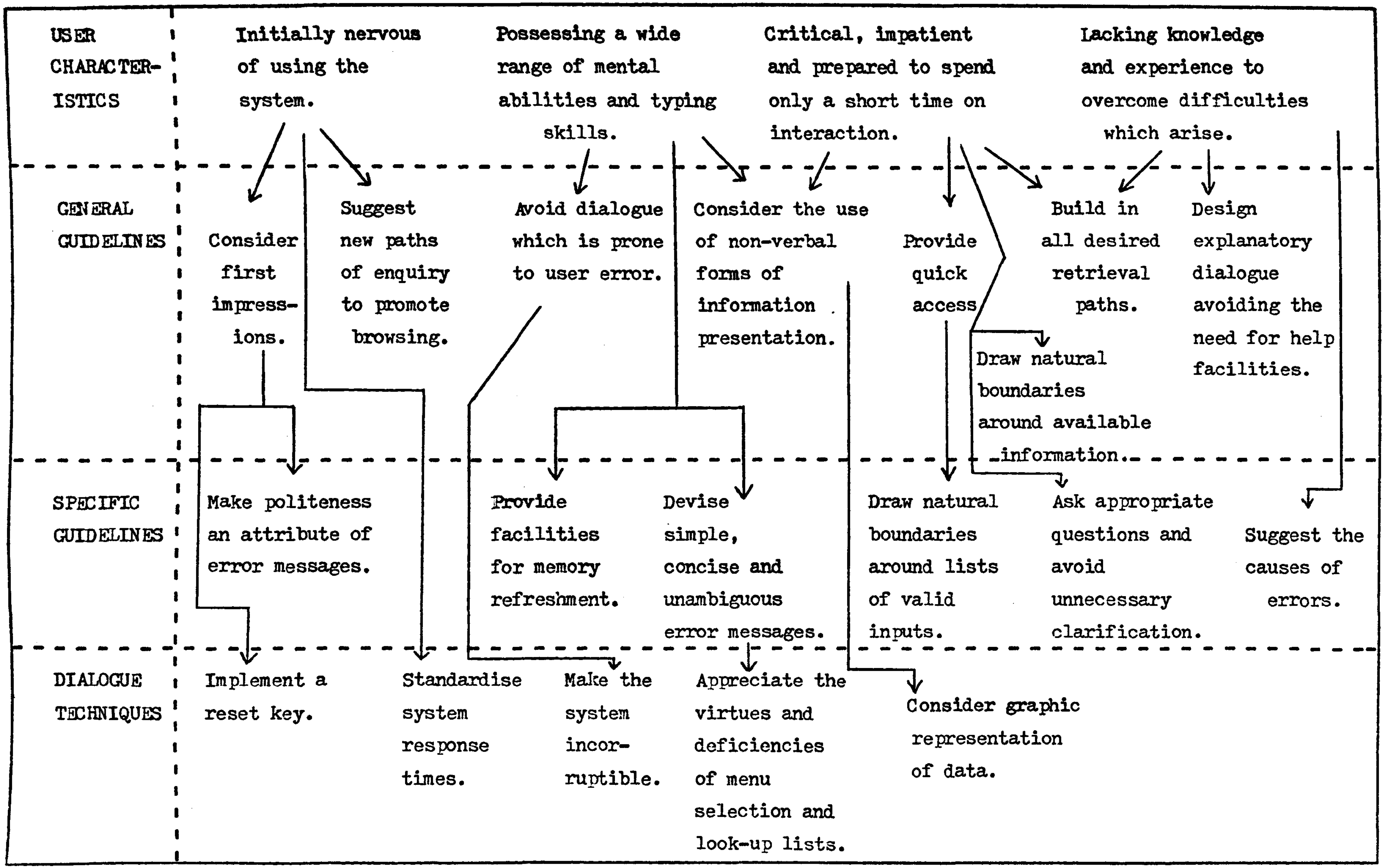
```
USER:     "EX 462"

COMPUTER: "EXAMINE ACCOUNT OF JONES LTD., LONDON"

USER:     "OR 79"

COMPUTER: "ORDERS FOR 1979

            12 JUN:  42 BOXES 80 COLUMN CARDS - BUFF
             3 SEP:  16 BOXES 40 COLUMN CARDS - BLUE
            26 OCT:   4 BOXES STANDARD LINE PRINTER PAPER"
```

Martin (1973, chapter 7) describes twenty three dialogue techniques applicable in a wide range of situations.

6.6.3 A structured summary of dialogue design information for public information systems

It has been argued that the three classes of dialogue design data are not independent since laws and theories give rise to dialogue guidelines which in turn produce dialogue techniques. At this point, then, it may be useful to summarise the design information accumulated during the ISLA study in terms of this classification. A network structure is superimposed to highlight the relationships between the various elements.

(Overleaf)


USER CHARACTERISTICS
Initially nervous of using the system.
Possessing a wide range of mental abilities and typing skills.
Critical, impatient and prepared to spend only a short time on interaction.
Lacking knowledge and experience to overcome difficulties which arise.
GENERAL GUIDELINES
Consider first impressions.
Suggest new paths of enquiry to promote browsing.
Avoid dialogue which is prone to user error.
Consider the use of non-verbal forms of information presentation.
Provide quick access
Build in all desired retrieval paths.
Design explanatory dialogue avoiding the need for help facilities.
Draw natural boundaries around available information.
SPECIFIC GUIDELINES
Make politeness an attribute of error messages.
Provide facilities for memory refreshment.
Devise simple, concise and unambiguous error messages.
Draw natural boundaries around lists of valid inputs.
Ask appropriate questions and avoid unnecessary clarification.
Suggest the causes of errors.
DIALOGUE TECHNIQUES
Implement a reset key.
Standardise system response times.
Make the system incorruptible.
Appreciate the virtues and deficiencies of menu selection and look-up lists.
Consider graphic representation of data.

Refer to Sections 6.1 - 6.4 for fuller explanation of the guidelines and techniques.

Three points should be made with regard to the above network;

Firstly, for purposes of clarity, a number of possible connections has been omitted.

Secondly, the network does not purport to be definitive in relation to dialogues for public information systems, and further research would both enlarge and modify it.

Finally, this form of 'structured summary' may be regarded as a source of ready reference for designers who lack the time to carry out comprehensive literature searches.

Certain of the guidelines previously described do not lend themselves to placement within the network. For completeness these are listed:

(a) Involve information experts and general public users, in system development.

(b) Encourage user feedback.

(c) Design the system to be easily adaptable.

(d) Provide additional dialogue facilities for staff associated with the location in which the system is installed.

6.6.4 Supplementary channels for the transmission of dialogue design information

Thus far, it has been proposed that the classification of dialogue design information will render it more adaptable to differing design environments. Moreover the application of a network structure will reveal the logical foundation of recommended guidelines and techniques, which may enhance their acceptability to designers.

However, the survey of Section 6.5 produced the finding that less than half of the sample of designers referred to published guidelines to assist in the creation of interactive dialogues. It must be accepted, therefore, that the restructuring of design information will only affect a proportion of the target population. Thus it is necessary to identify channels of communication in addition to the publication of literature, through which dialogue design principles can be passed on. A second finding of the survey was that, to obtain assistance during interactive program design, high proportions (82%) of subjects employed the techniques of consulting with colleagues and searching for similar programs within software libraries. It is thought that these information sources should be transformed into additional channels for the communication of dialogue principles. They are discussed under the headings 'dialogue design experts' and 'library subroutines based on recommended principles.'

6.6.4.1 Dialogue design experts

Eason et al (1974) have observed that for any group of co-workers, there is the tendency for one individual to become particularly interested in a certain aspect of the groups' tasks. The 'local expertise' which he develops will be acknowledged by the whole group who then refer to him when knowledge of that specific area is required. If this person is given special training in new task procedures, as they arise, he therefore becomes a useful information channel to the other workers. In relation to a group of software designers, it may be of value to select an individual who shows a strong interest in man-computer interaction, for training in the techniques of dialogue design. This person can then pass on his knowledge to colleagues as required.

6.6.4.2 Library subroutines based on recommended principles

There are certain tasks which an interactive program is likely to perform many times such as accepting a numeric input or matching an entered command with an internal list. It is proposed that a series of subroutines should be developed, based on established principles of dialogue design, to perform each of these tasks. They would then be made generally available as library subroutines for incorporation into interactive systems. These facilities may be welcomed by the system designer who would be thus saved the work of developing similar routines himself. Furthermore, having seen these subprograms in operation he may then be influenced to extend the principles upon which they are founded to enhance his own software writing.

Two subroutines which would be widely applicable to the design of interactive systems are now considered.

(1) Inputs of one or more integer numbers.

The technique of displaying a menu list of numbered options is, because of its simplicity, commonly used. The person selects one or more items from the list and types in the appropriate numbers. A library routine to handle these inputs may result in software containing important facilities which the programmer is not prepared to build in himself. During the survey of computer games (chapter 2) it was found that many of the programs halted when a non-numeric character was entered in error for a number. A library routine which accepted numeric values would not be corruptible in this way and should also be politely helpful in pointing out user's mistakes.

e.g.

COMPUTER: "WHICH PUBLIC GARDENS WOULD YOU BE INTERESTED IN VISITING?

(1) BURFORD HOUSE GARDENS	(2) CAMBRIDGE UNIVERSITY GARDENS
(3) GREGORY'S ROSE GARDENS	(4) HIDCOTE MANOR GARDENS
(5) SPRINGFIELDS	(6) TRENTHAM GARDENS

"TYPE IN THE APPROPRIATE NUMBERS SEPARATED BY COMMAS OR SPACES"

(control passes to subroutine)

USER: "@ :/14"

COMPUTER: "NUMBERS TYPED IN SHOULD BE BETWEEN 1 AND 6 PLEASE TRY AGAIN"

USER: ",1 , 4"

(return to main program)

COMPUTER: "HERE ARE THE DETAILS OF BURFORD HOUSE GARDENS AND HIDCOTE MANOR GARDENS....."

The ISLA program contained a numeric input subroutine for which a full listing is given in Appendix IV. In the call statement to this routine it is possible to set a parameter flag which allows the acceptance of a null input. This provides the user with the option to simply press the 'return' key if he requires none of the menu options displayed.

(2) Numeric range.

Where the user is asked to specify a numeric range of values it would be advantageous to allow him to enter this in free format. An alternative approach would entail the display of a cumbersome list of example inputs thus slowing down the interactive process. A comprehensive list appropriate to height range, for instance would consist of the following:

```
5FT 3IN  TO 6FT 3IN
5FT 6IN TO 6FT
4FT 11IN OR OVER
6FT 7IN OR UNDER
APPROX 5FT 8IN
```

Free format, on the other hand releases the user from the burden of fitting his intended range to one of the above examples. It also promotes the use of abbreviations as in:

```
5' - 5'7"
```

A subroutine, developed during the pilot study period to accept single heights in free format, was later extended to receive height ranges. A demonstration program of this is presented in Appendix I. However, a general purpose routine to accept numeric ranges would, ideally, be able to work in a variety of units such as feet and inches, kilos and grams or pounds and pence. The call statement to package would specify the units concerned.

Finally some general comments on free format inputs are made.

Interactive software which allows the operator to express his inputs in an unrestricted form tend to make interaction simpler for the naive user and relieves more experienced users of the task of committing set formats to memory. If the message requesting input is framed such that the domain for answering is highly restricted, computer comprehension of the user should not prove too difficult.

As Palme (1976) states:

"A well designed computer guided interaction can easily use natural or almost natural language, since the messages from the computer are in natural language and since the user's answers are short words which are easy to make the computer interpret".

The time required to develop routines which accept numeric inputs in free format is shorter than might be expected. This was found to be the case during the development of the height range routine. Newman (1978) reports that a final year degree student can design and test a program to accept almost any British format of the date in less than one day.

6.7 SUMMARY AND CONCLUSION

This chapter has presented a set of recommendations for the design of dialogues for on-line public information systems. However, a survey carried out among a group of application programmers indicated that limited use is made of such guidance. This led to the development of a strategy to improve the communication of dialogue design knowledge.

It was considered that recommendations should be classified to meet individual requirements for either general principles or specific suggestions. Also they may be structured to highlight the basis for their proposal.

The problem of communicating with designers who do not consult published literature, could be tackled via the training of dialogue design experts and the provision of library subroutines based upon tried and tested dialogue principles.

Following the development of program routines to handle two fairly specific forms of numerical input (Section 6.6.4.2), the question arises of how difficult the creation of more general software might be. As a test, the author considered the problem of devising a routine to process textual name inputs. A description of this study is related in the succeeding chapter.

CHAPTER 7

ISSUES RELATING TO THE CREATION OF A SUBROUTINE TO ACCEPT TEXTUAL INPUTS FROM NAIVE USERS

It was proposed in Chapter 6 that designers should be provided with a special library of subroutines to be employed within interactive programs. These routines would perform useful and frequently required functions relevant to man-computer dialogues. This chapter discusses the problems associated with the creation of a subroutine to accept inputs of textual names. In the context of public information systems these could, for example, include the names of streets, cinemas, theatres, public buildings or travel destinations. An algorithm called TEX is proposed which may be used as a basis for a piece of software of this kind.

7.1 THE CASE FOR TEXTUAL INPUTS

Prior to discussing the case for dialogues involving textual inputs, it is necessary to list some of the needs of naive users:

(i) They do not wish to spend a long period learning how to use the program beforehand.

(ii) Once using the program, they require short and simple instructions concerning their inputted responses.

(iii) If a user generated error occurs or the computer fails to understand an input then the user requires some system of back-up.

(iv) The user, unfamiliar with the internal logic of the computer program, may be presented with a reply apparently lacking in common sense.

This must be avoided if the user is not to become disillusioned with the program.

There follows a discussion on how user inputs of a textual nature can satisfy requirements (i) and (ii). Later, a system will be proposed for handling textual inputs, and its ability to satisfy (iii) and (iv) will become apparent.

During interaction with a program the naive user is, at various points asked to make choices. For instance, choosing where to branch within a program or choosing attributes for the specification of an enquiry. Where the number of choices is small, a menu selection approach is both simple and easy to understand. The following computer initiated statement is an example:

```
"PLEASE SPECIFY THE VOLTAGE LEVEL
        (1) 0  -  5 VOLTS
        (2) 6  - 15 VOLTS
        (3) OVER 15 VOLTS
 TYPE IN THE APPROPRIATE LINE NUMBER (1,2 OR 3)"
```

However, a large number of choices presents problems. Low speed display devices may require a long period to print out all the choices.

Also the screen of a VDU may not be large enough to contain them. Even with every choice accommodated, a full screen can be a daunting prospect to the inexperienced user.

Thus the input of a textual response without need of choice display can be useful. If presented in a straightforward manner, the computer message asking for user input can be quickly read and understood. For example:

"WHAT STATION DO YOU WISH TO TRAVEL TO? TYPE ITS NAME AND PRESS <RETURN>"

Perhaps the only pre-instruction required would be; "any alphabetic key may be used and the 'RETURN' key must be pressed after each input". Despite the free use of alphabetic keys, it is reasonable to expect most responses to be valid station names.

The computer will contain a "dictionary" of possible responses for matching with and recognising these valid inputs. However, there are a number of problems associated with this process. These are discussed in the following section.

It is assumed that each input consists of only one item of information but that this may contain more than one word.

e.g. "WESTON SUPER MARE"

7.2 PROBLEMS OF RECOGNISING TEXTUAL INPUTS

The problems of matching textual inputs with the computer dictionary can be split up into four main types, which are discussed below. It should be borne in mind that such problems do not necessarily apply to experienced users. Having never, or only occasionally met the program before, the naive user does not know which inputs "will work" or why an apparently nonsensical computer response occurs.

To illustrate the problems, the previous example of railway station names is extended.

7.2.1 Mistyping and misspelling

As Newman (1978) says "Systems are not in general very tolerant of spelling mistakes". It may be argued that naive users will take extra care over inputs and thus make few spelling errors. Conversely the nervousness of such users may precipitate mistyping and there is also no accounting for faulty keys on the keyboard.

Many authors, particularly Gilb and Weinberg (1977) recognise the need for at least some measure of computer tolerance of mistyped and misspelled inputs. The author has observed that, having mistyped an input, the naive user will gain confidence in the interactive process if the computer can recognise it and allow it to be corrected.

7.2.2 Shortened inputs and word order problems

When a user is required to type an alphabetic command without explicitly being told which commands are allowable, then attention needs to be given to the words and the word ordering he might use.

For example the computer dictionary may contain:

"LONDON ST PANCRAS"

but likely shortened inputs are:

"SAINT PANCRAS"
"ST. PANCRAS"
"LONDON"

Concerning word ordering, the dictionary entry may be:

"MANCHESTER PICCADILLY"

but the user is likely to type:

"PICCADILLY MANCHESTER"

Dictionary storage of differing but well used versions of the same item such as "DERRY" and "LONDONDERRY" is a useful approach for quick recognition but only to a limited extent. The dictionary would become too large and expensive of storage and searching if all likely abbreviations, word orderings, mis-typings and misspellings were included. Yet it is reasonable of the user to expect that many such inputs will be recognised.

7.2.3 Computer responses appearing to lack sense

Poorly written interactive programs can cause the computer to respond in what appears to be an unintelligent way. Naive users have no inside knowledge of why particular responses occur and will tend to consider them in human terms. Thus a non-sensical computer response which a human being would not have made is regarded by the user in a poor light.

Two such situations will be discussed regarding textual inputs.

7.2.3.1 Poor input matching

If an input fails to match on any dictionary entry, then displaying a list of similar inputs would be a suitable course of action. The problem is to decide whether a dictionary entry is similar to the input or not. The usual criteria is whether a partial match can be obtained. However, the computer may partially match an entry with the input when the two are quite dissimilar! Presenting the entry to the user in the form "Do you mean....." would, in this case, be undesirable.

Suppose the user typed in:

"TOOTING CENTRAL"

Having obtained a partial match on "TOOTING", it is sensible enough for the computer to respond with:

```
"I DONT RECOGNISE THIS,
 DO YOU MEAN TOOTING BECK?"
```

Alternatively a partial match on "CENTRAL" may be found which might produce the unintelligent reply:

```
"I DONT RECOGNISE THIS,
 DO YOU MEAN FINCHLEY CENTRAL?"
```

Some kind of signal is clearly required which indicates whether a partial match is sensible or not. It is later described within the TEX method how certain words are flagged as 'key elements' so that only matches on these words provide suitable material for clarification with the user. Thus "TOOTING" and "FINCHLEY" would be key elements (or keywords) but "CENTRAL" would not.

7.2.3.2 Unnecessary Inputs.

A program also causes annoyance if it asks the user to repeat an input which he has already given. This may occur in the following way. Having received "BRISTOL" as a destination, the computer may respond:

```
"WHEN YOU TYPED BRISTOL
 WHICH DID YOU INTEND?

    BRISTOL PARKWAY
    BRISTOL TEMPLE MEADS
 TYPE IN THE ONE YOU REQUIRE"
```

Rather than requesting the station name itself, a numeric menu selection mechanism would be both quicker and simpler.

Another example would be where a user has specified his destination, approximate time and day of travel. If he wants to change only one of these attributes, he should not be required to respecify all three.

7.2.4 Input failure without adequate back-up

The problem of unhelpful diagnostic responses to incorrect inputs is long standing.

As Eason (1976b) says, "If the system malfunctions or an error leads to an unusual response by the system, the user can experience the incident as traumatic and it can colour his attitude to the system for a considerable period".

Following an input, the most unhelpful reply is of the form:

"MESSAGE NOT UNDERSTOOD - RETYPE IT"

Chai (1973), Lancaster (1972) and Jones (1978) report this approach being adopted in particular systems they evaluated. The program has assumed that the user knows why his input was rejected and can therefore modify it himself.

The following dialogue illustrates a possible method of back-up on input failure:

COMPUTER:	"WHAT STATION DO YOU WISH TO TRAVEL TO? TYPE ITS NAME AND PRESS <RETURN>"
USER:	"WELLS"
COMPUTER:	"I HAVE IDENTIFIED YOUR INPUT AS WELLINGBOROUGH. IS THIS CORRECT? TYPE Y OR N AND <RETURN>"
USER:	"N"
COMPUTER:	"LETS TRY TO IDENTIFY WHICH STATION YOU REQUIRE. IN WHICH PART OF THE COUNTRY DOES IT LIE? (1) THE NORTH (2) THE MIDLANDS . . . (9) I DO NOT WANT TO CONTINUE"

The main aim should be to avoid a situation in which the user is unable to proceed with or to terminate the dialogue.

7.3 PREVIOUS WORK

A number of papers have been published which propose algorithms relevant to handling textual inputs. Morgan (1970) and Damerau (1964) propose methods for identifying some syntax errors in computer programs.

By "OR-ing" words in a program with a dictionary of the program keywords, they were able to detect single typing or spelling errors.

Muth and Tharp (1977) point out the savings in processing time and storage space by organising the computer's dictionary in the form of a tree structure. Such a structure stores letters separately, each forming a node in the tree. This allows for greater flexibility in matching entries with users' inputs.

Gilb and Weinberg (1977) and Knuth (1973) both describe the Sound Searching Method whereby each dictionary entry is accompanied by a coded key based on the first letter and the sound of three following consonants. Thus a misspelled or mistyped input from a user can be coded into a key which may still match its correct dictionary form. The method works well for phonetically spelled inputs as a wrongly used consonant will often have the same code as the correct one.

"PHYSICS" for instance has the same code as "FIZIXS".

Schek (1977) has devised a sophisticated method for matching common strings of characters in both the user's input and the dictionary entries. Schek's method is one of the few to consider the abbreviation and word order problems described. "LIVERPOOL LIME STREET" could be matched with "LIME STREET LIVERPOOL" on the basis of their two common strings "LIVERPOOL" and "LIME STREET".

7.4 "TEX" - A METHOD FOR RECOGNISING TEXTUAL INPUTS

While the methods of these authors are of interest, they do not address the important problem of giving the user a sensible reply (7.2.3).

A method is required which not only takes this into account but is simple enough for system designers to incorporate into interactive programs fairly easily.

The TEX algorithm attempts to fill this need and to tackle the input recognition problems described. Following its description, an evaluation of TEX is presented based upon the results of a trial which was carried out.

7.4.1 Allowing abbreviations

One further issue should be raised before describing the TEX method. Kennedy (1974) states that users typing in textual commands will quickly want to save time and effort by abbreviating them. Thus "LON" could be an acceptable input to mean "LONDON". It may be argued that a naive user will not require this type of facility. Conversely its inclusion would assist in making the program usable by experienced as well as naive users.

The author shares the latter view and TEX has been designed to allow input abbreviation. The allowable abbreviation for each word can be explicitly set by the system designer.

7.4.2 TEX word recognition procedure

This element in the TEX method is sufficiently complex as to require separate description. The user's input is initially split up into its individual word components and an attempt is made to identify each in the computer's dictionary.

The dictionary itself is arranged in alphabetical order. This example of a four word dictionary illustrates its structure:

WORD LENGTH	WORD	MINIMUM ABBREVIATION
5	ACTON	3
7	ADISHAM	3
6	BRIDGE	3
7	CENTRAL	4

The dictionary words themselves form one column, preceded by a column of the corresponding word lengths. Allowable abbreviations are indicated by the minimum abbreviation column. For example a minimum of 3 leading characters are allowed as an abbreviation of "ACTON". Thus "ACT" or "ACTO" are acceptable.

The minimum abbreviation should be long enough to uniquely identify each word from all others in the dictionary, and to avoid frequent misconstruing of input words. For example, if two leading characters were sufficient to represent "ACTON" then the mistyped input

"AC CRINGTON"

would be taken to mean "ACTON" plus the word "CRINGTON".

The process of identifying the input word commences at an appropriate point in the dictionary. For small dictionaries comparison may start at the beginning. However, this would be inefficient for larger dictionaries. As people rarely get the first letter of a word they are spelling wrong, a suitable approach is to compare the input word with those dictionary words beginning with the same letter.

The word recognition procedure deals with each input word in the following way:

(i) Compare the input, letter for letter with each dictionary word in turn. If an exact equivalent of the input word is found, then it is taken to have been successfully recognised.

(ii) Alternatively the input word may be a valid abbreviation of a dictionary word. If this occurs then similarly, the word is deemed recognised.

(iii) During the matching process a record is kept of any dictionary words which 'almost' match the input word. Having searched through the dictionary, if an input word is not recognised under (i) or (ii) then these closely matching words are displayed to the user. He then selects which, if any, represent the intended word.

A dictionary word is considered to 'almost' match an input word if they differ by no more than one character or by two adjacent characters being interchanged. This will occur if, on entering a word, the user makes one of the four commonest spelling or typing errors. (Muth and Tharp, 1977)

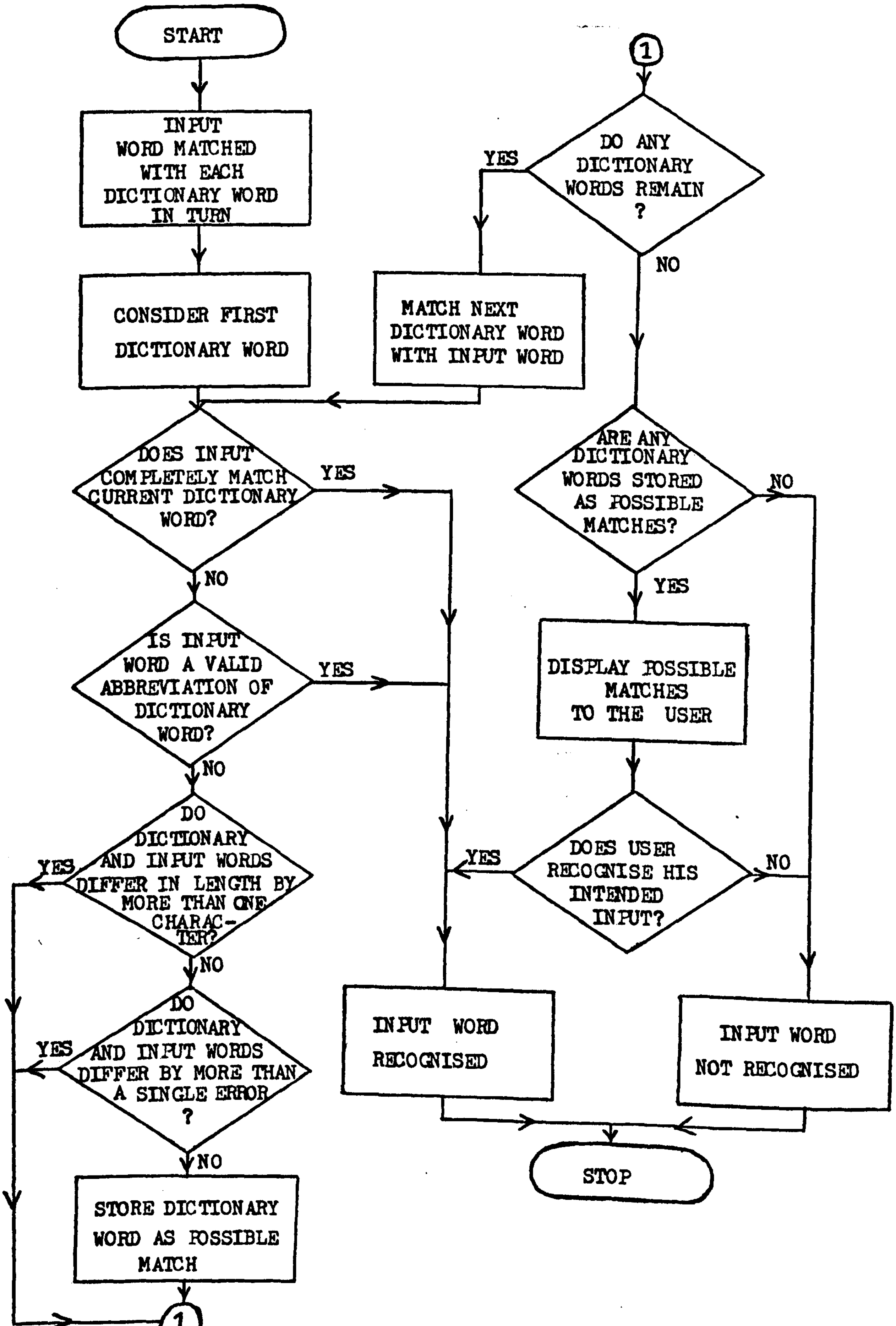
The four versions of "ABERDEEN" illustrate them:

A wrongly typed letter	-	"ABERTEEN"
A missing letter	-	"ABEREEN"
An added letter	-	"ABERDEEEN"
Two adjacent letters interchanged	-	"ABERDENE"

Note that all four versions have either an equal number of letters, one letter more or one letter less than the correct word. Thus dictionary words which differ in length from the input word by more than one letter need not be considered as possible single error forms. This observation was made by Damerau, (1964).

(iv) If the input word is not identified, then the next word is considered. Computer failure to recognise any of the input words will necessitate the use of a back-up system as described in 7.2.4.

Flowchart of the word recognition procedure applied to each input word.



7.4.3 Full TEX procedure

The full TEX method containing the previous word recognition procedure is now described.

The data structure required consists of two parts:

(i) The "DICTIONARY" as previously outlined but with two extra columns headed "COMPONENT" and "POINTERS". The values in these columns provide links to the second data structure.

(ii) The "TEXTUAL INPUT LIST" consisting of two columns. The column "ENTRY" contains the textual inputs which are acceptable to the system. The column "KEY" contains key values associated with each input. Each key is composed from one or two "COMPONENT" values.

Below is an example "DICTIONARY" and "TEXTUAL INPUT LIST" for nine British Rail Stations.

WORD LENGTH	WORD	MINIMUM ABBREVIATION	COMPONENT	POINTERS
3	AND	3	-	-
7	CENTRAL	3	10	-
4	EAST	2	15	-
4	ETON	2	20	2
3	HAM	3	25	1,8
11	HAMMERSMITH	6	30	3
6	HARROW	4	35	4,9
8	HOUNSLOW	3	40	5,6,7
10	WEALDSTONE	5	45	4
4	WEST	3	50	-
7	WINDSOR	3	55	2

TEXTUAL INPUT LIST	
KEY	ENTRY
1525	EAST HAM
2055	ETON AND WINDSOR
0030	HAMMERSMITH
3545	HARROW AND WEALDSTONE
1040	HOUNSLOW CENTRAL
1540	HOUNSLOW EAST
4050	HOUNSLOW WEST
2550	WEST HAM
3550	WEST HARROW

The generalised TEX procedure is embodied in the steps presented below:

(i) The user's input is received and stored as individual words together with their corresponding word lengths, i.e. number of characters.

(ii) Each word is identified in the DICTIONARY using the "word recognition procedure" explained in Section 7.4.2.

(iii) If one or more words can be identified, proceed to (iv). If none are recognised, proceed to (x).

(iv) For each identified word, store the equivalent COMPONENT and POINTERS (if present) in separate lists. Discard any duplicates in either list. Having processed each word, proceed to (v).

(v) The stored POINTERS point to possible interpretations of the user's input in the TEXTUAL INPUT LIST. From these it may be possible to select the correct interpretation. Firstly, however, the following four conditions are tested in the given order. If any one of them holds then different action is taken.

- if no POINTERS stored, go to (x)
- if any input words were unrecognised, go to (ix)
- if one POINTER stored, go to (viii)
- if more than two COMPONENT values stored go to (ix)

If none of the conditions hold, proceed to (vi).

(vi) If only one COMPONENT value is stored, create a second, of value zero. Combine the two COMPONENT values to form a unique 'MATCH' key. Derive MATCH by multiplying the smaller COMPONENT by 100 and adding the larger. Thus 20 and 55 become 2055.

(vii) Compare MATCH with the KEY values of the possible interpretations in the TEXTUAL INPUT LIST. If a match is found, retain the POINTER to that interpretation and proceed to (viii).
If no match is found proceed to (ix).

(viii) Display the character string in ENTRY indicated by the single POINTER, for user clarification. If the user's input has been correctly recognised, the procedure halts. Otherwise proceed to (x).

(ix) The list of possible interpretations cannot be reduced. Display them all to the user. If the user identifies the one he requires, then the procedure halts. Otherwise proceed to (x).

(x) The TEX procedure has not succeeded in recognising the user's input. A back-up system is now utilised which displays a numbered list of the input options available. The user then types in the number of his selected option. The list should include an escape option allowing the user to discontinue the current path of interaction.

Clarification of stage (viii) may be considered unnecessary in certain circumstances. In this case the wording for (viii) should read: "The user's input has been recognised as the character string in ENTRY indicated by the single POINTER. The procedure halts."

The mechanism whereby TEX selects sensible interpretations of the user's input from the TEXTUAL INPUT LIST is embodied in the dictionary POINTERS.

Consider the input "HAM". The equivalent dictionary entry contains pointers to the sensible interpretations, "EAST HAM" and "WEST HAM" in the TEXTUAL INPUT LIST.

Now consider the input "FINCHLEY CENTRAL". "FINCHLEY" will be unrecognised in the dictionary, but "CENTRAL" does have an equivalent entry. However, there are no associated pointers since no suitable interpretation can be made.

7.4.4 Illustration of TEX

To illustrate the TEX procedure, three example inputs are considered in the table overleaf. The previous DICTIONARY and TEXTUAL INPUT LIST are used as a basis for the input options available.

Procedure Steps			
(i) User's input	"HAMERSMITH"	"HAM"	"EAST HOUNSLOW"
(ii) Word recognition procedure	Computer: "DO YOU MEAN HAMMERSMITH?" User: "YES"	"HAM" RECOGNISED	"EAST" recognised "HOUNSLOW" recognised
(iii) Any words identified?	Yes, go to (iv)	Yes, go to (iv)	Yes, go to (iv)
(iv) Store COMPONENT and POINTERS for each word. (Each pointer indicates a possible interpretation of the input)	COMPS. POINTERS 30 3	COMPS. POINTERS 25 1 8	COMPS. POINTERS 15 5 40 6 7
(v) Test following conditions before reducing list of interpretations: - No pointers stored? - Any unrecognised input words? - Only one pointer stored? - More than two components?	Third condition holds Go to (viii)	None of these conditions hold Go to (vi)	None of these conditions hold Go to (vi)
(vi) Create MATCH value	-	MATCH = 0025	MATCH = 1540

(continued)

Procedure Steps			
(vii) Compare MATCH with KEY values of possible interpretations	-	MATCH not equal to 1525 EAST HAM MATCH not equal to 2550 WEST HAM Go to (ix)	MATCH not equal to 1040 HOUNSLOW CENTRAL MATCH equals 1540 HOUNSLOW EAST Go to (viii)
(viii) Display single interpretation to user	Computer: "IS HAMMERSMITH CORRECT? User: "YES"	-	Computer: "IS HOUNSLOW EAST CORRECT?" User: "YES"
(ix) Display all interpretations to user	-	Computer: "WHICH IS CORRECT? (1) EAST HAM (2) WEST HAM (3) NEITHER" User: 2	-
(x) Utilise back-up system	-	-	-

7.4.5 Evaluation

A computer program incorporating the TEX method was written to assess its level of performance. The program accepted typed inputs through a computer terminal and attempted to match them with a list of the names of 100 British football teams. The name or names associated with each input were then displayed to the user. Failure to associate any name with an input resulted in the output of a message to that effect. (In a practical system, such inputs would then be handled by a back-up system).

The program was run on a Burroughs 6700 main frame computer. The TEXTUAL INPUT LIST, containing 100 team names, and the DICTIONARY, consisting of 116 words, were read into arrays from backing storage at the start of the program. Thus during the assessment, all necessary data was held in core storage.

Ten subjects took part in the evaluation experiment, all having previously used a computer terminal but not the program itself. They also professed to having limited knowledge about the application area. Each subject was asked to interact with the program by typing in the names of ten teams expressed verbally by the experimenter. The aim was for the computer to recognise as many of the ten as possible. Each subject was given a different set of names so that all 100 stored in the TEXTUAL INPUT LIST were tested. Where possible, team names were shortened to the city or town of location. For example "CREWE ALEXANDRA" was verbally expressed as "CREWE". Then if a subject did not know the full name, he was forced to enter the incomplete form.

7.4.5.1 Results relating to recognition of inputs.

The TEX method successfully recognised 88 out of 100 inputs. The following table shows the proportion of inputs correctly and incorrectly formed, and the proportion with and without spelling or typing errors. The figures in brackets indicate the number of inputs recognised by the computer within each cell.

	With typing or spelling errors	Without errors	Totals
Correct form	35 (35)	13 (7)	48 (42)
Incorrect form	41 (36)	11 (10)	52 (46)
Totals	76 (71)	24 (17)	100 (88)

It can be seen from the table that 52 of the inputs were in an incorrect form and 76 were mistyped or misspelled. The high proportions which these figures represent gives evidence to support the view that incorrectly formed and mistyped inputs should not simply be rejected by the computer and that there is the need for a system such as TEX to take account of input variations.

Consideration of the twelve unrecognised inputs is instructive of deficiencies in the method.

Five inputs contained two or more typing and spelling errors. As explained in the description, TEX is limited to the recognition of words containing single errors. Extending the method would be possible but at the expense of increased complexity and computer processing times.

Four inputs used acceptable name versions not previously stored e.g. "SPURS". The inclusion of extra DICTIONARY entries would meet this problem.

Two inputs were unrecognised because of word concatenation. "WESTHAM" was one such example. The computer, considering this as a single word could not match it with either "WEST" or "HAM" in the DICTIONARY. The section on future work (7.5) suggests a possible solution to this.

Finally within the input "MAN CITY", the abbreviation "MAN" was intended to mean Manchester. This was rejected by the computer as it may equally well have referred to the Mansfield team. In practise, however the term "MAN" is widely used in reference to Manchester teams. It should therefore be left to the systems designer to decide whether ambiguous terms may or may not form valid abbreviations.

7.4.5.2 Results relating to processing times

The average computer processing time per input was 0.11 seconds. As the pointers facilitate random access to the TEXTUAL INPUT LIST, search times should remain reasonably constant regardless of list size.

However, the word recognition procedure requires serial searching. Thus processing times will increase with the size of DICTIONARY. In the experiment, the search for each input word was started at the first DICTIONARY entry. For larger DICTIONARIES, the approach of assuming the first letter of each input word to be correct and restricting comparison to dictionary words of the same letter would be more efficient. It is of interest to note that all 117 input words typed in during the experiment were prefixed by the correct letter.

7.4.6 Implementation

The interested reader may experiment with the TEX method by loading the FORTRAN program presented in Appendix V and running it through some on-line keyboard device. The TEXTUAL INPUT LIST contains only nine input options which is far too few to enable a formal test to be carried out. To increase the number it will be more practical to adapt the program to access the DICTIONARY and TEXTUAL INPUT LIST from an on-line disk file rather than from DATA statements in the program itself. The program in Appendix V should be sufficiently well documented with comments to allow this modification to be made.

7.5 FUTURE WORK ON THE TEX METHOD

Firstly, it would be of interest to investigate the advantages and disadvantages of introducing an additional stage into the word recognition procedure. At present, an input word containing two or more errors, or two words which become joined, will not be recognised. When this occurs, a sound matching procedure (such as Knuth's or Gilb and Weinberg's) could be put into operation. A sound code is created for the unrecognised word and compared with equivalent, stored codes for the dictionary words. Those with matching sound codes would then be presented to the user for clarification.

Secondly, for systems used by both naive and experienced operators, the latter group will wish to short cut familiar paths in the interactive dialogue.

Martin's (1973, p.314) "response-combining" technique mentioned in the introductory chapter is one means of providing short cuts which could be built into the TEX system.

As the user becomes more experienced he shortens the dialogue by entering sequences of inputs which reply in advance to computer requests for data.

For example, in response to the question:

"WHAT INFORMATION DO YOU REQUIRE? - FLIGHT, RAIL, BUS"

the user may type:

"RAIL + TO + LEEDS + SATURDAY"

This pre-empts further questions relating to the train journey itself and the information which is required can be supplied immediately. Abbreviation could, further reduce the user's input.

e.g. "R + T + LEE + SAT"

This technique avoids the need to question the user at the start of each interactive session as to his current level of experience.

7.6 CONCLUSION

Methods for recognising textual inputs from naive users should display a certain amount of 'intelligence' without being over complex. Gilb and Weinberg, (1977) say to prospective designers "Don't over complicate the problem. Allow for a small percentage of human-computer interaction and don't go overboard in the quest for a perfect system".

The system outlined in this chapter aims to promote practical ideas for incorporation into an interactive computer system without discouraging a system designer by being too sophisticated and too expensive in computer time.

CHAPTER 8

PROJECT REVIEW AND SUGGESTIONS FOR FURTHER RESEARCH

This project has been directed towards the problems of creating effective man-computer dialogues for use with full keyboard display devices.

Published material intended to provide general advice on interface design was often found to be conflicting or in need of further qualification. This appears, in part, to be a consequence of authors developing guidelines from particular applications which they have studied and offering them as generally applicable. In response to this, the investigation concentrated upon identifying principles related solely to computer based information systems for untrained and inexperienced users.

A survey of 'self-explanatory' computer games supported the widely held view that dialogue interfaces are often poorly designed. Many of these games contained fundamental defects such as failing to provide either an introduction, statement of objective or set of rules for play.

8.1 RESULTS AND CONCLUSIONS OF THE PROJECT

The following three sub-sections recall the main results and conclusions arising from the study. The section headings are:

(a) Findings regarding the design of experiments.

(b) Findings concerning the design of on-line information systems.

(c) The results of a study into the utilisation of dialogue design knowledge.

8.1.1 Findings regarding the design of experiments

During the period of pilot study, experiments were carried out to determine the problems of writing interactive software and to compare a number of dialogue styles. The results are set out in Chapters 3 and 4.

The most important outcome was the realisation that the methods employed for conducting experiments were subject to a number of artificial conditions which would not apply in a 'real world' environment. These are listed below:

(1) Subjects were invited to take part in computer interaction rather than doing so as a result of their own initiative.

(2) Subjects were informed of the information to be retrieved rather than searching for data of personal interest to them.

(3) The presence of the experimenter, acting as an observer of proceedings, may have affected subjects' behaviour.

(4) The selection of subjects from an academic environment produced samples which were unrepresentative of the general population of novice users.

Clearly, to gain more accurate results, these conditions had to be negated.

At this point the study was narrowed down to consider general public users rather than all naive users. An experiment was performed based upon a local information system (ISLA) installed for public use at Leicester's Central Information Bureau. (See Chapter 5) This allowed the removal of conditions 1, 2 and 4. During the initial part of the experiment, discrete observation of proceedings was attempted but then abandoned due to the inhibiting effect on users. Thus condition 3 was also rescinded.

In the absence of direct observation, alternative means were adopted to obtain feedback on system usage. These are now briefly appraised on the basis of the project's findings.

(1) Incorporating tests into the system.

Certain sections of dialogue may lend themselves to testing particular aspects of man-computer interaction. However, care should be taken that the incorporation of the test does not render the dialogue inappropriate to the user's needs.

A useful technique is for the computer to switch between two contrasting dialogue styles for alternate interactive sessions. For each style the computer records specific items such as user response times and error rates thus allowing the two interfaces to be compared.

(2) Recording system inputs and outputs.

This technique can assist in achieving a number of aims such as isolating system errors, tracing the routes taken to reach desired information and identifying areas of user difficulty. It is necessary to record all inputs to the system so that dialogue sessions can be reconstructed. However, computer outputs may be abbreviated to avoid the generation of large quantities of transcription data.

(3) Eliciting user's comments.

This may be achieved off-line by the use of a comments book or on-line through the computer terminal. The former method will tend to produce fuller comments since writing ability is usually superior to typing ability. The latter has the advantage that comments can be more easily linked with the log of system interaction.

8.1.2 Findings concerning the design of on-line public information systems

As a result of the main ISLA study, a number of guidelines were developed for designers of public information systems. A re-evaluation of published recommendations was also performed to establish their relevance in this context. These details were presented in the first four sections of Chapter 6.

Certain of the findings regarding the usage of the ISLA system were of particular interest:

(1) Users will not spend a long time at the terminal.
(2) There was a general desire for a faster data display rate.
(3) The suggestion was made that the introductory display should be shortened.

These three all indicated a wish to retrieve information from the system as quickly as possible.

A second requirement expressed by users was that the number of topics and the depth in which they were covered should be increased.

It is thought that for the system to be of practical value, both of these requirements must be satisfied. Unfortunately, as design aims, they are also contradictory. The designer may, for instance, create a system which links the user to his target information within a small number of interactive stages. However, this implies that the database is modest in the range of information which it holds. If, on the other hand, the intention is to develop a system which can offer a comprehensive service, this is likely to be at the expense of the user embarking on a long process of interaction.

Clearly the chances of success of a public information system would be greatly enhanced if it could provide both fast access and a comprehensive service. It appears necessary therefore to investigate ways in which these conflicting goals can be reconciled.

8.1.3 Results of a study looking at the use made of dialogue design knowledge

A survey, carried out on a sample of application programmers, revealed a significant failure to utilise published guidelines during the creation of interactive software. (See Section 6.5.) This led to the conclusion that a strategy was required to improve the communication of dialogue design knowledge. It was considered that design information be classified along a continuum ranging from general theories to highly specific dialogue techniques. This would make classes of material available, tailored to the various conditions under which computer systems are developed.

Alternative channels for the transmission of design knowledge were also proposed. These were:

(a) via trained specialists and

(b) via the proliferation of software support libraries based upon sound dialogue principles.

A final study enabled an algorithm to be developed for handling inputs of textual names. (See the TEX method, Chapter 7) This demonstrated the possibility of creating subroutines to assist the dialogue designer.

8.2 DIRECTIONS FOR FURTHER RESEARCH

Four paths of research relating to dialogues for naive users are described. These concern:

(a) The development of a sound methodology for conducting experiments.

(b) Further investigation into the design of on-line public information systems.

(c) An extension of the survey regarding the use made of dialogue guidelines and an evaluation of the proposed strategy for communicating dialogue design knowledge.

(d) An examination of the social effects of computerised information systems for the public.

8.2.1 The development of sound methodology for conducting experiments

It has been shown that one of the concerns of the project has been to improve upon the experimental methods employed during the two pilot studies. An event within the main ISLA study is now recalled which further questions the experimental approach.

It was hypothesized that the insertion of pauses during the output of a lengthy textual passage would alleviate any feelings, on the reader's part, of being overwhelmed with material. A section of dialogue was designed to test this hypothesis. Little consideration, however, was given to the method of displaying passages, and it was decided that printing would take place from the base of the VDU screen. This gave the text a gradual upward movement which, as a result, made it more difficult to read. It was found that the pauses did not simply divide the text into smaller portions but also served to halt this distracting upward movement. Thus the display mechanism invalidated the intended test. Had this flaw been recognised initially, then a successful experiment could have been designed. (Reference Section 5.7)

Work is needed to determine sound methods of experimentation in the man-computer interaction field. Innocent (1978) advocates an unplanned approach to experimental design so that all relevant processes can be identified. Once this has been achieved, formal experimentation can take place based upon a knowledge of which variables are of interest to study and which features must be designed into the experiment if valid results are to be achieved. In the example above, a preliminary experiment would have detected that output from the base of the screen was a distorting factor and shown that printing from the head of the screen was required to neutralise this.

8.2.2 Further investigation into the design of on-line public information systems

As discussed in Section 8.1.2, the designer of a public information system faces two conflicting requirements; that information can be obtained as quickly as possible and that a comprehensive service is provided. If these aims can be reconciled, the value of the system will be considerably enhanced.

Since the idea of a comprehensive service implies that the user may need to take part in a lengthy sequence of interactions, one strategy is to improve the efficiency of the man-computer communication channel. Three approaches for achieving this are noted:

(1) Computer generated voice output.

A number of commercially available products such as the 'speak and spell' educational toy and vocal computer chess games demonstrate that a computer can generate a good imitation of the human voice. The application of this technique to support computer output on a display screen could improve communication by stimulating the user's attentiveness towards his task. Less time would be required to scan the information appearing on the screen and quicker user responses would result. One of the main design problems would be to determine a suitable relationship between the output of speech and text. If this was not achieved, the addition of the voice could hinder rather than assist the reading of the display.

(2) Computer recognition of voice input.

Another method of speeding up the interactive process is to allow user inputs to be entered vocally rather than via a keyboard. If voice input were to be applied to a public information system, a major difficulty would be to provide the computer with an ability to process the many different tones of voice, styles of speech and regional accents which it would receive. One technique for overcoming this problem is to require each user to recite a list of words prior to interaction so that the computer can become orientated to the current speaker's voice. However, for public systems where the user wishes to embark upon interaction without delay, this calibration procedure would need to be condensed into a very short period of time.

(3) Computer understanding of natural language input.

It may be argued that if the computer can be made to accomodate to the user by conversing in his own language then dialogue between the two could be conducted in a very efficient manner. In many cases this would be true since the user can frame enquiries to the system in terms which conform exactly to his own conception of them. However, this advantage must be balanced against the possibility that a number of enquiries will not be understood by the system. Furthermore, if natural language queries were to be submitted via a keybaord, this would involve the user in typing lengthy textual sequences which may neutralise any advantage in speed of interaction over structured dialogues requiring briefer inputs.

Ballantine (1979) has reviewed recent developments and future possibilities within these three important research areas.

Clearly these approaches for enhancing a public information facility would currently be very expensive to employ. However, the rapidly falling cost of computer power indicates that a preparatory consideration of their role within such systems would be a useful undertaking.

8.2.3 An extension of the survey concerning the use made of dialogue guidelines and an evaluation of the proposed strategy for communicating dialogue design knowledge.

The finding that referral to published guidelines is not standard practise during the programming of dialogues, was based upon a small sample of eleven application programmers working in a polytechnic computer centre. While this provides an indication of the behaviour of programmers in a particular environment, no conclusions can be drawn regarding software designers in general. If the finding is correct, therefore, it is necessary to strengthen the evidence by extending the survey to cover a larger sample and a broader spectrum of programming environments. It would be of particular interest to study the practises employed within computer manufacturing companies and software houses.

A strategy was proposed for the effective communication of dialogue design knowledge. (Section 6.6.) This consisted of the information being classified in a suitable manner, presented in the form of structured summaries and communicated through a variety of channels. An evaluation of this proposal could be achieved by implementing it within a programming community. If the experiment was successful, then it may be possible to generalise the strategy to cover further aspects of software design such as program efficiency, portability and adaptability.

8.2.4 An examination of the social effects of computerised information systems for the public

The widespread installation of computerised public information facilities would have the potential of being used by very large numbers of people and therefore of producing certain consequences for society as a whole. Opinions will differ as to which of these developments are beneficial and which are harmful. However, it is essential to determine the nature and full range of consequences so that a sensible debate can take place and the most clearly undesirable consequences be avoided by careful system design.

A number of possible consequences are suggested;

8.2.4.1 Increasing the acceptance of computer devices

Despite the growing number of technologically advanced products, it appears that the public remains generally uneasy and suspicious of them (Griffin, 1980). This presents a barrier to the acceptance of computer controlled devices which require personal contact with the user such as multi-purpose home terminals assisting with, for example, household management, education or entertainment. Well designed public information systems could serve to allay these suspicions and smooth the path for other labour saving devices to be accepted.

8.2.4.2 Bridging the gap between the technologically competent and incompetent

Eason (1980) indicates the possibility that the availability of computer services could create division in society. One group would be the 'technologically competent' who have some type of technical background and would feel comfortable operating a computer terminal. The other group would consist of the 'technologically incompetent'; people without technical skills or aptitudes. In Eason's view:

> "The (technologically incompetent) may well become seriously disadvantaged if many valuable (computer) services are only available to the technologically competent and this could sow the seeds of social unrest."

However this project has shown that a public information system, if well designed, can be tailored to suit the capabilities of all user groups regardless of their expertise. If such systems are implemented, this could help to prevent divisions in society from occurring.

8.2.4.3 Providing answers to specific questions

It is reasonable to argue that if public computer systems can provide information in a more convenient and efficient manner than via traditional methods, then a more informed society will result.

Ivergard (1976) draws attention to the differences between this type of system and more traditional one-way forms of communication such as radio, television, books and newspapers. In these cases the individual selects from these output streams, the elements of knowledge which he requires. During this process he will absorb a great deal of miscellaneous information. A facility such as a public information system, however, has the potential to provide highly specific answers to user's queries. Ivergard extends the warning that:

> "A society in which we have a continuous personal choice as to what information we receive runs the risk of becoming very conservative."

He then suggests that such a society may be deprived of the stimulation of unrequested information which one-way broadcasting systems confer.

8.2.4.4 Centralisation of information sources

In the interest of developing the most cost-effective information system, the decision may be taken to centralise the source of information supporting the service. The consequences of data corruption could thus become magnified since errors in the database will misinform a large percentage of the population. This corruption could be either an accidental occurrence or the result of a deliberate act. The latter case would be particularly likely within societies where the media is subject to strict government control and censorship.

8.2.4.5 Conclusion - The importance of good dialogues

Consideration of the suggested social consequences reveals that any benefits which the general public obtain from information systems will depend upon the quality of the interactive dialogues which they provide. A system will only become accepted by the public if the dialogue provides them with the ability to use the system easily and in a pleasant manner. If the system is not accepted by one or more sections of the public, this may enlarge any divisions caused by differences in peoples technical expertise. Poor dialogues will also prevent people from utilising the system to its full capacity and benefiting fully from the broad spectrum of information which it can provide.

In a recent book which seems destined to be very widely read, Toffler (1980) paints a picture of an information society of tomorrow which summarises the trends that many people are now recognising; more people working from home, the collapse of the work ethic, opposition to the expert, greater two-way communication and participation in democratic processes at all levels. He sees a world in which citizens operate from 'electronic cottages'. Most of the trends he identifies depend for their successful implementation upon the general public being able to communicate electronically with computers and with one another. Unless the problems revealed in these studies are solved, the information society of which he and many others dream may be a non-starter.

REFERENCES

BALLANTINE, M. (1979)
"Conversing with computers - the dream and the controversy". Ergonomics Soc. Conf. on Man-Computer Communication at Loughborough University of Technology. Dec. pp. 117-132.

CHAI, D.T. (1973)
"An information retrieval system using keyword dialogue." Information Storage and Retrieval. Vol. 9, pp. 373-387.

CHERITON, D.R. (1976)
"Man-machine interface design for time-sharing systems." Proc. ACM National Conference. pp. 362-380.

DAMERAU, F.J. (1964)
"A technique for computer detection and correction of spelling errors." Comm. ACM Vol. 7, No. 3, pp. 171-176 (March).

EASON, K.D., DAMODARAN, L. and STEWART, T.F.M. (1974)
"A survey of man-computer interaction in commercial applications." LUTERG Report No. 144. October. Department of Human Sciences, Loughborough University of Technology.

EASON, K.D. (1976a)
"A task-tool analysis of manager-computer interaction." Proc. of NATO ASI Conference on Man-Computer Interaction. ed. B.Shackel, Loughborough University.

EASON, K.D. (1976b)
"Understanding the naive computer user." Computer Journal. Vol. 19, No. 1, pp. 3-7.

EASON, K.D. (1979)
"Dialogue design implications of task allocation between man and computer." Ergonomics Soc. Conf. on Man-Computer Communication, Loughborough University pp. 43-57, (Dec.)

EASON, K.D. (1980)
"The potential of computing for the public." Infotech State of the Art Report: "The Non-Expert User" Zurich, September 1980.

EDMONDS, E.A. (1974)
"A process for the development of software for non-technical users as an adaptive system." General Systems. Vol. 19, pp. 215-217.

EDMONDS, E.A. (1978)
"Adaptable man/machine interfaces for complex dialogues." Proc. Eurocomp 1978.

EDMONDS, E.A. (1981)
"Adaptive man-computer interfaces." Computing skills and the user interface." eds. M.J. Coombs and J.L. Alty. Academic Press. To appear.

EVANS, C.R. et al, (1970)
"On-line interrogation of hospital patients by a time-sharing terminal with computer/consultant comparison analysis." IEE Conf. on 'Man-Comp. Interaction,' Nat. Phys. Lab. Teddington, Eng. (Sept.)

EVANS, C.R. (1976)
"What I have learned about computer communication and what computers have taught me about human communication." Advanced Study Institute Conf. on Man-Computer Interaction. Mati, Greece (Sept.)

FITTER, M. (1979)
"Towards more 'natural' interactive systems." Int. Journal of Man-Machine Studies Vol. 11 pp. 339-350.

FOLEY, J.D. and WALLACE, V.I. (1974)
"The art of graphic man-machine communication." Proc. IEEE Vol. 62 No. 4 pp. 462-471 (April).

GAINES, B.R. and FACEY, P.V. (1975)
"Some experience in interactive system development and application." Proc. IEEE Vol. 63 No. 6 pp. 894-911 (June).

GILB, T. and WEINBERG, G.M. (1977)
"Humanised Input-techniques for reliable keyed input." Tolerating Variance - Chapter 8. Winthrop Publishers, Mass.

GOODMAN, T. and SPENCE, R. (1978)
"The effect of system response time on interactive computer-aided problem solving." Proc. ACM pp.100-104.

GRIFFIN, A. (1980)
"Attitudes towards microprocessor technology." Unpublished Final Year Project. Department of Human Sciences, Loughborough University.

GROSSBERG, M., WIESEN, R.A. and YNTEMA, D.B. (1976)
"An experiment on problem solving with delayed computer response."
IEEE Trans. on Systems, Man and Cybernetics Vol. 6 No.3 pp. 219-222 (March).

HALL, P.A.V. (1978)
"Man-computer dialogues for many levels of competence.
SCICON internal report.

HANSEN, W.J. (1971)
"User engineering principles for interactive systems."
AFIPS Conf. Proc. Vol. 39 pp. 523-532.

HARRIS, L.R. (1977)
"User orientated database query with the ROBOT natural language query system." Int. Journal Man-Machine Studies, No. 9 pp. 697-713.

HEBDITCH, D.L. (1973)
"Terminal languages for database access." Infotech State of the Art Report on Databases.

HONOLULU POLICE SYSTEM (1970)
"On-line data processing at the Honolulu Police Department."
IBM Manual K20 - 0278 White Plains N.Y.

INNOCENT, P.R. (1978)
"An experimental approach to man-computer interaction research."
Man Computer Interaction Research Group Report No. 15, Dept. Mathematics, Statistics, and Computing, Leicester Polytechnic.

IVERGARD, T.B.K. (1976)
"Man-computer interaction in public systems" NATO Advanced Study Institute Conference on Man-Computer Interaction. Mati, Greece (Sept.)

JONES, P.F. (1978)
"Four principles of man-computer dialogue." Computer Aided Design (G.B.) Vol. 10, No. 3, pp. 197-202 (May).

KENNEDY, T.C.S. (1974)
"The design of interactive procedures for man-machine studies."
Int. Journal of Man-Machine Studies Vol. 6, pp. 309-334.

KENNEDY, T.C.S. (1975)
"Some behavioural factors affecting the training of naive users of an interactive computer system." Int. Journal of Man-Machine Studies Vol. 7, pp. 817-834.

KNUTH, D.E. (1973)
"The art of computer programming." Vol. 3 pub. Wesley pp. 391-392.

LANCASTER, F.W., RAPPORT, R.L. and KIFFIN PENRY, J. (1972)
"Evaluating the effectiveness of an on-line natural language retrieval system." Information Storage and Retrieval. Vol. 8, pp. 223-245.

MARTIN, J. (1973)
"Design of Man-Computer Dialogues." Prentice-Hall, Englewood Cliffs N.J.

MEISTER, D. and FARR, D.E. (1967)
"The utilization of human factors information by designers." Human Factors pp. 71-87. Vol. 9.

MILLER, L.H. (1977)
"A study in man-machine interaction." AFIPS Conf. Proc. Vol. 46 pp. 409-421.

MILLER, R.B. (1968)
"Response time in man-computer conversational transactions." AFIPS Conf. Proc. Vol. 33 pp. 267-277.

MORGAN, H.L. (1970)
"Spelling correction in systems programs." Communications of the A.C.M. Vol. 13. No. 2 Feb. pp.90-94.

MUTH, F.E. and THARP, A.L. (1977)
"Correcting human error in alphanumeric terminal input." Information Processing and Management Vol. 13. pp.329-337.

NEWMAN, I.E. (1978)
"Personalised user interfaces to computer systems." On-line '78 Conference.

NICKERSON, R.S. (1969)
"Man-computer interaction: a challenge for human factors research." IEEE Transactions on Man-Machine Studies, pp.164-180.

PALME, J. (1976)
"Interactive software for humans."
Management Datamatics Vol. 5 No. 4 pp. 139-154.

PALME, J. (1979)
"Increasing user influence on computer systems." In
"Communicating with Computers." (eds) M. Sime and M. Fitter.
Academic Press.

ROETHLISBERGER, F.J. and DICKSON, W. (1939)
"Management and the Worker." Harvard University Press.

SCHEK, H.J. (1977)
"Tolerating fuzziness in keywords by similarity searches."
Kybernetics. Vol. 6 pp. 175-184.

SCHNEIDERMAN, B. (1979)
"Human factors experiments in designing interactive systems."
Computer - IEEE Computer Soc. Publication Vol. 12, No. 12,
pp. 9-19.

SEGAL, B.Z. (1975)
"Effects of method of error interruption on student performance
at interactive terminals." Univ. of Illinois Dept. of Comp.
Sci. Tech. report. UIUCDCS-R-75-727 (May).

SHACKEL, B. (1979)
"Dialogues and Language - Can computer ergonomics help?"
Ergonomics Soc. Conf. on Man-Computer Communication at
Loughborough University of Technology. Dec. pp. 4-42.

SPENCE, (1976)
"Human factors in interactive graphics." Computer Aided
Design. Vol. 8 pp. 49-53.

STEWART, T.F.M. (1976)
"Displays and the software interface." Applied Ergonomics
Vol. 7 No. 3 pp. 137-146.

STEWART, T.F.M. (1979)
"Communicating with dialogues." Ergonomics Soc. Conf. on
Man-Computer Communication at Loughborough University of
Technology. pp. 77-95 (Dec.)

SUNDAY TIMES (1980)
Colour magazine for 24th Feb.

TINKER, M.A. (1963)
"Legibility of print."
Iowa State University Press.

TOFFLER, A. (1980)
"The third wave", Collins.

WALTHER, G.H. and O'NEILL, H.F. (1974)
"On-line user computer interface - the effect of interface flexibility, terminal type, and experience on performance."
AFIPS Conf. Proc. Vol. 43 pp. 379-384.

WASSERMAN, A.I. (1973)
"The design of idiot-proof interactive systems."
AFIPS Conf. Proc. Vol. 42 pp. M34-M38.

WEIZENBAUM, J. (1966)
"ELIZA - a computer program for the study of natural language communication between man and computer." CACM Vol. 9 No. 1 pp. 36-45.

APPENDIX I

PROGRAM TO DEMONSTRATE THE ACCEPTANCE OF OF HEIGHT RANGE INPUTS IN FREE FORMAT AND SAMPLE PROGRAM RUN

```
0010 REM *      HEIGHT RANGE INPUT DEMONSTRATION PROGRAM
0020 REM *      ========================================
0030 REM * THIS BASIC PROGRAM DEMONSTRATES A METHOD FOR
0040 REM * ACCEPTING HEIGHT RANGE INPUTS IN FREE FORMAT
0050 REM * AND MAY BE USED AS A BASIS FOR THE DEVELOPMENT
0060 REM * OF A SUBROUTINE TO HANDLE NUMERIC RANGES WITH
0070 REM * VARIOUS UNITS.
0080 REM * INPUTS ARE ANALYSED USING A SYMBOL STATE TABLE
0090 REM * AS DESCRIBED IN CHAPTER NINE OF 'FORTRAN
0100 REM * TECHNIQUES' BY COLIN DAY, CAMB. UNIV. PRESS.
0110
0120 DIM W$(20),A$(26),D$(60),D(60),T(40,10),C(12),N(4),L(6)
0130 FOR I=1 TO 26
0140 REM * "A$" CONTAINS THE A-Z ALPHABET FOR IDENTIFYING
0150 REM * LETTERS IN THE USER'S INPUT.
0160 READ A$(I)
0170 NEXT I
0180 REM * "D$" FORMS THE DICTIONARY OF KEYWORDS/NUMBERS
0190 REM * AND "D" CONTAINS THE EQUIVALENT CODES RELEVANT
0200 REM * TO THE RANGE SPECIFICATION.
0210 READ D0
0220 FOR I=1 TO D0
0230 READ D$(I),D(I)
0240 NEXT I
0250 REM * "T" STORES VALUES WHICH MAKE UP THE SYMBOL
0260 REM * STATE TABLE.
0270 MAT T=ZER
0280 READ T0
0290 FOR I=1 TO T0
0300 READ X,Y,Z
0310 T(X,Y)=Z
0320 NEXT I
0330
0340 REM * THE USER ENTERS HIS INPUT INTO "I$"
0350 PRINT
0360 PRINT "TYPE IN THE HEIGHT RANGE IN FEET AND INCHES"
0370 PRINT "AND THEN PRESS <RETURN>"
0380 PRINT
0390 INPUT I$
0400 REM * AN INPUT OF ZERO TERMINATES THE PROGRAM.
0410 IF I$="0" GOTO 2670
0420
0430 REM * THE FOLLOWING SECTION IDENTIFIES SEQUENCES OF
0440 REM * LETTERS, NUMBERS AND MISCELLANEOUS CHARACTERS
0450 REM * WITHIN THE USER'S INPUT AND STORES THEM IN "W$".
0460 REM * "B" RECORDS THE STARTING POSITION OF THE
0470 REM * CURRENT STRING OF CHARACTERS BEING PROCESSED
0480 REM * AND IN BETWEEN STRINGS IS SET TO ZERO.
0490 REM * THE TOTAL NUMBER OF THESE STRINGS IS COUNTED IN
0500 REM * "W0".
0510 I$=I$+" "
0520 W0=B=0
0530 X$="BLANK"
0540 FOR I=1 TO LEN(I$)
0550 Y$=EXT$(I$,I,I)
0560 IF Y$<>" " GOTO 590
0570 Y$="BLANK"
0580 GOTO 710
0590 FOR J=0 TO 9
0600 IF Y$<>STR$(J) GOTO 630
0610 Y$="NUMBER"
0620 GOTO 710
```

```
0630  NEXT J
0640  FOR J=1 TO 26
0650  IF Y$<>A$(J) GOTO 680
0660  Y$="LETTER"
0670  GOTO 710
0680  NEXT J
0690  Y$="MISC"
0700  GOTO 730
0710  IF X$=Y$ GOTO 780
0720  IF B=0 GOTO 750
0730  WO=WO+1
0740  W$(WO)=EXT$(I$,B,I-1)
0750  IF Y$="BLANK" THEN B=0
0760  IF Y$<>"BLANK" THEN B=I
0770  X$=Y$
0780  NEXT I
0790  IF WO=0 GOTO 2620
0800
0810  REM * HERE THE CHARACTER STRINGS ARE CHECKED AGAINST
0820  REM * THE DICTIONARY OF KEYWORDS/NUMBERS IN "D$".
0830  REM * FOR THOSE WHICH ARE SUCCESSFULLY MATCHED, THE
0840  REM * EQUIVALENT CODES, (IN "D") ARE STORED IN "C".
0850  REM * THE TOTAL NUMBER OF CODES IS COUNTED IN "CO"
0860  REM * AND THE ACTUAL VALUES OF KEY NUMBERS IDENTIFIED
0870  REM * ARE RETAINED IN "N".
0880  CO=NO=0
0890  MAT N=ZER(10)
0900  FOR I=1 TO WO
0910  FOR J= 1 TO DO
0920  IF W$(I)=D$(J) GOTO 950
0930  NEXT J
0940  GOTO 1040
0950  CO=CO+1
0960  IF CO>12 GOTO 2620
0970  C(CO)=D(J) MOD 10
0980  IF C(CO)>=2 GOTO 1040
0990  NO=NO+1
1000  REM * THE INPUT CANNOT BE UNDERSTOOD
1010  REM * IF IT CONTAINS MORE THAN 4 NUMBERS ......
1020  IF NO>4 GOTO 2620
1030  N(NO)=D(J) DIV 10
1040  NEXT I
1050  REM *      ...... OR NO NUMBERS AT ALL.
1060  IF NO=0 GOTO 2620
1070
1080  REM * THE CODES STORED IN "C" ARE NOW APPLIED TO THE
1090  REM * STATE TABLE TO DISCOVER WHETHER THEY CONFORM TO
1100  REM * THE SYNTACTIC RULES ASSOCIATED WITH HEIGHT RANGE
1110  REM * INPUTS. THE FIRST 7 COLUMNS OF "T" REPRESENT THE
1120  REM * 7 CODE VALUES AND EACH ROW CORRESPONDS TO A NEW
1130  REM * STATE IN THE ANALYSING PROCESS.
1140  REM * CONSIDERING THE NON ZERO VALUES OF "T" ;
1150  REM *    THE NUMBER FORMED BY THE 2 RIGHT HAND DIGITS
1160  REM *    INDICATES THE NEXT STATE IN THE PROCESS.
1170  REM *    IF THERE ARE ANY DIGITS TO THE LEFT OF THESE
1180  REM *    TWO, THEY ARE STORED IN "A$" AS ACTION DIGITS
1190  REM *    WHICH TRANSLATE THE INPUT INTO THE INTENDED
1200  REM *    HEIGHT RANGE.
1210  REM * "S1,S2" CONTAIN THE CURRENT AND FOLLOWING STATES.
1220  REM * "P" POINTS THROUGH THE CODE LIST IN "C".
1230  S1=1
1240  P=1
```

```
1250  A$=""
1260  REM * ISOLATE THE NEXT STATE AND ANY ACTION DIGITS.
1270  S2=ABS(T(S1,C(P)) MOD 100)
1280  A=T(S1,C(P)) DIV 100
1290  REM * IF "P"="CO" THEN THE LAST CODE VALUE IS BEING
1300  REM * PROCESSED.
1310  REM * IF ALSO "A"=0 THEN NO ACTION DIGITS ARE
1320  REM * PRESENT WHICH IS AN ERROR CONDITION.
1330  IF P=CO THEN IF A=0 GOTO 1520
1340  REM * IF "A"<0 THEN ACTION DIGITS ARE ONLY STORED IF
1350  REM * THE LAST CODE VALUE IS BEING PROCESSED.
1360  IF A<0 THEN IF P<CO GOTO 1440
1370  IF A=0 GOTO 1440
1380  REM * ALL ACTION DIGITS FOUND ARE STORED IN A
1390  REM * CONTINUOUS STRING.
1400  A$=A$+STR$(ABS(A))
1410  IF P=CO GOTO 2130
1420  REM * IF "S2"=0, THEN NO NEW STATE IS INDICATED ;
1430  REM * AN ERROR CONDITION.
1440  IF S2=0 GOTO 1520
1450  P=P+1
1460  S1=S2
1470  GOTO 1270
1480
1490  REM * AN ERROR CONDITION HAS OCCURED . HOWEVER IT MAY
1500  REM * BE POSSIBLE TO CLARIFY THE USER'S INPUT
1510  REM * INTERACTIVELY.
1520  IF NO=1 GOTO 2620
1530  PRINT
1540  PRINT "PLEASE CLARIFY YOUR INPUT"
1550  PRINT "IS IT INTENDED TO MEAN :";
1560  IF NO<>2 GOTO 1590
1570  A$="13"
1580  PRINT N(1);"FT  - ";N(2);"FT  ?"
1590  IF NO<>3 GOTO 1620
1600  A$="123"
1610  PRINT N(1);"FT";N(2);"IN  - ";N(3);"FT  ?"
1620  IF NO<>4 GOTO 1650
1630  A$="1234"
1640  PRINT N(1);"FT";N(2);"IN  - ";N(3);"FT";N(4);"IN  ?"
1650  GOSUB 1810
1660  IF NO<>3 GOTO 2600
1670  A$="134"
1680  PRINT
1690  PRINT "SHOULD IT BE : ";N(1);"FT  - ";N(2);"FT";
1700  PRINT N(3);"IN  ?"
1710  GOSUB 1810
1720  A$="1247"
1730  PRINT
1740  PRINT "IS IT : ";N(1);"FT";N(2);"IN  - ";N(1);"FT";
1750  PRINT N(3);"IN  ?"
1760  GOSUB 1810
1770  GOTO 2600
1780
1790  REM * THIS OPEN SUBROUTINE HANDLES YES/NO REPLIES FROM
1800  REM * THE USER.
1810  PRINT "PLEASE TYPE Y OR N"
1820  INPUT I$
1830  IF I$="YES" THEN I$="Y"
1840  IF I$="NO" THEN I$="N"
1850  IF I$="Y" GOTO 2130
1860  IF I$<>"N" GOTO 1810
```

```
1870 RETURN
1880
1890 REM * THE ACTION DIGITS STORED IN "A$" ARE NOW
1900 REM * EXAMINED.  DIGIT VALUES BETWEEN 1 AND 4 SERVE
1910 REM * TO MAP THE NUMBERS CONTAINED WITHIN THE INPUT
1920 REM * STRING (IN "N") INTO THE ARRAY "L".
1930 REM * THIS MAPPING IS BASED UPON A FIXED FORMAT OF
1940 REM * THE HEIGHT RANGE WHICH CAN THUS BE READ
1950 REM * DIRECTLY FROM "L".
1960 REM *    E.G. IF "N" CONTAINS NUMBERS  5,6,4  AND
1970 REM *    "A$" = "134", THEN 5 WILL BE COPIED INTO
1980 REM *    "L(1)", 6 INTO "L(3)" AND 4 INTO "L(4)".
1990 REM *    THEREFORE AS ARRAY "L" NOW READS;  5,0,6,4
2000 REM *    THIS IMPLIES A HEIGHT OF  5FT 0  IN - 6FT 4IN.
2010
2020 REM * ACTION DIGITS 5,6 AND 7 DENOTE SUPPLEMENTARY
2030 REM * TASKS WHICH SHOULD BE PERFORMED.
2040 REM *    VALUES 5 AND 6 INDICATE THAT THE LOWER AND
2050 REM *    UPPER LIMITS (RESPECTIVELY) OF THE RANGE
2060 REM *    MUST BE ADJUSTED.
2070 REM *    E.G: IF THE INPUT IS:
2080 REM *              MORE THAN 5FT, LESS THAN 5FT 6IN
2090 REM *    THIS IS INTERPRETED AS:  5FT 1IN - 5FT 5IN
2100 REM *    VALUE 7 CAUSES AN INPUT OF THE FORM:
2110 REM *              5 FT 6 IN TO 8 IN
2120 REM *    TO BE AMENDED TO 5 FT 6 IN TO 5 FT 8 IN
2130 MAT L=ZER(6)
2140 N1=0
2150 FOR I=1 TO LEN(A$)
2160 P=VAL(EXT$(A$,I,I))
2170 IF P<=4 GOTO 2220
2180 IF P=5 THEN L(5)=+1
2190 IF P=6 THEN L(6)=-1
2200 IF P=7 THEN L(3)=L(1)
2210 GOTO 2250
2220 IF L(P)<>0 GOTO 1520
2230 N1=N1+1
2240 L(P)=N(N1)
2250 NEXT I
2260
2270 REM * IT MAY BE NECESSARY TO INTERCHANGE THE VALUES
2280 REM * OF "L" SO THAT "L(1)/L(2)" CONTAINS THE LOWER
2290 REM * LIMIT AND "L(3)/L(4)" THE UPPER LIMIT.
2300 IF L(1)=0 THEN 2430
2310 IF L(3)=0 THEN 2430
2320 IF L(1)+L(2)/12 <= L(3)+L(4)/12 GOTO 2430
2330 X=L(1)
2340 L(1)=L(3)
2350 L(3)=X
2360 X=L(2)
2370 L(2)=L(4)
2380 L(4)=X
2390
2400 REM * WERE THIS PROGRAM A SUBROUTINE,THE RANGE LIMITS
2410 REM * IN "L" WOULD BE PASSED BACK AS PARAMETERS TO
2420 REM * THE MAIN ROUTINE.
2430 PRINT
2440 PRINT "THIS INPUT IS UNDERSTOOD TO MEAN :"
2450 L(1)=L(1)*12+L(2)+L(5)
2460 IF L(1)<>0 THEN 2490
2470 PRINT "NO LOWER LIMIT"
2480 GOTO 2500
```

```
2490  PRINT "LOWER LIMIT ";L(1) DIV 12;"FT ";L(1) MOD 12;"IN"
2500  L(3)=L(3)*12+L(4)+L(6)
2510  IF L(3)<>0 THEN 2540
2520  PRINT"NO UPPER LIMIT"
2530  GOTO 2550
2540  PRINT "UPPER LIMIT ";L(3) DIV 12;"FT ";L(3) MOD 12;"IN"
2550  PRINT
2560  PRINT "TYPE IN A FURTHER HEIGHT RANGE"
2570  PRINT "OR TYPE 0 TO FINISH"
2580  PRINT
2590  GOTO 390
2600  PRINT
2610  PRINT "IN THAT CASE"
2620  PRINT
2630  PRINT "YOUR INPUT HAS NOT BEEN UNDERSTOOD"
2640  PRINT "REMEMBER THAT A RANGE IS NEEDED"
2650  PRINT "RATHER THAN A SINGLE HEIGHT."
2660  GOTO 2550
2670  STOP
2680  DATA "A","B","C","D","E","F","G","H","I","J","K","L","M"
2690  DATA "N","O","P","Q","R","S","T","U","V","W","X","Y","Z"
2700  DATA 50
2710  DATA "0",001,"1",011,"2",021,"3",031,"4",041,"5",051
2720  DATA "6",061,"7",071,"8",081,"9",091,"10",101,"11",111
2730  DATA "ONE",011,"TWO",021,"THREE",031,"FOUR",041
2740  DATA "FIVE",051,"SIX",061,"SEVEN",071,"EIGHT",081
2750  DATA "NINE",091,"TEN",101,"ELEVEN",111
2760  DATA "FEET",2,"FOOT",2,"FT",2,"'",2
2770  DATA "INCHES",3,"INCH",3,"INS",3,"IN",3
2780  DATA "TO",4,"-",4
2790  DATA "ABOVE",5,"BIGGER",5,"GREATER",5,"HIGHER",5
2800  DATA "MORE",5,"OVER",5,"PLUS",5,"TALLER",5,"+",5
2810  DATA "BELOW",6,"LESS",6,"LOWER",6,"SHORTER",6
2820  DATA "SMALLER",6,"UNDER",6
2830  DATA "NOT",7,"NO",7
2840  DATA 71
2850  DATA 1,1,02,  1,5,14,  1,6,18,    1,7,24,    2,2,03
2860  DATA 2,4,10,  3,1,04,  3,4,10,    3,5,101,   3,6,301
2870  DATA 3,7,22,  4,3,05,  4,4,06,    4,5,1201,  4,6,3401
2880  DATA 4,7,23,  5,4,06,  5,5,1201,  5,6,3401,  5,7,23
2890  DATA 6,1,07,      7,2,12308,  7,3,124700,  8,1,-409
2900  DATA 9,3,400,    10,1,11,    11,2,1312,   12,1,-413
2910  DATA 13,3,400,   14,1,15,    15,2,1516,   16,1,-217
2920  DATA 16,6,18,    16,7,24,    17,2,03,     17,3,201
2930  DATA 17,6,218,   17,7,224,   18,1,19,     19,2,3620
2940  DATA 20,1,-421,  20,5,14,    20,7,24,     21,2,03
2950  DATA 21,3,401,   21,5,414,   21,7,424,    22,5,300
2960  DATA 22,6,100,   23,5,3400,  23,6,1200,   24,5,25
2970  DATA 24,6,29,    25,1,26,    26,2,327,    27,1,-428
2980  DATA 27,5,14,    27,2,24,    28,2,03,     28,3,401
2990  DATA 28,5,414,   28,7,424,   29,1,30,     30,2,131
3000  DATA 31,1,-232,  31,6,18,    31,7,24,     32,2,03
3010  DATA 32,3,201,   32,6,218,   32,7,224
3020  END
```

(ALL USER INPUTS ARE UNDERLINED)

```
TYPE IN THE HEIGHT RANGE IN FEET AND INCHES
AND THEN PRESS <RETURN>

5FT 9IN TO 6FT 3

THIS INPUT IS UNDERSTOOD TO MEAN :
LOWER LIMIT  5  FEET 9 INCHES
UPPER LIMIT  6  FEET 3 INCHES

TYPE IN A FURTHER HEIGHT RANGE
OR TYPE 0 TO FINISH

4'6 - 5'

THIS INPUT IS UNDERSTOOD TO MEAN :
LOWER LIMIT  4  FEET 6 INCHES
UPPER LIMIT  5  FEET 0 INCHES

TYPE IN A FURTHER HEIGHT RANGE
OR TYPE 0 TO FINISH

5 FOOT 4 TO 6

PLEASE CLARIFY YOUR INPUT
IS IT INTENDED TO MEAN :  5 FT 4 IN  -  6 FT  ?
PLEASE TYPE Y OR N
N

SHOULD IT BE :  5 FT  -  4 FT 6 IN  ?
PLEASE TYPE Y OR N
N

IS IT :  5 FT 4 IN  -  5 FT 6 IN  ?
PLEASE TYPE Y OR N
Y

THIS INPUT IS UNDERSTOOD TO MEAN :
LOWER LIMIT  5  FEET 4 INCHES
UPPER LIMIT  5  FEET 6 INCHES

TYPE IN A FURTHER HEIGHT RANGE
OR TYPE 0 TO FINISH

UNDER 5FT 2

THIS INPUT IS UNDERSTOOD TO MEAN :
NO LOWER LIMIT
UPPER LIMIT  5  FEET 1 INCHES

TYPE IN A FURTHER HEIGHT RANGE
OR TYPE 0 TO FINISH

SIX FEET TWO INCHES TO FOUR INS

THIS INPUT IS UNDERSTOOD TO MEAN :
LOWER LIMIT  6  FEET 2 INCHES
UPPER LIMIT  6  FEET 4 INCHES
```

TYPE IN A FURTHER HEIGHT RANGE
OR TYPE 0 TO FINISH

<u>5' 10 - 5' 7</u>

THIS INPUT IS UNDERSTOOD TO MEAN :
LOWER LIMIT 5 FEET 7 INCHES
UPPER LIMIT 5 FEET 10 INCHES

TYPE IN A FURTHER HEIGHT RANGE
OR TYPE 0 TO FINISH

<u>0</u>

APPENDIX II

A SELECTION OF COMMENTS WRITTEN BY USERS OF ISLA

Topic	Comment
Street directions	"Good. Rather slow getting to street but directions clear and precise".
Train times	"It says the journey to Wellingborough takes 1 hr. 40 mins. (Note: This information was incorrect). Very interesting, well thought out program.
	People are likely to type spelling mistakes. I hope it copes with these satisfactorily".
Historical places	"Just tried it out of curiosity. Great".
-	"Some subsidiary questions could be permanently displayed on the wall to save time".
-	"Very useful indeed. Please expand with print option as well for trains, etc."
-	"Came to see if it was working. It wasn't but I haven't given up yet!" (Note: ISLA not available due to computer failure).
Films and train service	"Instructions easy, information adequate. Bit slow".
Street directions	"Apparently not available for persons travelling by motor car!"
Films, theatre	"Good and well explained but slow. I could have looked it up in a newspaper faster. Enjoyed the questions at the end. Do away with screen 1. Screen 2; Typing 'OK', a waste of time but I see why you do it". (Screen 1 contained the introduction. On screen 2 the user was asked to type 'OK' and press 'RETURN' to get him acquainted with the return key).

Topic	Comment
Weather	"Screen printout rather slow!"
Trains	"No difficulties at all except when it suddenly said LOG OFF NOW! I didn't know how to log off so I cleared off instead". (Note: This was a computer generated message, independent of ISLA, warning that the service was about to close for the day).
Pubs	"Refused to give me details of pub typed in. It just kept referring to the Spread Eagle, Charles St."
-	"Found it curiously interesting, easy to follow instructions, an enjoyable way of seeking out information".
Discos, weather	"Very good, interesting, doesn't make information sound boring and useless. I don't see why it cannot have more topics to choose from".
Train times and fares	"Is a necessary part of the service required. Will save a lot of time for staff".
-	"Not enough topics".
Trains	"Expand station library".
Trains	"I got the info I wanted so very good".
Didn't get started	"On pressing 'return' button several times, no change on screen from a mass of ??O.V.H.G?? etc. This does nothing to foster my faith in computers which for me is a dirty word. What is the justification of modern expense on technology if this is the result?" (Note: ISLA not available due to computer failure).

Topic	Comment
Lots of them	"Not possible to ask questions".
Various: Quiz Trains Evening out suggestion	 "Very good but rather slow" "Did not have Hull information" "Excellent"
Cinema information	"Pretty good machine, it's a shame there aren't more. The readout appears a bit slow".
Hotels and restaurants	"Very useful but can be tedious at times, though probably unavoidable with complexity and magnitude of this section".
Restaurants	"A bit slow. It gives no indication of price. However, question about favourite place quite interesting".
Films	"Future programmes for cinemas for say 3 months in advance required. Look into the possibility of reviews".
-	"Beats T.V."

APPENDIX III

TRANSCRIPTION OF THE QUESTIONNAIRE DESIGNED TO INVESTIGATE THE USE MADE OF INTERACTIVE DIALOGUE GUIDELINES

I am a research student at Leicester Polytechnic interested in the problems associated with the design of interactive software. As a computer specialist, your response to this questionnaire would be of great value.

This document is composed of two sections. Part two will be given to you when you have replied to part one.

PART ONE

The spread of visual display computer terminals has enabled people in many professional fields and walks of life to benefit from the use of interactive programs.

Typical applications are:

business simulation programs,
graphics packages for the display of scientific data,
tutorial programs supplementing a course of lectures,
and public information systems.

Consider a situation in which you were commissioned to design an interactive system of the type described. Which, if any of the following approaches would you employ in carrying out the task?

Tick one or more of the appropriate boxes

☐ Rely solely on your previous programming experience

☐ Consult your colleagues and discuss ideas with them

☐ Refer to books or research papers for guidelines on interactive system design

☐ Ask potential users what they would expect from the program

☐ Look in the software library to see if any similar programs exist

☐ None of the above

If you would adopt any techniques not listed, please indicate them below;

PART TWO

Suppose it is brought to your attention that the following set of guidelines had been published as an aid to interactive system design:

(a) "Use the user's model": Use a model of the activity being undertaken which corresponds to that of the user, and program the interactive dialogue as if it were a conversation between two users mutually accepting this model.

(b) "Observability and controllability": Envisage the system as an automaton controlled by the user and make it simple to observe and control. In particular, provide a 'reset' command which aborts the current activity cleanly and print out the current state of a variable when requesting a new value.

(c) "Immediate feedback": Give the user feedback by making an immediate unambiguous response to each of his inputs. This should be sufficient to identify the type of activity taking place.

(d) "Involve the users": Involve the users in the design of the system. The designer's perception of 'ease of use', 'ease of learning' etc. is likely to be quite different from that of the ultimate end user.

(e) "Help facilities": All dialogues can benefit from a help facility which the user can call on when he is unclear as to what is possible or what has happened.

(f) "Error checking": Minimise errors by creating logical sequences of operations. However, mistakes will nonetheless occur so check that common errors do not have a catastrophic effect. Validation of data on entry is useful but over reliance on automatic checks can cause too much data to be falsely rejected. Also users may be encouraged to rely on these checks and to become careless typists.

(g) "Provide flexibility": Flexibility should be built into the program to handle different levels of user experience. A friendly and wordy style of computer dialogue may be ideal for casual users but it would be extremely irksome if it had to be experienced several times a day.

Having read these guidelines would you, on balance, find them useful to you as an interactive system designer?

Tick the appropriate box.

☐ YES ☐ NO ☐ DON'T KNOW

If you wish to expand upon your answer or have any further comments, please write them below:

Thankyou for your help,

Martin Maguire.

APPENDIX IV

SUBROUTINE TO ACCEPT AN INPUT OF ONE OR MORE INTEGER NUMBERS

```
      SUBROUTINE NUMSIN(N,NUM,LOWEST,HIGHST,MOST,RETN)
C*    ================================================
C*
C*  THIS FORTRAN SUBROUTINE REQUIRES THE USER TO TYPE IN ONE
C*  OR MORE INTEGER NUMBERS, FOR EXAMPLE IN RESPONSE TO A
C*  MENU DISPLAY ON THE SCREEN.
C*  THE ROUTINE RECEIVES THE FOLLOWING PARAMETERS;
C*  "LOWEST"=THE SMALLEST NUMBER WHICH MAY BE TYPED IN.
C*  "HIGHST"=THE HIGHEST NUMBER.
C*  "MOST"  =THE HIGHEST NUMBER OF NUMBERS WHICH MAY BE
C*           TYPED.
C*           (THIS VALUE MUST BE LESS THAN OR EQUAL TO 20)
C*  "RETN"=  THIS MAY TAKE EITHER THE VALUE " " OR "ALLOW".
C*           IF " " THEN THE USER MUST TYPE IN AT LEAST ONE
C*           NUMBER.
C*           IF "ALLOW" THEN A BLANK INPUT IS ALSO ALLOWED,
C*           "N" TAKING THE VALUE 0 ON EXIT.
C*
C*  THE FOLLOWING VALUES ARE RETURNED TO THE CALLING ROUTINE
C*  "NUM"=IS AN ARRAY CONTAINING THE ACTUAL NUMBERS
C*        THEMSELVES.
C*  "N"=IS THE NUMBER OF THEM.
C*
C*
C*  "CALC" =EACH NUMBER TYPED IN IS CALCULATED IN "CALC".
C*  "FOUND"=WHEN A NUMBER IS BEING PROCESSED,"FOUND" IS SET
C*          TO "YES" DURING THE SEARCH FOR THE FIRST NUMBER
C*          OR BETWEEN TWO NUMBERS,"FOUND" = "NOTYET".
C*          THUS IF "NUM"=0 & "FOUND"="NOTYET",THEN NO
C*          NUMBER HAS BEEN FOUND. IF HOWEVER "NUM"=0 &
C*          "FOUND"="YES" THEN ZERO HAS BEEN ENTERED.
C*
C*  NOTE THAT CONTINUATION LINES ARE DENOTED BY A SINGLE
C*  ASTERISK.
C*
      IMPLICIT INTEGER(A-Z)
      DIMENSION NUM(30),HUMAN(101),DIGIT(10)
C*  "HUMAN"=RECEIVES THE USERS INPUT.
C*  "DIGIT"=CONTAINS THE DIGITS 0 TO 9 IN CHARACTER FORM
C*          SO THAT DIGITS CAN BE RECOGNISED IN "HUMAN".
C*
      DATA (DIGIT(I),I=1,10)/"0","1","2","3","4","5",
     *                       "6","7","8","9"/
    1 N=0
      CALC=0
      DO 2 I=1,30
      NUM(I)=0
    2 CONTINUE
      FOUND="NOTYET"
      READ(5,3)(HUMAN(I),I=1,100)
    3 FORMAT(100A1)
      DO 4 I=1,100
      END=101-I
      IF(.NOT.HUMAN(I).IS." ")GOTO 5
    4 CONTINUE
      IF(RETN.IS."ALLOW")GOTO 19
      GOTO 7
    5 DO 17 I=1,END+1
      DO 6 J=1,10
      IF(.NOT.DIGIT(J).IS.HUMAN(I))GOTO 6
      CALC=CALC*10+(J-1)
      FOUND="YES"
```

```
      GOTO 17
    6 CONTINUE
      IF(FOUND.IS."NOTYET")GOTO 17
      IF(CALC.GE.LOWEST.AND.CALC.LE.HIGHST)GOTO 12
    7 RANGE=HIGHST-LOWEST
      IF(RANGE.EQ.1.AND.MOST.EQ.1)WRITE(6,8)LOWEST,HIGHST
    8 FORMAT(11X,"PLEASE TYPE IN EITHER ",I2," OR ",I2/)
      IF(RANGE.GT.1.AND.MOST.EQ.1)WRITE(6,9)LOWEST,HIGHST
    9 FORMAT(11X,"PLEASE TYPE IN A NUMBER FROM ",I2," TO ",
     *       I2/)
      IF(RANGE.EQ.1.AND.MOST.GT.1)WRITE(6,10)LOWEST,HIGHST
   10 FORMAT(11X,"PLEASE TYPE IN EITHER ",I2,",",I2,
     *      " OR BOTH"/)
      IF(RANGE.GT.1.AND.MOST.GT.1)WRITE(6,11)LOWEST,HIGHST
   11 FORMAT(11X,"NUMBERS TYPED IN SHOULD BE IN THE RANGE "
     *     ,I2," - ",I2/)
      GOTO 1
   12 IF(N.EQ.0)GOTO 14

C*    CHECK IF CURRENT NUMBER HAS ALREADY BEEN STORED.
      DO 13 J=1,N
      IF(NUM(J).EQ.CALC)GOTO 17
   13 CONTINUE
   14 N=N+1
      NUM(N)=CALC
      FOUND="NOTYET"
      CALC=0

C* CHECK IF THE MAX. NUMBER OF NUMBERS HAS BEEN EXCEEDED
      IF(N.LE.MOST)GOTO 17
      IF(MOST.EQ.1)WRITE(6,15)
   15 FORMAT(11X,"UNFORTUNATELY YOU CAN ONLY TYPE ONE"
     *      " NUMBER",/11X,"PLEASE TRY AGAIN."/)
      IF(MOST.GT.1)WRITE(6,16)MOST
   16 FORMAT(11X,"UNFORTUNATELY YOU CAN ONLY TYPE ",I2,
     *           " NUMBERS AT THE MOST"
     *      /11X,"PLEASE TRY AGAIN"/)
      GOTO 1
   17 CONTINUE
      IF(N.GE.1)GOTO 19
      WRITE(6,18)
   18 FORMAT(11X,"NO NUMBER HAS BEEN RECOGNISED."
     *      /11X,"TRY AGAIN, REMEMBERING TO USE DIGITS"
     *           " RATHER THAN WORDS."/)
      GOTO 1
   19 RETURN
      END
```

APPENDIX V

DEMONSTRATION PROGRAM OF THE TEX METHOD AND SAMPLE PROGRAM RUN

```
C*          DEMONSTRATION PROGRAM OF TEX METHOD
C*    =====================================================
C*       NON STANDARD FORTRAN FEATURES:
C*       INVERTED COMMAS DELIMIT CHARACTER STRINGS
C*       ASTERISKS DENOTE CONTINUATION LINES
C*       SUBSCRIPT VARIABLES MAY ALSO BE SUBSCRIPTED
C*    =====================================================
      IMPLICIT INTEGER(A-Z)
      DOUBLE PRECISION ENTRY
      DIMENSION ALPHA(26),INPUT(51),ILEN(10),IWORD(10,15),
     *          STORE(10),ICOM(10),LINK(10),
     *          DLEN(11),ABB(11),DWORD(11,15),DCOM(11),
     *          PNTR(11,4),KEY(9),ENTRY(9,2)

C* ARRAYS
C*    "ALPHA" CONTAINS A-Z ALPHABET FOR INPUT WORD
C*            IDENTIFICATION
C*    "INPUT" RECEIVES THE INPUT STRING
C*    "IWORD,ILEN" STORES INPUT WORDS INDIVIDUALLY (ONE
C*            PER ROW) & THE CORRESPONDING WORD LENGTHS.
C*    "DWORD,DLEN,ABB,DCOM,PNTR" FORM THE DICTIONARY.
C*            RESPECTIVELY THEY CONTAIN;
C*            THE WORDS,THEIR LENGTHS,MINIMUM ABBREVIATIONS,
C*            COMPONENTS AND POINTERS (TO THE TEXTUAL INPUT
C*            LIST.)
C*    "STORE" RECORDS POSSIBLE DICTIONARY WORD MATCHES ON
C*            INPUT WORDS.
C*    "ICOM,LINK" RECORD THE COMPONENT AND POINTERS OF EACH
C*            INPUT WORD IDENTIFIED IN THE DICTIONARY .
C*    "ENTRY,KEY" COMPRISE THE TEXTUAL INPUT LIST.
C*            "ENTRY" CONTAINS ALLOWABLE INPUTS AND "KEY"
C*            CONTAINS THE APPROPRIATE KEY FOR EACH.
C*    ==========================================================
      DATA (ALPHA(I),I=1,26)/"A","B","C","D","E","F","G",
     *           "H","I","J","K","L","M","N","O","P","Q",
     *           "R","S","T","U","V","W","X","Y","Z"/
      DATA BLANK,YES,NO/" ","Y","N"/
      DATA D/11/
      DATA (DLEN(I),ABB(I),(DWORD(I,J),J=1,15),DCOM(I),
     *     (PNTR(I,K),K=1,4),I=1,11)/
     *  3, 3,"A","N","D",12*" "                      , 0,0,0,0,0,
     *  7, 3,"C","E","N","T","R","A","L",8*" ",10,0,0,0,0,
     *  4, 2,"E","A","S","T",11*" "                  ,15,0,0,0,0,
     *  4, 2,"E","T","O","N",11*" "                  ,20,2,0,0,0,
     *  3, 3,"H","A","M",12*" "                      ,25,1,8,0,0,
     * 11, 6,"H","A","M","M","E","R","S","M","I","T","H"
     *                                       ,4*" ",30,3,0,0,0,
     *  6, 4,"H","A","R","R","O","W",9*" "           ,35,4,9,0,0,
     *  8, 3,"H","O","U","N","S","L","O","W",7*" "
     *                                             ,40,5,6,7,0,
     * 10, 5,"W","E","A","L","D","S","T","O","N","E",5*" "
     *                                             ,45,4,0,0,0,
     *  4, 3,"W","E","S","T",11*" "                  ,50,0,0,0,0,
     *  7, 3,"W","I","N","D","S","O","R",8*" ",55,2,0,0,0/
      DATA (KEY(I),(ENTRY(I,J),J=1,2),I=1,9)/
     *    1525,"EAST HAM"," ",
     *    2055,"ETON AND WIN","DSOR",
     *    0030,"HAMMERSMITH"," ",
     *    3545,"HARROW AND W","EALDSTONE",
     *    1040,"HOUNSLOW CEN","TRAL",
     *    1540,"HOUNSLOW EAS","T",
     *    4050,"HOUNSLOW WES","T",
```

```
     *    2550,"WEST HAM"," ",
     *    3550,"WEST HARROW"," "/
      WRITE(6,1)
    1 FORMAT("1",9X,"THIS PROGRAM DEMONSTRATES THE TEX "
     *              "METHOD."
     *        //10X,"INPUTS WILL BE MATCHED AGAINST THE "
     *              " FOLLOWING"
     *         /10X,"LIST OF 8 LONDON UNDERGROUND STATIONS;"
     *        //10X,"EAST HAM, WEST HAM, HAMMERSMITH,"
     *         /10X,"WEST HARROW, HARROW & WEALDSTONE,"
     *         /10X,"HOUNSLOW CENTRAL, HOUNSLOW EAST, "
     *              "HOUNSLOW WEST."
     *        //10X,"THE CLOSEST MATCH OR MATCHES WITH THE "
     *              "INPUT"
     *         /10X,"WILL BE DISPLAYED FOR INSPECTION.")
C*
C*   THE USER TYPES IN THE INPUT STRING - IF IT IS BLANK
C*   THEN THE PROGRAM HALTS.
      WRITE(6,2)
    2 FORMAT(/10X,"NOW TYPE IN THE NAME OF AN UNDERGROUND "
     *       "STATION" /10X,"AND THEN PRESS <RETURN>"/)
    3 READ(5,4)(INPUT(I),I=1,50)
    4 FORMAT(50A1)
      DO 5 I=1,50
      J=51-I
      IF(.NOT.INPUT(J).IS.BLANK)GOTO 6
    5 CONTINUE
      GOTO 46
C*
C*   "LENGTH" RECORDS INPUT STRING LENGTH
    6 LENGTH=J+1
C*
C*   EACH INPUT WORD IS NOW STORED IN A SEPARATE ROW OF
C*   "IWORD".  "ONWORD & POINT" ARE VARIABLES USED IN
C*   THIS PROCESS.  "W & ILEN" RECORDS THE NUMBER AND
C*   LENGTHS OF WORDS IDENTIFIED.
      W=0
      POINT=0
      ONWORD=NO
      DO 7 I=1,10
      DO 7 J=1,15
      IWORD(I,J)=BLANK
    7 CONTINUE
      DO 11 I=1,LENGTH
      DO 8 J=1,26
      IF(INPUT(I).IS.ALPHA(J))GOTO 9
    8 CONTINUE
      IF(ONWORD.IS.NO)GOTO 11
      ILEN(W)=POINT
      ONWORD=NO
      POINT=0
      GOTO 11
    9 IF(ONWORD.IS.NO)GOTO 10
      POINT=POINT+1
      IWORD(W,POINT)=INPUT(I)
      GOTO 11
   10 W=W+1
      POINT=1
      IWORD(W,1)=INPUT(I)
      ONWORD=YES
   11 CONTINUE
      IF(W.GT.0)GOTO 13
```

```
      WRITE(6,12)
   12 FORMAT(10X,"NO WORDS RECOGNISED - PLEASE TRY AGAIN"/)
      GOTO 3
C*
C*    EACH INPUT WORD IS CONSIDERED FOR POSSIBLE IDENTIF-
C*    -CATION IN THE DICTIONARY. TO COMPARE AN INPUT AND
C*    DICTIONARY WORD, MATCHING PROCEDES LETTER FOR LETTER.
C*    "IP & DP" INDICATE THE CURRENT LETTERS BEING COMPARED.
   13 NC=0
      NL=0
      UNREC=NO
      DO 33 I=1,W
      S=0
      DO 18 J=1,D
      IP=0
      DP=0
   14 IP=IP+1
      DP=DP+1
C*
C*    AN INPUT WORD IS JUDGED TO MATCH A DICTIONARY WORD IF;
C*    (1) IP & DP SIMULTANEOUSLY POINT TO BLANK CHARACTERS,
C*        THE ENDS OF BOTH WORDS HAVING BEEN REACHED, OR IF
C*    (2) IP POINTS TO THE END OF THE INPUT WORD WHICH IS
C*        EQUAL TO OR LONGER THAN THE MINIMUM ABBREVIATION
C*        OF THE DICTIONARY WORD.
      IF(IWORD(I,IP).IS.BLANK.AND.
     *     (DWORD(J,DP).IS.BLANK.OR.IP-1.GE.ABB(J)))GOTO 26
      IF(IWORD(I,IP).IS.DWORD(J,DP))GOTO 14
C*
C*    MATCH FAILURE BETWEEN THE CURRENT INPUT AND DICTIONARY
C*    WORDS HAS OCCURRED. CONTINUE MATCHING IF THEIR LENGTHS
C*    DIFFER BY <= 1.
      DIFF=DLEN(J)-ILEN(I)
      IF(ABS(DIFF).GT.1)GOTO 18
C*
C*    ASSUME INPUT WORD ERROR OF INCORRECT,SWOPPED,INSERTED
C*    OR MISSING LETTER. ADJUST THE POINTERS ON THE BASIS OF
C*    INPUT AND DICTIONARY WORD LENGTHS.
      IF(ILEN(I).GE.DLEN(J))IP=IP+1
      IF(DLEN(J).GE.ILEN(I))DP=DP+1
      IF(DIFF.NE.0)GOTO 16
C*
C*    IF ERROR FOUND TO BE TWO LETTERS SWOPPED, RE-ADJUST
C*    POINTERS
      IF(IWORD(I,IP-1).IS.DWORD(J,DP).AND.
     *   IWORD(I,IP).IS.DWORD(J,DP-1))GOTO 15
      GOTO 16
   15 IP=IP+1
      DP=DP+1
C*
C*    IF "DP" INDICATES A BLANK CHARACTER,THEN INPUT AND
C*    DICTIONARY WORDS MATCH BUT FOR THE SINGLE ERROR.
   16 IF(DWORD(J,DP).IS.BLANK)GOTO 17
      IF(IWORD(I,IP).IS.DWORD(J,DP))GOTO 15
C*
C*    MATCH FAILS A SECOND TIME. DISCOUNT CURRENT DICTIONARY
C*    WORD AND CONTINUE WITH ITS SUCCESSOR.
      GOTO 18
C*
C*    RECORD THE SINGLE-ERROR-MATCH DICTIONARY WORDS IN
C*    "STORE". IF NO ERROR-FREE-MATCH IS FOUND FOR AN INPUT
C*    WORD,THEN DISPLAY THE RECORDED WORDS FOR THE USER TO
```

```
C*    CLARIFY HIS INTENTION.
   17 S=S+1
      STORE(S)=J
   18 CONTINUE
      IF(S.EQ.0)GOTO 25
      WRITE(6,19)(IWORD(I,J),J=1,15)
   19 FORMAT(/10X,"BY THE WORD .... ",15A1)
      DO 24 J=1,S
   20 WRITE(6,21)(DWORD(STORE(J),K),K=1,15)
   21 FORMAT(10X,"DID YOU MEAN ... ",15A1,"- TYPE Y OR N"/)
   22 READ(5,4)INPUT(1)
      IF(INPUT(1).IS.NO)GOTO 24
      IF(INPUT(1).IS.YES)J=STORE(J)
      IF(INPUT(1).IS.YES)GOTO 26
      WRITE(6,23)
   23 FORMAT(10X,"PLEASE TYPE Y OR N"/)
      GOTO 22
   24 CONTINUE
C*    NOTE THAT AN INPUT WORD WAS UNRECOGNISED.
   25 UNREC=YES
      GOTO 33
C*
C*    CURRENT INPUT WORD NOW IDENTIFIED.
C*    THE COMPONENT ("DCOM") AND  POINTERS ("PNTR") ARE
C*    EXTRACTED FROM THE DICTIONARY ENTRY AND STORED IN
C*    "ICOM" AND "LINK".
C*    ANY DUPLICATE COMPONENTS AND POINTERS ARE DISCARDED.
C*    "NC" COUNTS "ICOM" VALUES. "NL" COUNTS "LINK" VALUES.
   26 IF(DCOM(J).EQ.0)GOTO 29
      IF(NC.EQ.0)GOTO 28
      DO 27 I1=1,NC
      IF(ICOM(I1).EQ.DCOM(J))GOTO 29
   27 CONTINUE
   28 NC=NC+1
      ICOM(NC)=DCOM(J)
   29 N=NL
      DO 32 I1=1,4
      IF(PNTR(J,I1).EQ.0)GOTO 33
      IF(N.EQ.0)GOTO 31
      DO 30 I2=1,N
      IF(LINK(I2).EQ.PNTR(J,I1))GOTO 32
   30 CONTINUE
   31 NL=NL+1
      LINK(NL)=PNTR(J,I1)
   32 CONTINUE
   33 CONTINUE
C*
C*    EACH "LINK" VALUE POINTS TO A POSSIBLE INTERPRETATION
C*    OF THE USERS INPUT IN "ENTRY".
C*    IT MAY BE POSSIBLE TO SELECT THE CORRECT INTERPRETATION.
C*    HOWEVER IF ANY OF THE FOLLOWING 4 CONDITIONS HOLD,
C*    DIFFERENT ACTION IS TAKEN ;
C*    (1) IF NO "LINK" VALUES STORED - ACTIVATE BACK-UP SYSTEM
      IF(NL.EQ.0)GOTO 42
C*    (2) IF ANY INPUT WORDS WERE UNRECOGNISED - DISPLAY ALL
C*        INTERPRETATIONS.
      IF(UNREC.IS.YES)GOTO 37
C*    (3) IF ONLY ONE LINK STORED - DISPLAY SINGLE
C*        INTERPRETATION TO USER.
      IF(NL.EQ.1)GOTO 35
C*    (4) IF MORE THAN TWO "ICOM" VALUES STORED - DISPLAY ALL
C*        INTERPRETATIONS.
```

```
      IF(NC.GT.2)GOTO 37
C*
C*   NONE OF THE ABOVE CONDITIONS APPLY SO CREATE A "MATCH"
C*   VALUE FROM THE TWO COMPONENT VALUES IN "ICOM".
C*   (IF ONLY ONE,CREATE A SECOND VALUE ZERO.)
      IF(NC.EQ.1)ICOM(2)=0
      IF(ICOM(1).GT.ICOM(2))MATCH=ICOM(2)*100+ICOM(1)
      IF(ICOM(2).GT.ICOM(1))MATCH=ICOM(1)*100+ICOM(2)
C*
C*   "MATCH" COMPARED WITH EACH "KEY" POINTED TO BY "LINK".
C*   - IF MATCH IS FOUND, THE USERS INPUT IS IDENTIFIED AS
C*     THE CORRESPONDING STRING IN "ENTRY".
C*   - OTHERWISE DISPLAY ALL INTERPRETATIONS TO USER.
      DO 34 I=1,NL
      IF(MATCH.NE.KEY(LINK(I)))GOTO 34
      LINK(1)=LINK(I)
      GOTO 35
   34 CONTINUE
      GOTO 37
C*   THE COMPUTERS INTERPRETATIONS OF THE USERS INPUT ARE
C*   NOW DISPLAYED. IF NONE ARE CORRECT, THEN THE BACK-UP
C*   SYSTEM SHOULD BE PUT INTO OPERATION.
   35 WRITE(6,36)(ENTRY(LINK(1),J),J=1,2)
   36 FORMAT(/10X,"THE INPUT HAS BEEN IDENTIFIED"
     *       /10X,"AS "/10X,"   ",2A12
     *      //10X,"THE USER SHOULD NOW CLARIFY THIS.")
      GOTO 44
   37 WRITE(6,38)
   38 FORMAT(/10X,"THE INPUT HAS BEEN RECOGNISED"
     *       /10X,"AS ONE OF THE FOLLOWING ;"/)
      DO 40 I=1,NL
      WRITE(6,39)I,(ENTRY(LINK(I),J),J=1,2)
   39 FORMAT(10X,"(",I1,") ",2A12)
   40 CONTINUE
      WRITE(6,41)
   41 FORMAT(/10X,"THE USER NOW SPECIFIES WHICH IS CORRECT.")
      GOTO 44
C*
C*   ARRIVAL HERE DENOTES FAILURE TO INTERPRETE THE USERS
C*   INPUT. THE BACK-UP SYSTEM SHOULD NOW BE UTILISED.
   42 WRITE(6,43)
   43 FORMAT(/10X,"THE INPUT IS NOT SIMILAR TO ANY ITEM"
     *       /10X,"IN THE TEXTUAL INPUT LIST.")
C*
C*   THE USER MAY TERMINATE THE PROG. OR TRY FURTHER INPUTS.
   44 WRITE(6,45)
   45 FORMAT(/10X,"CONTINUE BY INPUTTING ANOTHER NAME OR "
     *             "SIMPLY"
     *       /10X,"PRESS <RETURN> TO STOP THE PROGRAM."//)
      GOTO 3
   46 STOP
      END
```

(ALL USER INPUTS ARE UNDERLINED)

THIS PROGRAM DEMONSTRATES THE TEX METHOD.

INPUTS WILL BE MATCHED AGAINST THE FOLLOWING
LIST OF 8 LONDON UNDERGROUND STATIONS;

EAST HAM, WEST HAM, HAMMERSMITH,
WEST HARROW, HARROW & WEALDSTONE,
HOUNSLOW CENTRAL, HOUNSLOW EAST, HOUNSLOW WEST.

THE CLOSEST MATCH OR MATCHES WITH THE INPUT
WILL BE DISPLAYED FOR INSPECTION.

NOW TYPE IN THE NAME OF AN UNDERGROUND STATION
AND THEN PRESS <RETURN>

HOUNDSLOW

BY THE WORD .... HOUNDSLOW
DID YOU MEAN ... HOUNSLOW - TYPE Y OR N

Y

THE INPUT HAS BEEN RECOGNISED
AS ONE OF THE FOLLOWING ;

(1) HOUNSLOW CENTRAL
(2) HOUNSLOW EAST
(3) HOUNSLOW WEST

THE USER NOW SPECIFIES WHICH IS CORRECT.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.

WINDSOR AND ETON

THE INPUT HAS BEEN IDENTIFIED
AS
ETON AND WINDSOR

THE USER SHOULD NOW CLARIFY THIS.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.

HAMMER

THE INPUT HAS BEEN IDENTIFIED
AS
HAMMERSMITH

THE USER SHOULD NOW CLARIFY THIS.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.

EAST HARROW AND WEST

THE INPUT HAS BEEN RECOGNISED
AS ONE OF THE FOLLOWING ;

(1) HARROW AND WEALDSTONE
(2) WEST HARROW

THE USER NOW SPECIFIES WHICH IS CORRECT.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.

HAMPTON

THE INPUT IS NOT SIMILAR TO ANY ITEM
IN THE TEXTUAL INPUT LIST.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.

CENTRAL

THE INPUT IS NOT SIMILAR TO ANY ITEM
IN THE TEXTUAL INPUT LIST.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.

I WANT WEST HAM

THE INPUT HAS BEEN RECOGNISED
AS ONE OF THE FOLLOWING ;

(1) EAST HAM
(2) WEST HAM

THE USER NOW SPECIFIES WHICH IS CORRECT.

CONTINUE BY INPUTTING ANOTHER NAME OR SIMPLY
PRESS <RETURN> TO STOP THE PROGRAM.